

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр

на тему: **«РОЗРОБКА МЕДИЧНОЇ СИСТЕМИ ОБЛІКУ ПАЦІЄНТІВ
ПРИВАТНОЇ КЛІНІКИ НА МОВІ С++ З ВИКОРИСТАННЯМ СУБД
ACCESS»**

Виконав: студент 4 курсу, групи ПД–41
спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності/спеціалізації)

_____ Добридіна В.Є.
(прізвище та ініціали)

Керівник _____ Коба А.Б.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

Київ –2021

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

Кафедра Інженерії програмного забезпечення
Ступінь вищої освіти - «Бакалавр»
Спеціальність підготовки – 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри
Інженерії програмного
забезпечення

Негоденко

О.В.

“ ____ ” _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТА

ДОБРИДІЙ ВАЛЕРІЙ ЄВГЕНІВНІ

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка медичної системи обліку пацієнтів приватної клініки на мові С++ з використанням СУБД Access»

Керівник роботи: Трінтіна Н.А., к.т.н., доцент кафедри ІІЗ

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом вищого навчального закладу від «12» березня 2021 року №65.

2. Строк подання студентом роботи «01» червня 2021 року

3. Вхідні дані до роботи

Література по роботі з мовою програмування С++;

Науково-технічна література з питань, пов'язаних з програмним забезпеченням щодо роботи з базами даних;

4. Зміст розрахунково-пояснювальної записки(перелік питань, які потрібно розробити).

4.1 Огляд літератури за темою і вибір напрямів досліджень.

4.2 Розробка медичної системи обліку пацієнтів приватної клініки.

4.3 Практична реалізація результатів дослідження.

4.4 Обґрунтування економічної ефективності розробки медичної системи обліку пацієнтів приватної клініки.

4.5 Охорона праці та безпека в надзвичайних ситуаціях.

5. Перелік демонстраційного матеріалу (назва основних слайдів)

1. Актуальність проблеми
2. Існуюче програмне забезпечення
3. Принцип роботи медичної системи
4. Опис медичної системи
5. Архітектура бази даних
6. Логічна діаграма компонентів архітектури програмного забезпечення

6. Дата видачі завдання «19»квітня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	02.11-07.11	Виконано
2	Вимоги до системи	08.11-10.11	Виконано
3	Проектування медичної системи	11.11-18.11	Виконано
4	Опис медичної системи	19.11-28.11	Виконано
5	Концепція та архітектура програмного забезпечення	29.11-04.12	Виконано
6	Вступ, висновки, реферат	05.12-12.12	Виконано
7	Розробка обов'язкових демонстраційних матеріалів	13.11-15.12	Виконано
8	Попередній захист роботи	16.12	
9	Здача роботи	24.12	

Студент _____
(підпис) (прізвище та ініціали)

Керівник роботи _____
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Текстова частина магістерської роботи 117 с., 30 рис., 35 джерел.

СИСТЕМА, ОБЛІК, МЕДИЧНИЙ ЗАКЛАД, ПАЦІЄНТ, БАЗА ДАНИХ, ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, МОДЕРНІЗАЦІЯ

Об'єкт дослідження – сучасні системи обліку пацієнтів приватної клініки.

Предмет дослідження – процес розробки нової найоптимальнішої системи обліку пацієнтів приватної клініки.

Мета роботи – аналіз існуючих методів та засобів обліку пацієнтів приватної клініки та пропозиція нової, найоптимальнішої.

Методи дослідження – системний аналіз, дедуктивний та індуктивний методи використовувалися при визначенні суті понять «Діагноз», «Лабораторне дослідження». Інформаційну базу дослідження складають чинні вітчизняні законодавчі та нормативно-правові акти, методична документація сучасних розробок у сфері обліку пацієнтів.

У роботі удосконалено метод обліку пацієнтів, який засновано на принципі адаптивної рандомізації, для використання в приватних поліклінічних медичних закладах; вперше розроблено програмний додаток щодо реалізації методів та засобів обліку пацієнтів медичної допомоги, з метою підвищення ефективності процесів планування виробничих ресурсів та обслуговування пацієнтів за рахунок застосування нових підходів до планування медичних прийомів.

Результати роботи можуть бути використані при проведенні модернізації системи обліку пацієнтів приватної клініки.

Отже, розроблено та описано медичну систему обліку пацієнтів приватної клініки.

У якості вихідних даних є персональні дані пацієнтів.

Даний додаток може бути використано у медичній сфері, яка потребує автоматичної обробки інформації і її подальшої обробки.

Галузь використання – обробка інформації.

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ	8
ВСТУП.....	9
РОЗДІЛ 1 ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ І ВИБІР НАПРЯМІВ ДОСЛІДЖЕНЬ	12
1.1. Управління обліковими ресурсами у державних медичних закладах	12
1.2. Аналіз закордонних джерел	14
1.3. Аналіз аналогів, щодо об'єкту проектування.....	16
1.3.1. Медична система «Медучет»	16
1.3.2. Медична система «Учет пациентов»	17
1.3.3. Медична система «АРМ амбулаторный прием».....	18
1.3.4. Порівняльний аналіз аналогів	20
1.4. Обґрунтування потреби у новій інформаційній системі	20
РОЗДІЛ 2 РОЗРОБКА МЕДИЧНОЇ СИСТЕМИ ОБЛІКУ ПАЦІЄНТІВ ПРИВАТНОЇ КЛІНІКИ	24
2.1. Постановка завдання.....	24
2.2. Програмне середовище розробки та його загальна характеристика ...	25
2.3. Вибір бази даних	27
2.4. Функціональні вимоги та вимоги до операційної системи.....	36
2.5. Проектування медичної системи обліку пацієнтів приватної клініки	42
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ	59
3.1. Перелік та призначення режимів та структура діалогу	59
3.2. Характеристика та структура даних з якими працює програма	60
3.3. Опис медичної системи обліку пацієнтів приватної клініки.....	66
3.4. Тестування медичної системи обліку пацієнтів приватної клініки	69
РОЗДІЛ 4 ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ МЕДИЧНОЇ СИСТЕМИ ОБЛІКУ ПАЦІЄНТІВ ПРИВАТНОЇ КЛІНІКИ	74
4.1 Техніко-економічне обґрунтування витрат на виконання розробки програмного продукту обробки, формування та передачі звітів виконаних завдань	74
4.2 Аналіз впровадження розробленого додатку	87
РОЗДІЛ 5 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ.....	91
5.1. Характеристика приміщення	91
5.2. Аналіз шкідливих та небезпечних факторів	93
5.3. Розробка заходів з охорони праці	96
РОЗДІЛ 6 ЕКОЛОГІЯ	103
6.1. Аналіз впливу забруднення атмосфери на організм людини	103
6.2. Нормування вмісту забруднюючих речовин в повітрі	107

6.3. Засоби контролю за чистотою повітря у виробничому приміщенні.	110
ВИСНОВКИ	113
СПИСОК ЛІТЕРАТУРИ	115
ДОДАТОК А	119

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

БД	база даних
ВОЯ	внутрішня оцінка якості
ГОСТ	галузевий стандарт
ЗОЯ	зовнішня оцінка якості
ЗУЯ	загальне управління якістю
КДЛ	клініко-лабораторне дослідження
КК	контрольна картка
КЯ	контроль якості
ЛІС	лабораторно-інформаційна система
ЛПЗ	лікувально-профілактичний заклад
МОЗ	Міністерство охорони здоров'я
ОС	операційна система
ПДЗ	гранично допустимі значення
ПК	персональних комп'ютер
ППП	періодична перевірка підконтрольності
AMD	Advanced Micro Devices (процесор)
HDD	hard disk drive (жорсткий диск)
RAM	Random Access Memory (оперативна пам'ять)

ВСТУП

Актуальність дослідження. Лікарня – вид цивільного стаціонарного медичного закладу, спрямованого на лікування хворих і (або) спеціалізовану поглиблену диференціальну діагностику захворювань в основному в стаціонарних умовах.

Очевидно, що обсяг інформації, оброблюваної лікарнею, дуже великий. На даний момент, вся інформація про перебіг лікування пацієнтів встановлення поточного діагнозу в лікарнях зберігається в паперовому вигляді в реєстратурах і архівах. В електронному вигляді представлені в основному тільки інформація про пацієнтів (ПІБ, місце проживання, вік і т.п.), інформація про лікарів, відділеннях і про замовлені талонах. Створення програмного засобу, що дозволяє автоматизувати облік пацієнтів лікарні, аналіз їх симптомів та встановлення діагнозу, тобто що дозволяє зберігати і обробляти не тільки їх паспортну інформацію, але також інформацію про їх симптоми, діагноз, лікаря, що їх лікує, значно спростило б пошук інформації, забезпечило б економію паперу, місця, робочого часу. Дозволило б запобігти втрату даних в результаті пожежі, наприклад, дозволило б створювати резервні копії бази даних, обмежити доступ до інформації і т.п.

Спираючись на вищевикладене актуальність даної магістерської роботи не викликає сумнівів.

Мета та завдання дослідження. Метою даної магістерської роботи є аналіз існуючих методів та засобів обліку пацієнтів приватної клініки та пропозиція нової, найоптимальнішої.

Для досягнення поставленої мети у роботі необхідно виконати низку завдань:

- здійснити огляд літератури за темою дослідження та обрати перспективні напрямки дослідження;

- виконати розробку сучасного засобу обліку пацієнтів приватної клініки;
- здійснити вибір програмного середовища розробки;
- описати функціональні вимоги та вимоги до операційної системи;
- запропонувати механізми проектування засобу обліку пацієнтів приватної клініки;
- виконати практичну реалізацію результатів дослідження;
- навести перелік та призначення режимів та структури діалогу;
- здійснити опис програмного продукту;
- протестувати систему засобу обліку пацієнтів приватної клініки;
- обґрунтувати економічну ефективність розробки засобу обліку пацієнтів приватної клініки;
- навести заходи з охорони праці та безпеки у надзвичайних ситуаціях.

Об'єкт та предмет дослідження. Об'єктом роботи виступають сучасні системи обліку пацієнтів приватної клініки.

Предметом є процес розробки нової найоптимальнішої системи обліку пацієнтів приватної клініки.

Методи наукового дослідження. Для вирішення поставлених завдань використовувалися як загальнонаукові, так і спеціальні методи наукового пізнання. Системний аналіз, дедуктивний та індуктивний методи використовувалися при визначенні суті понять «Діагноз», «Лабораторне дослідження». Інформаційну базу дослідження складають чинні вітчизняні законодавчі та нормативно-правові акти, методична документація сучасних розробок у сфері обліку пацієнтів.

Методологічну основу методів дослідження складає системний підхід при вирішенні проблеми забезпечення високого рівня якості обліку пацієнтів.

Практичне значення отриманих результатів. Результати роботи можуть бути використані при проведенні модернізації системи обліку пацієнтів приватної клініки.

Наукова новизна отриманих результатів. У рамках проведеного дослідження здійснено наукове дослідження методів та засобів обліку пацієнтів медичної допомоги, на основі проведеного дослідження є можливість розкрити основні наукові досягнення:

удосконалено метод обліку пацієнтів, який засновано на принципі адаптивної рандомізації, для використання в приватних поліклінічних медичних закладах;

вперше розроблено програмний додаток щодо реалізації методів та засобів обліку пацієнтів медичної допомоги, з метою підвищення ефективності процесів планування виробничих ресурсів та обслуговування пацієнтів за рахунок застосування нових підходів до планування медичних прийомів.

Достовірність дослідження. При реалізації проекту було вивчено значне число літературних джерел, посібників і статей, що охоплюють обрану тематику. Повний їх список представлений в кінці проекту.

Структура дипломної роботи. Робота складається зі вступу, шести розділів, що включають в себе двадцять один підрозділ, висновків та списку літератури з 35 найменувань, загальний обсяг роботи становить 117 сторінок.

РОЗДІЛ 1

ОГЛЯД ЛІТЕРАТУРИ ЗА ТЕМОЮ І ВИБІР НАПРЯМІВ ДОСЛІДЖЕНЬ

На даний момент в Україні у державних медичних закладах переважають такі способи організації планування виробничих ресурсів, як «жива черга», талонна система та запис у часову сітку. Дані про пацієнтів зберігаються у паперовому вигляді, це перш за все, вимагає значних людських ресурсів, як на ведення документації так і на пошук необхідної у разі потреби. На ринку ж приватних медичних закладів існує визначений лідер ринку – медична інформаційна система, в якій є можливість управління виробничими ресурсами – «Медучет».

1.1. Управління обліковими ресурсами у державних медичних закладах

У державній медицині історично склалося так, що основним способом організації розподілення ресурсів є «жива черга», тобто по суті планування виробничих ресурсів (робочий персонал, кабінети) зводиться до затвердження графіку прийомів лікарів у певних кабінетах, а управління будь-якими іншими аспектами не відбувається. Крім того, час, який виділено на прийом, формувався за допомогою командно-адміністративних методів та не брав до уваги витрати часу пацієнтів та особливості конкретного закладу. Такий спосіб організації знижує видатки медичного закладу на планування, але збільшує неекономічні видатки пацієнтів (спричиняє очікуванню у черзі у випадку перевантаження лікаря) та не дає можливості управляти тривалістю прийому, часом очікування пацієнтів, аналізувати інформацію по затримкам прийомів, утилізації ресурсів та іншим важливим показникам.

Ведення документації, а саме карток пацієнтів відбувається виключно вручну медичним персоналом. Причиною існування такої системи дотепер є відсутність у медичних закладів потреб відслідковувати такі показники (через відсутність таких вимог з боку системи охорони здоров'я), а також відсутність конкуренції між державними медичними закладами.

Певним удосконаленням системи живої черги стала система талонів на прийом, які видавалися із зазначенням на них певного часу, а отже вирішували дві проблеми: неспроможність лікаря прийняти всіх бажаних пацієнтів (талонів видавалося кожен день обмежена кількість), коли одночасно приходить багато пацієнтів і так чи інакше комусь із пацієнтів доводиться чекати довше за інших (талони видаються із зазначеним на них часом, таким чином більш-менш рівномірно розподіляючи пацієнтів протягом часу прийому). Іншою варіацією талонної системи став запис пацієнта (по прізвищу) у заздалегідь розплановану часову сітку (так зване «простирадло»). Варто сказати, що талонна система перекладала на пацієнта організаційну відповідальність (він повинен був мати із собою талон, як право на прийом), тоді як запис у часову сітку більше навантажував медичний заклад (оскільки інформація про запис мала передаватися лікарю перед прийомом). Такий різновид запису вже був принципово новим рівнем планування у медичних закладах, але також не знімав проблеми черг, оскільки за талонами до «рідких» лікарів все рівно збиралася жива черга, так само як і на запис до таких лікарів. Також такий спосіб не давав змоги варіювати тривалість прийому, закладаючи стандартний часовий інтервал, а також не давав можливості аналізувати завантаження лікарів.

Загалом можна сказати, що у вітчизняних медичних закладах процес управління медичними прийомами поставлений досить слабо, він не є спрямованим на задоволення пацієнтів, і будь-які його удосконалення здебільшого виявляються особистими ініціативами співробітників.

1.2. Аналіз закордонних джерел

Закордонні джерела, які були проаналізовані під час підготовки дипломної роботи, також описують рішення наступних проблем систем ведення обліку пацієнтів:

1. Пацієнти, що потрапляють у заклад без попереднього запису на прийом (walk-ins). З одного боку, такі пацієнти створюють можливість для медичного закладу надати послугу та, окрім отриманої виручки, задовольнити клієнта, який потім повернеться знов. З іншого боку, якщо тримати в наявності вільні ресурси, які могли б використовуватися для обслуговування інших клієнтів, це може суттєво зменшити ефективність використання ресурсів та фінансовий результат.

Найбільш поширений спосіб забезпечити обслуговування таких пацієнтів – резервування частини інтервалів, з неможливістю планування прийомів, яке знімається за декілька годин до початку зарезервованого часу.

В умовах українського приватного медичного закладу, суттєва кількість таких пацієнтів не характерна, та резервування звичайно не проводиться, бо знижує фінансові результати.

2. Пацієнти, що не приходять на прийом (no-shows). Це дуже суттєва проблема, якщо про факт того, що пацієнт не прийде, становиться відомим вже коли час для прийому минув. Клініка у такому разі витрачає ресурси марно та не має можливості скомпенсувати свої витрати.

При одночасній наявності декількох лікарів, які можуть замінювати один іншого (одна спеціальність), відносно невеликої тривалості прийому (до 30 хв.) та суттєвої кількості пацієнтів, які не з'являються, рішенням проблеми може бути надлишковий запис (overbooking), тобто на визначений інтервал створюється більше

записів на прийом, ніж клініка може обслужити (пропорціонально відсотку пацієнтів; що не з'явилися, наприклад, якщо 3 лікаря приймають 9 пацієнтів на годину та відомо, що відсоток пацієнтів, які не з'явилися, становить 15%, клініка може записати на цю годину не 9, а 10 пацієнтів – у середньому обслуговування не буде суттєво страждати, а коефіцієнт утилізації збільшиться.

Іншими заходами для зниження кількості пацієнтів, які не з'являються,

є:

- нагадування про прийом (за SMS, E-Mail або по телефону);
- включення в документи та стандарти обслуговування прохання в разі неможливості відвідати прийом, попередити про це клініку якнайшвидше;
- зв'язок з клієнтом до запланованої події (за 30-60 хвилин) та підтвердження, що він буде присутній; якщо клієнт відмінює прийом, клініка ще має можливість використати цей час;
- попередня оплата чи акції, умовами яких є вчасне прибуття в клініку;
- виявлення найбільш недисциплінованих клієнтів та запис їх виключно на непопулярні проміжки часу або непопулярних лікарів (тобто якщо вони навіть не з'являться, клініка не втрачає);
- використовується у деяких закордонних системах охорони здоров'я – штраф або збільшення вартості страховки для клієнтів, які не з'являються вчасно.

В умовах приватної клініки в Україні ця проблема звичайно не є надто серйозною, особливо для найпопулярніших лікарів, запис до яких завжди завантажений; у більшості випадків використовується попереднє нагадування про прийом.

1.3. Аналіз аналогів, щодо об'єкту проектування

1.3.1. Медична система «Медучет»

Медична система «Медучет» наразі є лідером ринку у приватному сегменті серед медичних інформаційних систем. У програмі реалізована велика кількість функцій, у тому числі і ведення графіків лікарів та запис на прийом у рамках блоку «Графік прийому лікарів».

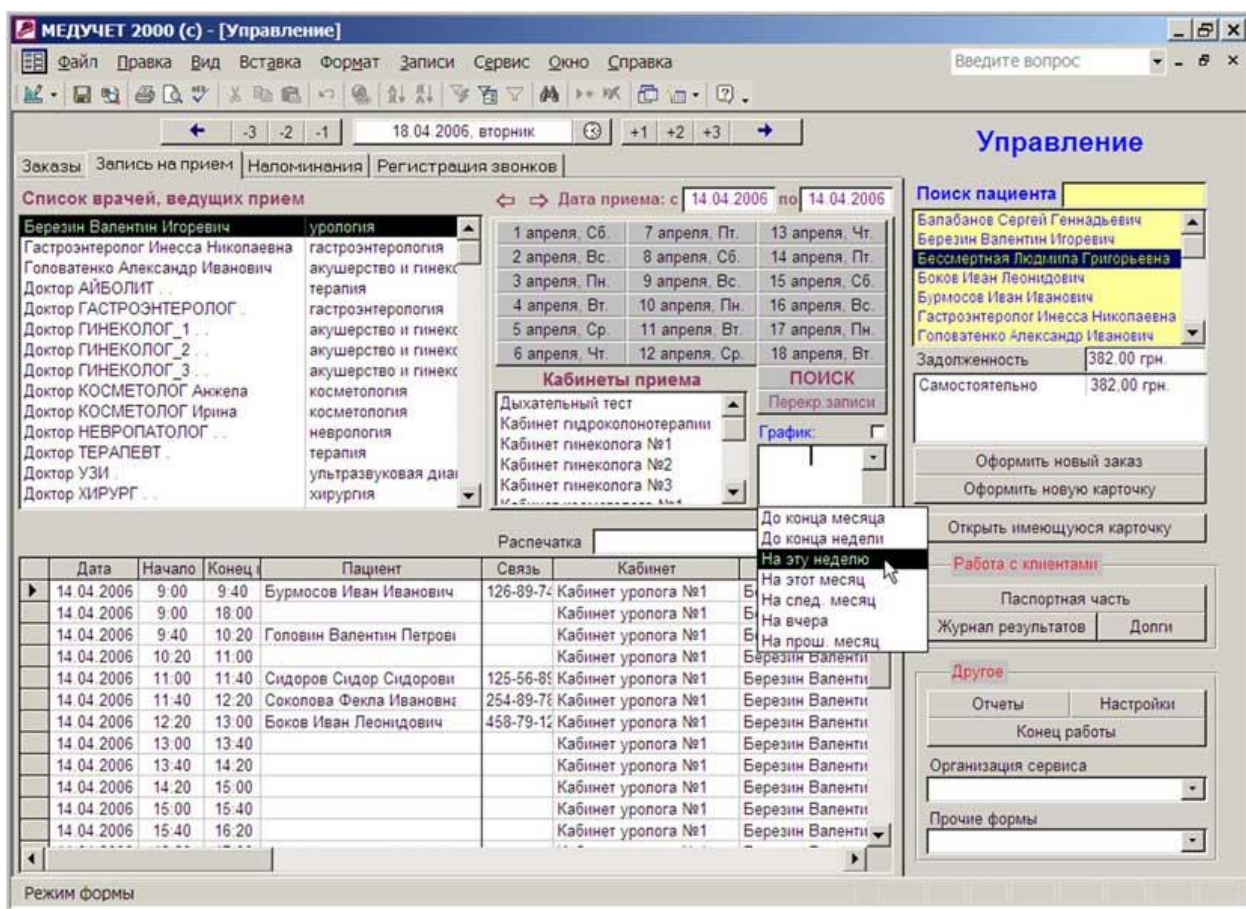


Рис. 1.1. Медична система «Медучет»

Цей блок дозволяє складати графіки прийомів лікарів та записувати пацієнтів до лікарів. Програма відслідковує зайнятість кабінетів та попереджає помилковий запис двох різних пацієнтів в один кабінет на один і той самий час. Також є можливість подивитися вільні інтервали декількох

лікарів одночасно, а також внести пацієнта у лист очікування, якщо у момент запису вільних інтервалів немає. Також є можливість зателефонувати пацієнту (наявні його контактні дані) а також є можливість проводити SMS-розсилку для нагадування про візит. Програма надає звіти по завантаженню кабінетів в різний час дня, порівнює доходність різних кабінетів протягом заданого періоду часу, і оцінює ефективність роботи адміністраторів при попередньому записі пацієнтів.

«Медучет» дозволяє працювати в локальній мережі за технологією «клієнт-сервер», або із використанням віддаленого доступу через мережу Інтернет, а також можливість ведення єдиної бази даних для всіх територіально віддалених підрозділів.

До недоліків системи «Медучет» можна віднести: застаріла технологія для інтерфейсної частини (програма розроблена на Microsoft Access 2003 + VBA), відсутність у ресурсній моделі аналізу ресурсних потреб послуг та варіабельного інтервалу прийому – таким чином, система гарантує тільки незайнятість часу, а всі інші аспекти планування повинна пам'ятати людина, яка записує пацієнтів на прийом.

Робота у розподіленому середовищі із нестабільним інтернет-зв'язком в системі «Медучет» не передбачена (передбачено роботу або в локальній мережі, або через Інтернет-з'єднання із єдиною центральною базою даних). Відповідно, якщо в підрозділі медичного закладу, у якому встановлена система «Медучет», пропаде з'єднання із центральною базою даних, то будь-яка робота із планування ресурсів буде зупинена, що в свою чергу стане причиною незручностей в обслуговуванні пацієнтів.

1.3.2. Медична система «Учет пациентов»

Облік пацієнтів – програма призначена для ведення бази даних пацієнтів, фіксації наданих послуг, обстежень, висновків.

Наявність журналу прийому, генерація офісних документів по шаблонах, потужний експорт і імпорт даних, зберігання картинок в БД, посилення на файли будь-яких форматів. Пошук, фільтри, сортування, групування по будь-якому полю. Можливість гнучко змінювати конфігурацію бази даних, додаючи поля і таблиці настроювання правил кольоровиділення, настройка нагадувань, створення довільних звітів.

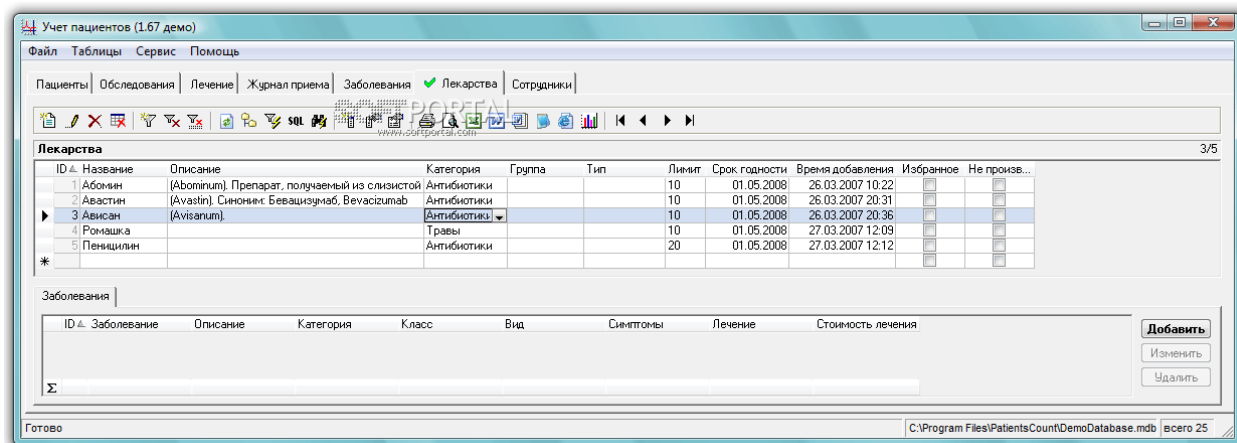


Рис. 1.2. Медична система «Учет пациентов»

Мережевий і багатокористувацький режими, гнучка настройка прав доступу кожному користувачеві і багато інші зручності.

Головним недоліком медичної системи «Учет пациентов» є те, о система не безкоштовна, середня вартість програми становить 2000 грн.

1.3.3. Медична система «АРМ амбулаторный прием»

АРМ амбулаторний прийом – програма призначена для лікарів поліклініки та медичного персоналу. Дозволяє спростити роботу із заповненням всіх необхідних документів.

Програма дозволяє виконувати наступні функції:

- ведення журналу прийому пацієнтів;

- ведення і роздруківка щоденників амбулаторної карти
- планування поліклінічного прийому;
- можливість роботи з диспансерними хворими – планування відвідувань, прийом, швидка постановка на облік, зняття з обліку
 - можливість роздруківки всіляких довідок, як-то: статистичний талон, довідка звернення, направлення на видачу листка непрацездатності, талон на закінчений випадок захворювання, направлення на фізіолікування, напрямок до медичного закладу, направлення на ВК, довідка для отримання путівки, санаторно-курортна карта тощо;
 - використання шаблонів для швидкої підготовки довідок та напрямків;
 - аналіз і уявлення в графічному вигляді результатів діяльності лікаря поліклініки за надання лікарської допомоги в рамках обліково-звітної форми 39 (лікарська відомість);
 - генерацію звітної форми по диспансерним хворим;
 - можливість контролю виконання плану по випадках захворювання, виходячи з їх вартості в залежності від алгоритму оплати.

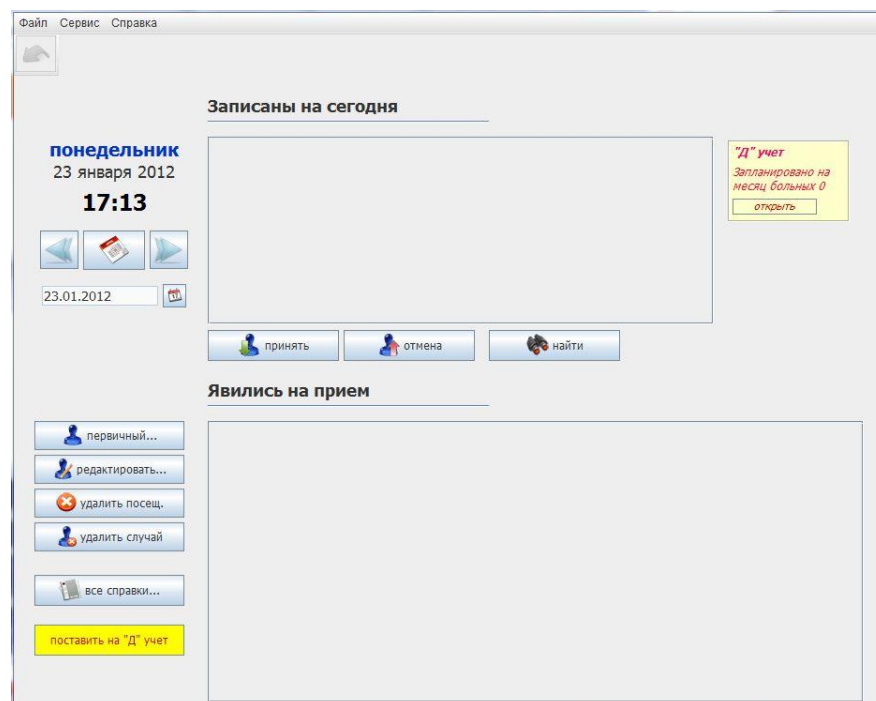


Рис. 1.3. Медична система «АРМ амбулаторный прием»

Головним недоліком даного програмного продукту є вартість, складність налаштування та відсутність підказок при роботі з системою.

Також до недоліків варто віднести обмеженість функцій та відсутність розподіленого доступу.

1.3.4. Порівняльний аналіз аналогів

На основі вищевикладеного аналізу аналогів розробки здійснимо порівняльний аналіз та результати занесемо до таблиці 1.1.

Таблиця 1.1

Порівняльний аналіз аналогів

№	Параметр	«Медучет»	«Учет пациентов»	«PM амбулаторный прием»
1	2	3	4	5
1	Журнал прийому пацієнтів	+	+	+
2	Журнал зайнятості лікарів	+	-	+
3	Планування поліклінічного прийому	+	-	+
4	Функція нагадування	-	+	-
5	Завантаження кабінетів	+	-	-
6	Ведення амбулаторної карти	-	+	+
7	Використання обліково-звітної форми	+	-	+
8	Графічний вивід результатів	-	-	+
9	Наявність Інтернет зв'язку	+	-	-
10	Об'єм займаної пам'яті, кб	990	689	560
11	Вартість, грн	4500	2000	500

1.4. Обґрунтування потреби у новій інформаційній системі

Описане існуюче на ринку України рішення дозволяє управляти виробничими ресурсами у певній мірі, але, як вже було сказано, вона не дає можливості достатньо гнучно планувати виробничі ресурси, зокрема медичні

прийоми, здійснювати автоматичне ведення амбулаторних карток пацієнтів та картотеку. Основні характеристики, яких не вистачає для більш досконалого управління виробничими ресурсами це:

1. Варіабельна тривалість прийому
2. Аналіз ресурсних потреб послуг
3. Здійснення автоматизації ведення документації

Розглянемо кожну з цих характеристик.

Метою існування комерційних медичних закладів є отримання прибутку, як і будь-якої комерційної структури. Через це медичний заклад є зацікавленим, щоб тривалість прийому була не більшою за таку тривалість, яка може забезпечити прийнятний рівень собівартості витрачених ресурсів (час лікаря, кабінету). З іншого боку, через існування конкуренції та наявність певних стандартів надання медичних послуг, медичний заклад є зацікавленим, щоб тривалість прийому була не меншою від такої тривалості, яка може забезпечити достатній рівень медичної якості.

У вітчизняних державних медичних закладах ця проблема вирішувалася дуже просто – в умовах «живої черги» прийом тривав «скільки треба», тобто якщо у пацієнта було мало питань та мало послуг, його прийом тривав недовго, а якщо у пацієнта було багато питань та ще додатково різні процедури, то такий прийом міг затягнутися надовго, що відповідним чином могло відбитися на часі очікування у черзі. Тобто «жива черга» та визначення тривалості прийому по факту робили час очікування погано прогнозованим, навіть при відомій кількості людей у черзі попереду. Так само і система талонів/запис у часову сітку не давала можливості взяти до уваги набір бажаних послуг та маніпуляцій, які пацієнт хоче отримати від лікаря, оскільки тривалість прийому для такої системи також була фіксованою. До речі, у західній медицині, де одним з основних показників є коефіцієнт утилізації ресурсів, тривалість прийому може визначатися із історичної статистичної тривалості подібних прийомів,

враховуючи також варіабельність, яка дозволяє математично визначати баланс між коефіцієнтом утилізації та часом очікування.

Ресурсні потреби послуги – це ресурси, які необхідні для виконання даної послуги – виконавець (лікар чи медсестра), кабінет, певне обладнання (ультразвуковий сканер, рентгенологічний апарат та ін). Коли відбувається планування прийому, важливо переконатися, що пацієнт зможе отримати усі замовлені послуги, а отже необхідно проаналізувати усі послуги на предмет того, ким та де вони можуть виконуватися. Не у кожному кабінеті із зрозумілих причин можна зробити ультразвукову діагностику (треба, щоб сканер був у кабінеті), та не кожен лікар може провести консультацію з певної спеціальності (треба щоб лікар мав цю спеціальність). Інший приклад: не всі дитячі лікарі можуть консультувати дітей перших двох років життя. Наразі в державних та приватних медичних закладах більшу частину ресурсних потреб лікарі-координатори (співробітники реєстратури) тримають в пам'яті або у певних списках. А у випадку великої медичної установи це стає неможливим, і, отже, існує імовірність того, що пацієнта запишуть на прийом, а послуги зможуть бути виконані не всі (через відсутність необхідного обладнання чи кваліфікації лікаря). І навпаки, може статися ситуація, що пацієнта не запишуть на прийом через нібито відсутню можливість надати певні послуги, а насправді така можливість є.

Відповідно, існує потреба в наявності функції реєстрації асоціативних зв'язків між послугами та виконавцем, а також між послугами та кабінетами для забезпечення прозорості та надійності при плануванні прийомів. Після реєстрації цих асоціативних зв'язків необхідно мати механізм для автоматичного підбору доступних виконавців та кабінетів в залежності від набору асоціативних зв'язків у заданого набору послуг.

З цією метою була проаналізована ресурсна модель приватного медичного центра та реалізована можливість автоматично підбирати набір ресурсів, який відповідає ресурсним потребам заданого набору послуг у заданому часовому інтервалі. Ресурсні потреби послуг реєструються

персоналом, обізнаним із деталями того, який набір послуг можна виконати у певному кабінеті, та який набір послуг може виконати певний виконавець (лікар чи медсестра). Як результат, лікарі-координатори взагалі не повинні знати можливості із виконання послуг виконавцями, та деталі кабінетів у клініці, тому що інформаційна система не дасть записати пацієнта до виконавця чи кабінету, із яким не пов'язані усі задані послуги.

Здійснення автоматизації ведення документації. Всі пацієнти, що прибули до приватної клініки допомоги будуть проходити автоматичну реєстрацію через базу даних системи, де їх дані будуть зберігатися на протязі багатьох років. Відповідно формування картки пацієнта буде автоматичним, на основі введення певних симптомів, лікар отримає змогу на екран монітору побачити попередній результат/діагноз, який і визначить подальшу долю пацієнта, його направлення на до обстеження, або розподіл у профільну лікарню.

Вищеописані потреби медичних закладів є наразі актуальними, тож їх реалізація в програмному модулі обліку пацієнтів приватної клініки дозволить медичним закладам функціонувати більш ефективно, знизити кількість помилок та проблем в обслуговуванні та підвищити надійність в питаннях бронювання спільних ресурсів в умовах нестабільного Інтернет-з'єднання.

РОЗДІЛ 2

РОЗРОБКА МЕДИЧНОЇ СИСТЕМИ ОБЛІКУ ПАЦІЄНТІВ ПРИВАТНОЇ КЛІНІКИ

Розробка сучасного засобу обліку пацієнтів приватної клініки ґрунтується на окреслених недоліках виявлених у рамках функціонування аналогів, а також на завданні поставленому у рамках даного магістерського дослідження. Розробка якісного програмного додатку обліку пацієнтів приватної клініки дозволить медичним закладам функціонувати більш ефективно, знизити кількість помилок та проблем в обслуговуванні, а також скоротити людські ресурси та час, що витрачається на ведення документації.

2.1. Постановка завдання

Скласти програму автоматизації обліку пацієнтів приватної клініки.

Кількість пацієнтів не обмежена.

Режими роботи програми:

- 1) введення даних, видалення, редагування;
- 2) індикація результатів (симптоми, ППІ пацієнта, лікаря, поточні симптоми, лікування);
- 3) можливість встановлення двох діагнозів одночасно, за умови однакових симптомів.

Результати повинні зберігатися у БД.

2.2. Програмне середовище розробки та його загальна характеристика

Вибір того чи іншого програмного засобу визначається як специфікою розробки програмного забезпечення та його популярністю, так і фінансовими можливостями розробника.

В даний час найбільш поширеним середовищем є Visual Studio 2013 (мова програмування C++)

Мова і засоби розробки Visual C ++ допомагають у розробці власних програм Магазину Windows, власних додатків для настільного комп'ютера і керованих виразів, які виконуються в середовищі .NET Framework.

На сьогодні мова програмування C++ є досить популярною.

C++ - мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Розроблена Б'ярном Страуструпом (англ. Bjarne Stroustrup) в AT&T Bell Laboratories (Мюррей-Хілл, Нью-Джерсі) у 1979 році та початково отримала назву «Сі з класами». Згодом Страуструп перейменував мову у C++ у 1983 р. Базується на мові C. Визначена стандартом ISO/IEC 14882:2003.

У 1990-х роках C++ стала однією з найуживаніших мов програмування загального призначення. Мову використовують для системного програмування, розробки програмного забезпечення, написання драйверів, потужних серверних та клієнтських програм, а також для розробки розважальних програм таких як відеоігри. C++ суттєво вплинула на інші, популярні сьогодні, мови програмування: C# та Java.

При створенні C++ прагнули зберегти сумісність з мовою C. Більшість програм на C справно працюватимуть і з компілятором C++. C++ має синтаксис, заснований на синтаксисі C (див. список операторів мов C та C++).

Нововведеннями C++ порівняно з C є:

- підтримка об'єктно-орієнтованого програмування через класи;
- підтримка узагальненого програмування через шаблони;
- доповнення до стандартної бібліотеки;
- додаткові типи даних;
- обробка винятків;
- простори імен;
- вбудовані функції;
- перевантаження операторів;
- перевантаження імен функцій;
- посилання і оператори управління вільно розподіленою пам'яттю.

Мова C++ багато в чому є надмножиною C. Нові можливості C++ включають оголошення у вигляді виразів, перетворення типів у вигляді функцій, оператори `new` і `delete`, тип `bool`, посилання, розширене поняття константності та змінності, функції, що підставляються, аргументи за замовчанням, перевизначення, простори імен, класи (включаючи і всі пов'язані з класами можливості, такі як успадкування, функції-члени (методи), віртуальні функції, абстрактні класи і конструктори), перевизначення операторів, шаблони, оператор `::`, обробку винятків, динамічну ідентифікацію і багато що інше. C++ є також мовою строгого типування і накладає більше вимог щодо дотримання типів, порівняно з C.

У C++ з'явилися коментарі у вигляді подвійної косої риски (`«//»`), які були в попереднику C — мові BCPL.

Деякі особливості C++ пізніше були перенесені в C, наприклад ключові слова `const` і `inline`, оголошення в циклах `for` і коментарі в стилі C++ (`«//»`). У пізніших реалізаціях C також були представлені можливості, яких немає в C++, наприклад макроси `vararg` і покращена робота з масивами-параметрами.

2.3. Вибір бази даних

У технології БД існує важлива область, яку не зачіпають офіційні стандарти. Це здатність до взаємодії з іншими БД – методи, за допомогою яких різні БД можуть обмінюватися інформацією. У Windows ця функція реалізується за допомогою протоколу ODBC (відкритий доступ до баз даних). ODBC є стандартом міжплатформного доступу до БД, підтримуваними всіма провідними СУБД [5].

Зростання популярності комп'ютерних мереж справило великий вплив на управління БД і надало SQL нові можливості. У міру поширення мереж додатки, які раніше працювали на центральному міні-комп'ютері або мейнфреймі, переводяться на сервери і робочі станції ЛОМ. У таких мережах SQL пов'язує програми, які працюють на робочій станції, і СУБД, керуючу спільно використовуваними даними на сервері. З появою трирівневої архітектури Internet мова SQL стала сполучною ланкою між керуючим додатком (другий рівень – сервер додатків або Web-сервер) і сервером БД (третій рівень).

Централізована архітектура (рис. 2.1).

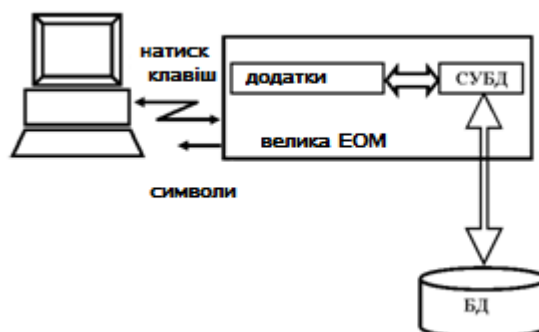


Рис. 2.1. Централізована архітектура

Централізована архітектура використовувалася в початкових СУБД для міні-комп'ютерів. При подібній архітектурі і СУБД, і самі дані розміщуються на центральному міні-комп'ютері або мейнфреймі разом з додатками, які приймають вхідну інформацію з користувачького терміналу і відображають на ньому ж дані. Характерним є те, що при збільшенні числа користувачів зростає навантаження на систему.

Архітектура файл / сервер (рис. 2.2).

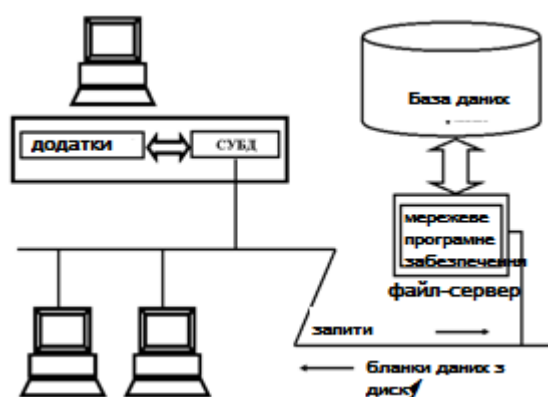
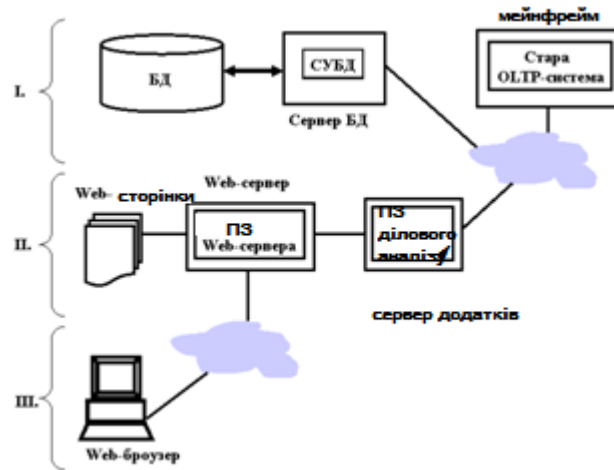


Рис. 2.2. Архітектура файл / сервер

При такій архітектурі додаток, що виконується на ПК, може отримати «прозорий» доступ до файлового сервера, на якому зберігаються спільно використовувані файли. Коли додатку, що працює на ПК, потрібно отримати дані з такого файлу, мережеве програмне забезпечення автоматично зчитує необхідний блок даних з сервера. Ця архітектура підтримувалася першими версіями СУБД. При виконанні звичайних запитів вона забезпечувала прийнятну продуктивність. Однак, при послідовному перегляді БД, СУБД постійно запрошувати нові записи з БД, яка фізично розташована на сервері. У цьому випадку сильно зростає навантаження на мережу, що призводить до зниження продуктивності системи в цілому.

Архітектура клієнт-сервер (розглядалася раніше, у 2-му питанні).

Трирівнева архітектура Internet (рис. 2.3)



I. - Інформаційний рівень

II. - Прикладний рівень

III. - Клієнтський рівень

OLTP – оперативна обробка транзакцій

Рис. 2.3. Трирівнева архітектура Internet

На початку свого розвитку WWW була середовищем перегляду статичних документів і розвивалася незалежно від ринку СУБД. Однак, у міру розвитку виявилось, що це дуже зручний спосіб доступу до корпоративних БД.

Методи зв'язування Web-серверів і СУБД в ході свого розвитку вилилися в трирівневу архітектуру.

Інтерфейсом користувача є Web-браузер, що виконується на ПК. Браузер взаємодіє з Web-сервером, рівень якого можна оцінити як прикладний. Якщо користувач запитує щось більше ніж просто Web-

сторінки, Web-сервер адресує запит серверу додатків, чия роль полягає в аналізі запиту. Запит може включати звернення до успадкованої системи, що виконується на мейнфреймі (OLTP-системи), або до корпоративної БД. Це вже інформаційний рівень. У цій архітектурі SQL закріпився як стандартний засіб взаємодії між другим і третім рівнями.

Кілька років тому для Сховищ Даних було запропоновано використовувати схеми даних, що одержали назви "зірка" і "сніжинка". Суть технології проектування цих схем полягає у виділенні із загального обсягу інформації власне аналізованих даних (або фактів) і допоміжних даних (званих вимірами). Необхідно, однак, віддавати собі звіт в тому, що це призводить до дублювання даних у Сховищі, зниження гнучкості структури і збільшення часу завантаження. Все це – плата за ефективний і зручний доступ до даних, необхідний в СППР [20].

Незважаючи на те що передбачити, яку саме інформацію і в якому вигляді захоче отримати користувач, працюючи з СППР, практично неможливо, вимірювання, за якими проводиться аналіз, досить стабільні. У процесі підготовки того чи іншого рішення користувач аналізує зріз фактів по одному або декільком вимірам. Аналіз інформації, виходячи з понять вимірювань і фактів, іноді називають багатовимірним моделюванням даних (MultiDimensional Modelling, MDM). Таблиці фактів зазвичай містять великі обсяги даних, тоді як таблиці вимірів намагаються зробити поменше. Цього підходу бажано дотримуватися тому, що запит по вибірці з об'єднання таблиць виконується швидше, коли одна велика таблиця об'єднується з декількома малими. При практичній реалізації БД невеликі таблиці вимірів іноді вдається цілком розмістити в оперативній пам'яті, що різко підвищує ефективність виконання запитів.

Оскільки в Сховищах Даних, поряд з детальними, повинні зберігатися і агреговані дані, у випадку "сніжинки" або "зірки" з'являються таблиці агрегованих фактів (агрегатів). Подібно звичайним фактам, агрегати можуть

мати вимір. Крім того, вони повинні бути пов'язані з детальними фактами для забезпечення можливої деталізації. На практиці Сховища часто включають в себе декілька таблиць фактів, пов'язаних між собою вимірами, які таким чином розділяються між декількома таблицями фактів. Така схема носить назву "розширена сніжинка", і саме вона, як правило, зустрічається в сховища даних.

Для досягнення найвищої продуктивності іноді використовують підхід, при якому кожна "зірка" розташовується в окремій базі даних або на окремому сервері. Хоча такий підхід призводить до збільшення розміру дискового простору за рахунок дублювання розділених вимірів, він може виявитися досить корисним при організації Вітрин Даних.

Існує величезна кількість програмних засобів, використання яких забезпечує належний рівень розвитку інформаційних технологій в освіті та науці. Група СУБД представлена на ринку програмних продуктів, дуже широко. Це, наприклад, такі системи, як Paradox, Clarion, FoxBASE. FoxPro, Clipper, Access.

Система управління базами даних (СУБД) - це програмний механізм, призначений для запису, пошуку, сортування, обробки (аналізу) та друкування інформації, що міститься в базі даних.

Найважливішою характеристикою будь-якої СУБД є використовуваний в ній тип транслятора (інтерпретатор чи компілятор). Програми, написані для системи-інтерпретатора, не працюють без наявності самої цієї системи. Безперечною перевагою інтерпретаторів для програмістів є зручність розробки та налагодження програмних продуктів.

СУБД Access має досить високі швидкісні характеристики і входить до складу надзвичайно популярного в нашій країні і за кордоном пакету Microsoft Office.

MS Access - це сама популярна сьогодні настільна система керування базами даних. Але корінь успіху, швидше за все, полягає в прекрасній

реалізації продукту, розрахованого як на починаючого, так і кваліфікованого користувача.

Набір команд і функцій по потужності і гнучкості відповідають більшості сучасних вимог до представлення і обробки даних. В Access підтримується різноманітні спливаючі і багаторівневі меню, робота з вікнами та мишею, реалізовані функції низькорівневого доступу до файлів, управління кольорами, настройки принтера, представлення даних у вигляді електронних таблиць і. т.п. Система також має засоби швидкої генерації екранів, звітів і меню, підтримує мову управління запитамі SQL, має вбудовану мову Visual Basic for Application (VBA), що дозволяє створювати масиви, свої типи даних, викликати DLL-функції, за допомогою OLE Automation контролювати роботу додатків, які можуть функціонувати як OLE-сервери. Можливо навіть цілком створювати бази даних, за допомогою кодування, добре працює в мережі [20].

Головна якість Access, що залучає до нього багатьох користувачів, - тісна інтеграція з Microsoft Office. Наприклад, скопіювавши в буфер графічний образ таблиці, відкривши Microsoft Word і застосувавши вставку з буфера, ми відразу одержимо в документі готову таблицю з даними з БД.

Вся робота з базою даних здійснюється через вікно контейнера бази даних. Звідси здійснюється доступ до всіх об'єктів, а саме: таблицям, запитам, формам, звітам, макросам, модулям. Вони є основними компонентами БД.

Опишемо їх детальніше:

Таблиця - це фундаментальна структура системи управління реляційної бази даних, де зберігаються дані у вигляді записів (рядків) і полів (стовпців). Найчастіше таблиця використовується для зберігання відомостей по одному конкретному питанню.

Запит - питання про дані, що зберігаються в таблицях, або інструкція на відбір записів, що підлягають зміні. Зазвичай вони використовуються для виділення специфічних груп записів, щоб задовольнити певним критерієм.

Форма - об'єкт Microsoft Access, в якому можна розмістити елементи керування, призначені для введення, зображення і зміни даних у полях таблиць. Access дозволяє створювати форми, що включають в себе інші форми, вони називаються підлеглими. Форми дозволяють обчислювати значення і виводити на екран результат.

Звіт дозволяє представляти певну користувачем інформацію в заданому вигляді, переглядати і роздруковувати її.

Макрос - одна або кілька макрокоманд, які можна використовувати для автоматизації конкретної задачі.

Модуль - набір процедур, функцій, оголошень і констант мови Visual Basic For Applications, які управляють базами даних Access [9].

Користувачам, малознайомим із поняттями реляційних баз даних, Access дає можливість розділяти свої складні за структурою таблиці на декілька, пов'язаних по ключовим полям.

Реляційний підхід до побудови бази даних припускає відображення реальних об'єктів у вигляді інформаційних об'єктів чи об'єктів предметної області. Інформаційні об'єкти описують реальні об'єкти за допомогою сукупності взаємопов'язаних реквізитів [10].

Сучасні підходи до проектування реляційних баз даних ґрунтуються на використанні ER-моделі.

Основними поняттями ER-моделі є сутність, зв'язок і атрибут [17].

Сутність - це реальний об'єкт або об'єкт, що представляється, інформація про який повинна зберігатися і бути доступна. У діаграмах ER-моделі сутність представляється у вигляді прямокутника, що містить ім'я сутності. При цьому ім'я сутності - це ім'я типу, а не де якогось конкретного

примірника цього типу. Для більшої виразності і кращого розуміння ім'я суті може супроводжуватися прикладами конкретних об'єктів цього типу (рис.2.4).

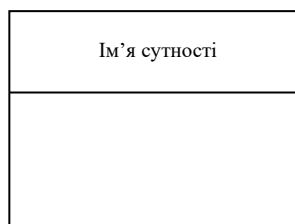


Рис. 2.4. Приклад типу сутності

Примірник сутності - це конкретний представник даної суті. Примірники сутностей повинні бути помітні, тобто сутності повинні мати деякі властивості, унікальні для кожного екземпляра цієї суті.

Безліч з одного або декількох атрибутів, значення яких однозначно визначають кожен екземпляр сутності, називаються ідентифікатором (ключем). Кожна сутність повинна мати хоча б один ідентифікатор. Якщо ідентифікаторів кілька, один з них вибирається як привілейований. Для ідентифікації конкретних екземплярів сутності використовуються атрибути, що виконують роль ідентифікаторів.

Атрибут - поійменована характеристика сутності, що визначає його властивості і приймає одне значення безлічі [20]. Кожному атрибуту присвоюється унікальне ім'я, яке позначає його зміст і значення. Атрибути зображуються у вигляді списку їхніх імен усередині блоку асоційованої сутності (рис.2.5).

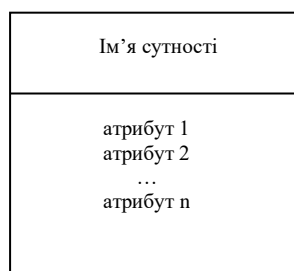


Рис. 2.5. Приклад типу сутності з атрибутами

Ключ суті - це не надлишковий набір атрибутів, значення яких в сукупності є унікальними для кожного екземпляра сутності. Не надмірність полягає в тому, що видалення будь-якого атрибуту з ключа порушує його унікальність. Сутність може мати кілька різних ключів. Ключові атрибути зображуються на діаграмі підкресленням.

Зв'язок - це асоціація, що зображується графічно, та встановлюється між двома сутностями. Ця асоціація завжди є бінарною і може існувати між двома різними сутностями або між сутністю і її самою (рекурсивний зв'язок). У будь-якому зв'язку виділяються два кінці, на кожному з яких вказується ім'я зв'язку [17].

Зв'язок представляється у вигляді лінії, що зв'язує дві сутності або веде від суті до неї ж самої.

Існують три фундаментальних типи зв'язку: один - до - одного (1:

1), один - до - багатьох (1: M), багато - до - багатьох (M: N) [16].

Зв'язок один - до - одного (1:1) існує, коли один примірник однієї сутності пов'язаний з єдиним екземпляром іншої сутності.

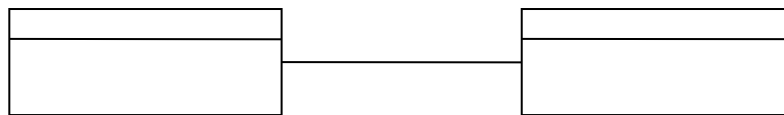


Рис. 2.6. Тип зв'язку "один-до-одного"

Зв'язок один - до - багатьох (1: M) існує, коли один примірник однієї сутності пов'язаний з одним або більше екземпляром іншої сутності і кожен примірник другої суті пов'язаний тільки з одним примірником першої сутності.

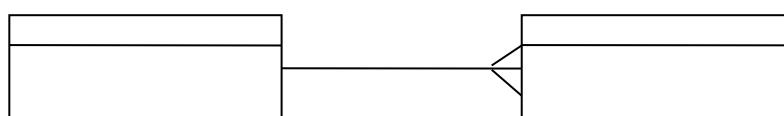


Рис. 2.7. Тип зв'язку "один-до-багатьох"

Зв'язок багато - до - багатьох (M: N) існує, коли один примірник однієї сутності пов'язаний з одним або більше екземпляром іншої сутності і кожен примірник другої суті пов'язаний з одним або більше екземпляром першої сутності.

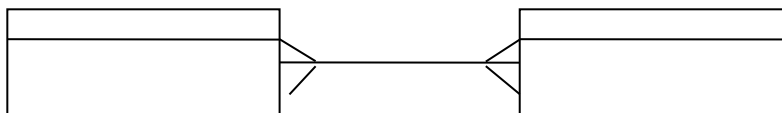


Рис.2.8. Зв'язок "багато-до-багатьох"

Таким чином, Access - це типова настільна база даних. На маленькому підприємстві з невеликою кількістю комп'ютерів, ресурсів Access цілком може вистачити для обслуговування всього діловодства. Тобто всі користувачі можуть звертатися до однієї бази даних, установленної на одній робочій станції, яка не обов'язково повинна бути виділений сервером. Для того щоб не виникали проблеми збереження і доступу до даних можна скористатися засобами захисту, які надає Access.

На відміну від інших розглянутих засобів розробки, СУБД Access має русифікований інтерфейс і частково переведений на російську мову файл контекстної допомоги [4].

Перераховані фактори визначили вибір СУБД Access в якості середовища розробки засобу обліку пацієнтів приватної клініки.

2.4. Функціональні вимоги та вимоги до операційної системи

Програмний продукт розробляється на підставі наступних державних стандартів:

1. ГОСТ 19.401-78. ЕСПД. Текст програми. Вимоги до змісту та оформлення.
2. Міжнародний стандарт ISO/IEC 12207. Інформаційні технології. Процеси життєвого циклу програмного забезпечення.
3. ГОСТ 34.601-90. Автоматизовані системи. Стадії створення.
4. ГОСТ 34.602-89. Інформаційна технологія. Комплекс стандартів на автоматизовані системи. Технічне завдання на створення автоматизованої системи.
5. ГОСТ 34.603-92. Інформаційна технологія. Види випробувань автоматизованих систем.

Програмний продукт, що розробляється в рамках магістерського проекту повинен задовольняти наступному переліку функціональних вимог:

- розподіл доступу користувачів до інформації;
- можливість перегляду доступної інформації;
- оптимізація роботи приватної клініки;
- доступ до необхідної інформації щодо роботи приватної клініки.

Вхідні дані

Вхідними даними при роботі з програмним продуктом повинні бути характер проведених робіт, пацієнти, лікарі, симптоматика. Вхідна мова програми – російська.

Вихідні дані

Вихідними даними при роботі програми є дані, що виводяться на екран користувача це результати здійснення обстеження пацієнтів з попереднім діагнозом. Мова вихідних даних у додатку – російська.

Функціональні вимоги:

1) Вимоги до інтерфейсу програмного засобу. Інтерфейс програмного засобу повинен бути простим, зрозумілим для користувача, інформативним, наявність інформаційних і допоміжних панелей обов'язково. На панель

швидкого доступу повинні бути винесені кнопки, що дублюють пункти меню. Природність інтерфейсу полягає в тому, що він не змушує користувача істотно змінювати звичні для нього способи вирішення завдань. В системі ця властивість реалізована наступним чином: збережені позначення і термінологія предметної області, використовуються знайомі поняття та образи, що забезпечують інтуїтивне розуміння інтерфейсу. Узгодженість інтерфейсу дозволяє користувачеві використовувати раніше набуті знання для з'ясування нових аспектів роботи. Так як в основу підсистеми обліку успішності студентів закладений стандартний користувальницький інтерфейс роботи в середовищі Windows, то проєктована система володіє даними властивостями. Дружній інтерфейс: вивчення будь-якого програмного продукту відбувається методом проб і помилок, тому для цього в підсистемі передбачено попередження таких ситуацій, які можуть викликати помилку. Простота інтерфейсу, не потрібно ускладнювати сприйняття і розуміння інформації, що виникає перед очима користувача. Для цього при виконанні чергового кроку завдання перед користувачем буде відображатися мінімально необхідна інформація. Необхідно також уникати багатослівних командних імен та повідомлень і розміщувати елементи управління на екрані з урахуванням їх смислового значення і логічного взаємозв'язку.

Гнучкість, необхідно враховувати як професійний досвід роботи з комп'ютером, так і продуктивність праці конкретного користувача. Естетична привабливість полягає в тому, щоб на екрані формувалася така середовище, яка допомагала користувачеві зрозуміти надану інформацію, зосереджуватися на ній і, по можливості, зменшувати навантаження на людину (неприпустиме використання яскравих фарб в заголовках меню, різких переміщень, зображення котрі лякають і т. д.).

2) Вимоги до розділення доступу. Розроблюваний програмний засіб має бути доступний за своїми функціональними можливостями кожному користувачеві без обмеження. Обмеження накладаються тільки на доступ до

виконуваного файлу і файлів з даними, доступ до яких визначається системним адміністратором або користувачем комп'ютера.

3) Вимога до програмного забезпечення:

- Операційна система – не нижче MS WINDOWS 2007 / NT / XP;
- Наявність графічних редакторів

4) Вимога до апаратного забезпечення:

- Наявність відеокарти RADEON XFX Radeon R9 280X 3GB DDR5 Black edition (R9-280X_TDBD) або RADEON R7 250 (R7250-OC-2GD3);
- Процесор Intel Core i3-4160 Box, 2x3,6GHz

5) Вимоги до ергономіки та технічної естетики. Програмний засіб повинен бути оптимізований для перегляду при дозволі 1024 * 768, 1280 * 1024 без горизонтальної смуги прокрутки і без порожніх полів для основних типів дозволу. Результатом розробки має стати програмний засіб, який встановлюється на ПЕОМ без будь яких специфічних вимог до апаратної та програмної частини комп'ютера. Зручність і простота у використанні припускають комфортну і продуктивну роботу з даним програмним засобом

б) Вимоги до реактивності системи.

До показників, що описує реактивність системи, відносять всі тимчасові характеристики, які повинні забезпечувати комфортну роботу користувача. При проектуванні системи накладається умова швидкого і точного введення даних, знаходження рішення, швидкого формування звітів по результатам обчислень.

Час формування вихідної документації (розпізнаних образів), включаючи час друку не повинно перевищувати 1 хвилини.

7) Вимоги до захисту інформації. Захист інформації ґрунтується на доступі до робочого місця користувача (персонального комп'ютера) певного кола осіб, які мають права на користування даним комп'ютером.

8) Вимоги до надійності системи. Надійність системи має на увазі надійну роботу ПК, на якому встановлена проєктована інформаційна система.

Вимоги до операційної системи

Для реалізації комп'ютерної системи обліку пацієнтів обрано операційну систему Windows XP, так як програма, яка вирішує дану задачу моделювання, повинна забезпечувати зручний графічний інтерфейс для кращого розуміння моделі.

Операційна система Microsoft Windows XP (від англ. EXPerience - досвід), відома також під кодовим найменуванням Microsoft Codename Whistler. Спочатку в плани корпорації Microsoft входила розробка двох незалежних операційних систем нового покоління. Перший проєкт отримав робочу назву Neptune, ця ОС повинна була стати черговим оновленням Windows Millennium Edition, новою системою лінійки Windows 9X. Другий проєкт, що називався Odyssey, передбачав створення ОС на платформі Windows NT, яка повинна була прийти на зміну Windows 2000[19]. Однак керівництво Microsoft порахувало недоцільним розосереджувати ресурси на просування двох різних ОС, внаслідок чого обидва напрямки розробок були об'єднані в один проєкт - Microsoft Whistler. Можливо, саме завдяки цьому рішення Windows XP поєднує в собі переваги вже знайомих користувачам операційних систем попередніх поколінь: зручність, простоту в інсталяції та експлуатації ОС сімейства Windows 98 і Windows ME, а також надійність і багатофункціональність Windows 2000. В даний час Windows XP для настільних ПК і робочих станцій випускається в трьох модифікаціях: Home Edition для домашніх персональних комп'ютерів, Professional Edition - для офісних ПК і, нарешті, Microsoft Windows XP 64bit Edition - це версія Windows XP Professional для персональних комп'ютерів, зібраних на базі 64-бітного процесора Intel Itanium з тактовою частотою більше 1 ГГц.

Для запуску Microsoft Windows XP необхідний персональний комп'ютер, що відповідає наступним мінімальним системним вимогам: процесор - Pentium-сумісний, тактова частота від 233 МГц і вище; обсяг оперативної пам'яті - 64 Мбайт; вільний дисковий простір - 1,5 Гбайт.

Якщо порівняти Windows XP з більш ранніми версіями Microsoft Windows, в новій операційній системі легко виявити безліч значних відмінностей. Незважаючи на те, що ця ОС була розроблена на основі вже добре знайомої користувачам платформи NT і, на перший погляд, за своїми характеристиками багато в чому схожа з Microsoft Windows 2000, фактично Windows XP відноситься до принципово іншого покоління операційних систем сімейства Windows. Тепер користувач Windows не прив'язаний до якого-небудь стандартного інтерфейсу, що встановлюється в системі за замовчуванням. Якщо не подобається традиційний вид вікон, елементів управління і Панелі завдань, які дісталися новій ОС "у спадок" від Windows 2000, то можна без праці змінити їх, завантаживши з Інтернету будь-який з сотень спеціально розроблених "Тим". Традиційне Головне меню, що відкриває доступ до встановлених на комп'ютері програм, що зберігаються на дисках документів і налаштувань операційної системи, також зазнало ряд значних змін. Тепер при натисканні кнопки Пуск з'являється динамічне меню, що містить значки лише п'яти програм, якими користуються найчастіше. Завдяки цьому можна почати роботу з потрібними додатками значно швидше. Тут же розташовані значки браузера Microsoft Internet Explorer 6 і поштового клієнта Outlook Express 6, кнопки Вихід із системи (Log Off) і Вимкнення комп'ютера (Turn Off Computer).

У середовищі Microsoft Windows користувачеві часто доводиться одночасно працювати з декількома документами або набором різних програм. При цьому неактивні програми згортаються в Панель завдань, внаслідок чого вона рано чи пізно переповнюється значками, і перемикання між завданнями стає скрутним. Для того щоб розвантажити Панель завдань і звільнити більше робочого простору для відображення значків запущених

додатків, в Windows XP використовується так званий алгоритм угрупування завдань, згідно з яким однотипні програми, що працюють на комп'ютері одночасно, об'єднуються в логічну візуальну групу.

До складу Windows XP включений спеціальний механізм - швидке перемикання сеансів (Fast User Switching), із застосуванням якого можна швидко, без реєстрації підключати до роботи з операційною системою нових користувачів і груп користувачів. З'явилася також можливість перемикатися між кількома сеансами роботи без необхідності зберігати дані або перезавантажувати систему. При цьому кожен з користувачів може самостійно змінювати налаштування Windows і працювати з власними файлами і документами, створювати, редагувати та зберігати будь-які дані незалежно від інших користувачів Windows XP. Для кожного нового сеансу роботи операційна система відводить спеціальну ділянку верхньої пам'яті в розмірі 2 Мбайт, проте цей обсяг ніяк не обмежує кількість прикладних програм, які можуть бути запуснені користувачем. Зокрема, механізм Fast User Switching дає можливість користувачеві, який працює, наприклад, з текстовим редактором, ненадовго відлучитися від комп'ютера, а під час його відсутності інший користувач може відкрити власний сеанс Windows і попрацювати в Інтернеті або завантажити гру. При цьому текст, редагований відсутнім користувачем, як і раніше зберігається в пам'яті: повернувшись до комп'ютера, користувач може продовжити роботу з документом з того місця, де вона була перервана, без пере завантаження системи і не запускаючи заново відповідну програму.

2.5. Проектування медичної системи обліку пацієнтів приватної клініки

Ціль розробки програмного продукту реалізувати процес автоматизації діяльності приватної клініки. На виході роботи програмного продукту

користувач отримує діагноз, зведені дані по всіх аналізах та симптомах за певний період.

По результатах проведеного аналізу та беручи до уваги вищезазначені функції, було розроблено дерево цілей системи, що зображене на рис. 2.9. та дерево функцій які має виконувати система рис. 2.10.

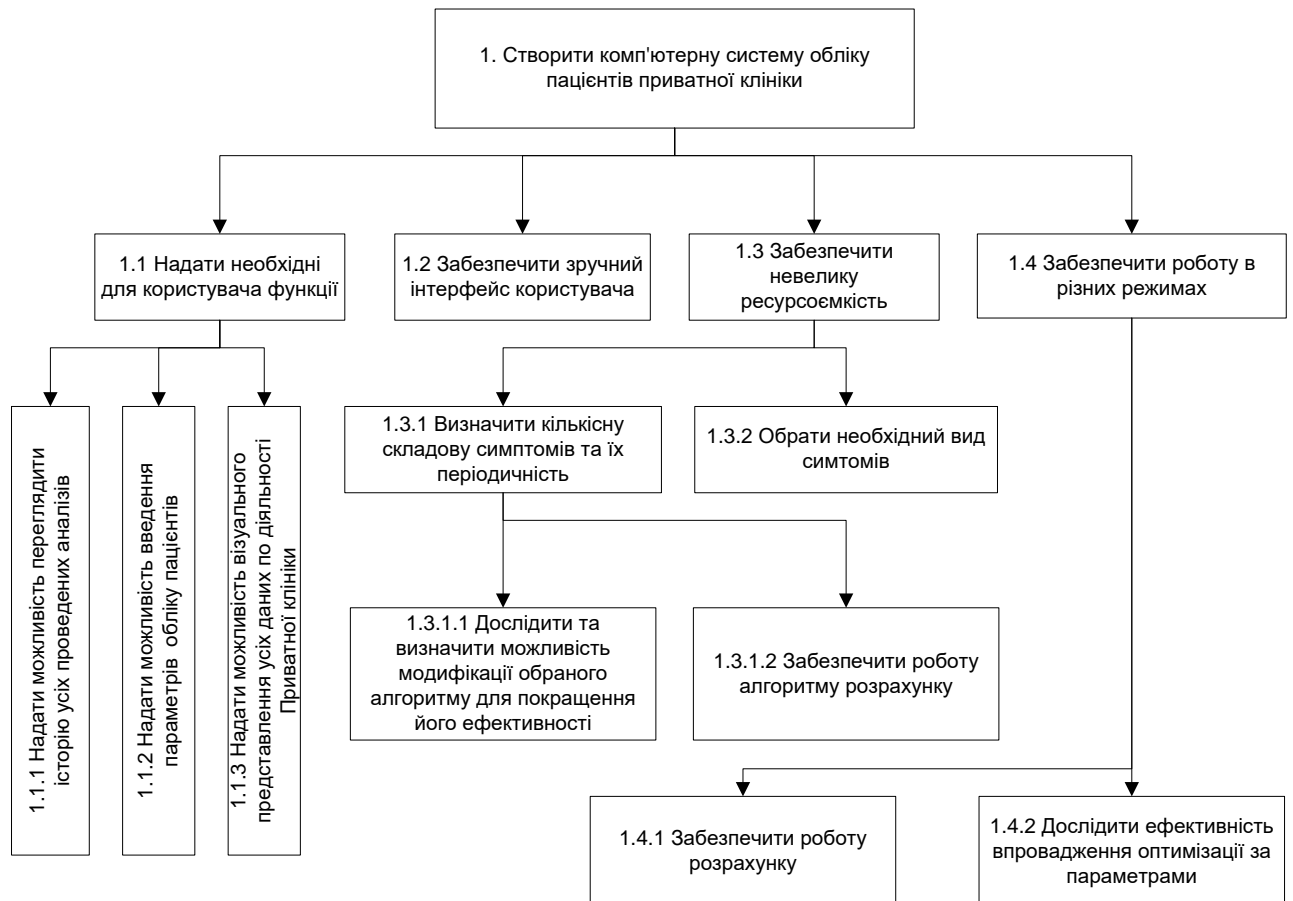


Рис.2.9. Декомпозиція цілей

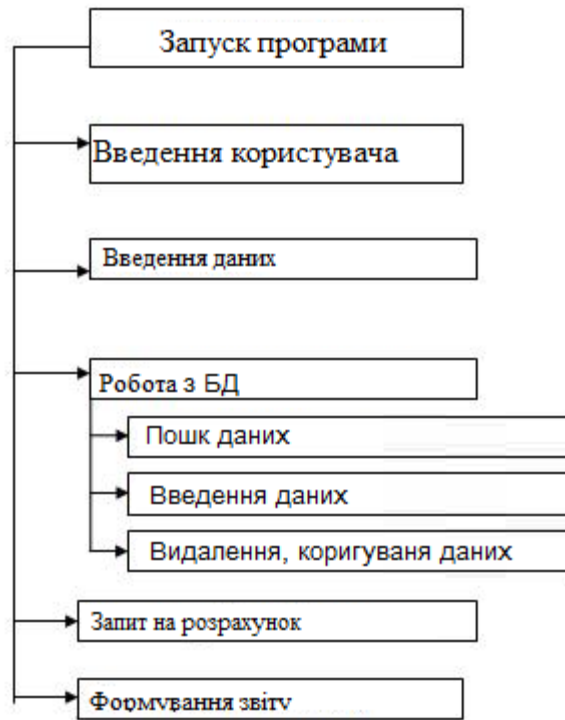


Рис. 2.10. Декомпозиція функцій

Структуру інформаційної системи становить сукупність окремих її частин, званих підсистемами. Підсистема – це частина системи, виділена за будь-якою ознакою [14, с. 32].

Загальну структуру інформаційної системи можна розглядати як сукупність підсистем, незалежно від сфери застосування. У цьому випадку говорять про структурну ознаку класифікації, а підсистеми називають забезпечуючими. Таким чином, структура будь-якої інформаційної системи може бути представлена сукупністю підсистем, що забезпечують її функціонування (рис. 2.11)

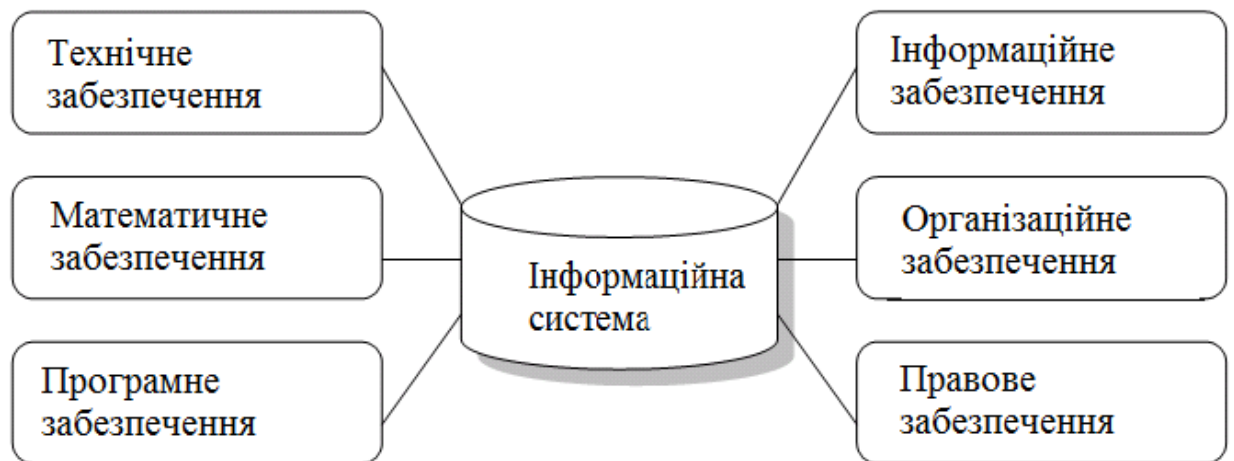


Рис. 2.11. Структура інформаційної системи як сукупність що забезпечують підсистеми [22, с. 23]

Серед підсистем які забезпечують систему зазвичай виділяють інформаційне, технічне, математичне, програмне, організаційне та правове забезпечення.

Призначення підсистеми інформаційного забезпечення полягає у своєчасному формуванні і видачі достовірної інформації для прийняття управлінських рішень.

Інформаційне забезпечення – сукупність єдиної системи класифікації і кодування інформації, уніфікованих систем документації, схем інформаційних потоків, які циркулюють в організації, а також методології побудови баз даних.

Уніфіковані системи документації створюються на державному, республіканському, галузевому та регіональному рівнях [26, с. 54]. Головна мета – це забезпечення порівнянності показників різних сфер суспільного виробництва. Розроблені стандарти, де встановлюються вимоги:

- до уніфікованих систем документації;
- до уніфікованих форм документів різних рівнів управління;

- до складу та структури реквізитів і показників;
- до порядку впровадження, ведення та реєстрації уніфікованих форм документів;

Однак, незважаючи на існування уніфікованої системи документації, при обстеженні більшості організацій постійно виявляється цілий комплекс типових недоліків:

- надзвичайно великий обсяг документів для ручної обробки;
- одні і ті ж показники часто дублюються в різних документах;
- робота з великою кількістю документів відволікає від фахівців рішення безпосередніх завдань;

Є показники, які створюються, але не використовуються, та ін.

Тому усунення зазначених недоліків є одним із завдань, що стоять при створенні інформаційного забезпечення.

Схеми інформаційних потоків відображають маршрути руху інформації та її обсяги, місця виникнення первинної інформації й використання результативної інформації. За рахунок аналізу структури подібних схем можна виробити заходи щодо вдосконалення системи управління. Як приклад простої схеми потоків даних можна привести схему, де відображені всі етапи проходження службової записки або запису в базі даних про наслідки повені.

Побудова схем інформаційних потоків, що дозволяють виявити обсяги інформації і провести її детальний аналіз, забезпечує:

- виняток дублюючої і невживаної інформації;
- класифікацію та раціональне подання інформації.

При цьому докладно повинні розглядатися питання взаємозв'язку руху інформації по рівнях управління. Слід виявити, які показники необхідні для прийняття управлінських рішень, а які ні. До кожного виконавця повинна надходити тільки та інформація, яка використовується.

Для створення інформаційного забезпечення необхідно:

- чітке розуміння цілей, завдань, функцій всієї системи;
- виявлення руху інформації від моменту виникнення і до її використання на різних рівнях, поданої для аналізу у вигляді схем інформаційних потоків;
- оптимізація роботи приватної клініки;
- наявність і використання системи класифікації та кодування;
- володіння методологією створення концептуальних інформаційно-логічних моделей, що відображають взаємозв'язок інформації;
- створення масивів інформації на машинних носіях, що вимагає наявності сучасного технічного забезпечення.

Технічне забезпечення – комплекс технічних засобів, призначених для роботи інформаційної системи, а також відповідна документація на ці кошти і технологічні процеси.

Комплекс технічних засобів складають:

- комп'ютери будь-яких моделей;
- пристрої збору, накопичення, обробки, передачі та виведення інформації;
- пристрої передачі даних і ліній зв'язку;
- оргтехніка та пристрої автоматичного знімання інформації;
- експлуатаційні матеріали і ін.

Документацією оформляються попередній вибір технічних засобів, організація їх експлуатації, технологічний процес обробки даних, технологічне оснащення. Документацію можна умовно розділити на три групи:

1. загальносистемну, що включає державні й галузеві стандарти з технічного забезпечення
2. спеціалізовану, що містить комплекс методик по всіх етапах розробки технічного забезпечення

3. нормативно-довідкову, використовувану при виконанні розрахунків з технічного забезпечення

До теперішнього часу склалися дві основні форми організації технічного забезпечення (форми використання технічних засобів): централізована і частково або повністю децентралізована.

Централізоване технічне забезпечення базується на використанні в інформаційній системі великих ЕОМ та обчислювальних центрів.

Децентралізація технічних засобів передбачає реалізацію функціональних підсистем на персональних комп'ютерах безпосередньо на робочих місцях.

Перспективним підходом слід вважати, вочевидь, частково децентралізований підхід – організацію технічного забезпечення на базі розподілених мереж, що складаються з персональних комп'ютерів і великий ЕОМ для зберігання баз даних, загальних для будь-яких функціональних підсистем.

Математичне та програмне забезпечення – сукупність математичних методів, моделей, алгоритмів і програм для реалізації цілей і завдань інформаційної системи, а також нормального функціонування комплексу технічних засобів [22, с. 56].

До засобів математичного забезпечення відносяться:

- засоби моделювання процесів управління;
- типові завдання управління;
- методи математичного програмування, математичної статистики, теорії масового обслуговування та ін.

До складу програмного забезпечення входять загальносистемні і спеціальні програмні продукти, а також технічна документація.

До загальносистемного програмного забезпечення відносяться комплекси програм, орієнтованих на користувачів і призначених для

вирішення типових задач обробки інформації. Вони служать для розширення функціональних можливостей комп'ютерів, контролю і управління процесом обробки даних.

Спеціальне програмне забезпечення являє собою сукупність програм, розроблених при створенні конкретної інформаційної системи. У його склад входять пакети прикладних програм (ППП), які реалізують розроблені моделі різного ступеня адекватності, що відображають реального функціонування об'єкта.

Технічна документація на розробку програмних засобів повинна містити опис задач, завдання на алгоритмізацію, економіко-математичну модель задачі, контрольні приклади.

Організаційне забезпечення - сукупність методів і засобів, що регламентують взаємодію працівників з технічними засобами і між собою в процесі розробки та експлуатації інформаційної системи.

Організаційне забезпечення реалізує наступні функції:

1. аналіз існуючої системи управління організацією, де буде використовуватися ІС, і виявлення завдань, що підлягають автоматизації;
2. підготовку до вирішення завдань на комп'ютері, включаючи технічне завдання на проектування ІВ і техніко-економічного обґрунтування ефективності ІВ;
3. розробку управлінських рішень щодо складу та структури організації, методології вирішення завдань, спрямованих на підвищення ефективності системи управління.

Правове забезпечення – сукупність правових норм, що визначають створення, юридичний статус і функціонування інформаційних систем, що регламентують порядок отримання, перетворення і використання інформації.

Головною метою правового забезпечення є зміцнення законності. До складу правового забезпечення входять закони, укази, постанови державних органів влади, накази, інструкції та інші нормативні документи міністерств,

відомств, організацій, місцевих органів влади. У правовому забезпеченні можна виділити загальну частину, регулюючу функціонування будь-якої інформаційної системи, і локальну частину, регулюючу функціонування конкретної системи.

Правове забезпечення етапів розробки інформаційної системи включає нормативні акти, пов'язані з договірними відносинами розробника і замовника і правовим регулюванням відхилень від договору.

Правове забезпечення етапів функціонування інформаційної системи включає:

- статус інформаційної системи;
- права, обов'язки і відповідальність персоналу;
- правові положення окремих видів процесу управління;
- порядок створення і використання інформації та ін.

Візуальне моделювання в UML можна представити як певний процес по рівневого спуску від найбільш загальної і абстрактної концептуальної моделі вихідної системи до логічної, а потім і до фізичної моделі відповідної програмної системи. Для досягнення цих цілей спочатку будується модель у формі так званої діаграми варіантів використання (use case diagram), яка описує функціональне призначення системи або, іншими словами, те, що система буде робити в процесі свого функціонування. Діаграма варіантів використання є вихідним концептуальним уявленням або концептуальною моделлю системи в процесі її проектування і розробки.

Розробка діаграми варіантів використання переслідує мети:

Визначити загальні межі і контекст модельованої предметної області на початкових етапах проектування системи.

Сформулювати загальні вимоги до функціональної поведінки проектованої системи.

Розробити вихідну концептуальну модель системи для її подальшої деталізації у формі логічних і фізичних моделей.

Підготувати вихідну документацію для взаємодії розробників системи з її замовниками і користувачами.

Суть даної діаграми полягає в наступному: проектована система представляється у вигляді безлічі сутностей або акторів, що взаємодіють з системою за допомогою так званих варіантів використання. При цьому актором (actor) або дійовою особою називається будь-яка сутність, що взаємодіє з системою ззовні. Це може бути людина, технічний пристрій, програма або будь-яка інша система, яка може служити джерелом впливу на модельовану систему так, як визначить сам розробник. У свою чергу, варіант використання (use case) служить для опису сервісів, які система надає акторові. Іншими словами, кожен варіант використання визначає деякий набір дій, який чинять системою при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізовано взаємодія акторів з системою.

Побудуємо діаграму варіантів використання програмного продукту (рис. 2.12), що розробляється.

Основою програми є облік діяльності приватної клініки, тобто користувач буде використовувати даний програмний продукт за для автоматизації роботи приватної клініки, ведення оглядів та бази пацієнтів.

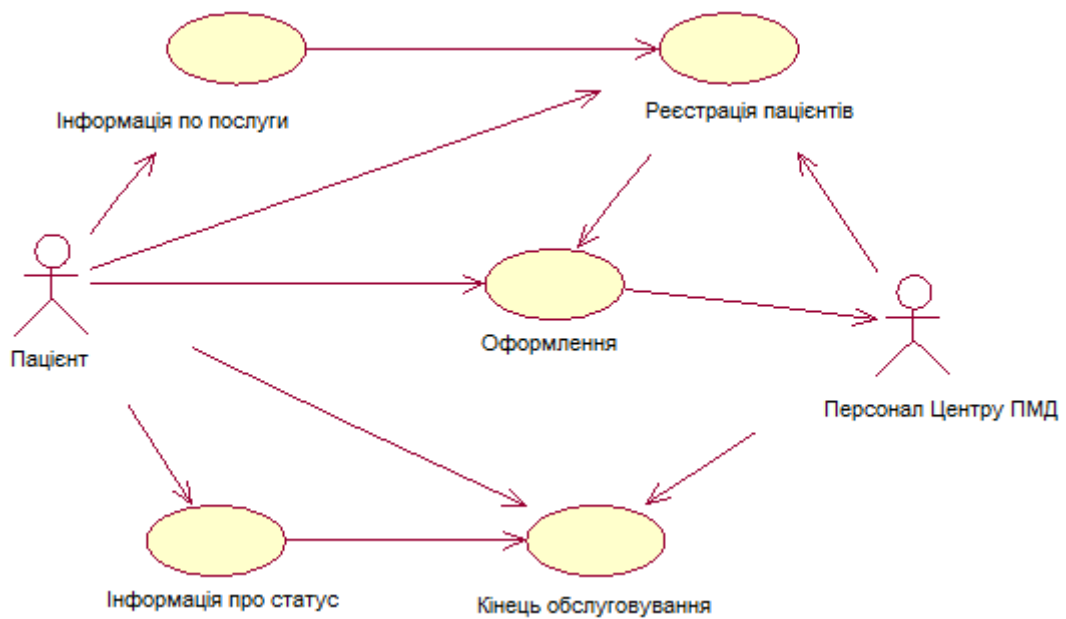


Рис. 2.12. Діаграма варіантів використання

Програмою передбачено:

- формування вихідних даних на розрахунок;
- визначення попереднього діагнозу;
- формування результатів.

Загальна структура ІС зображена на Рис.1.13 та складається з дев'яти модулів, на які вона поділяється за функціональними ознаками:

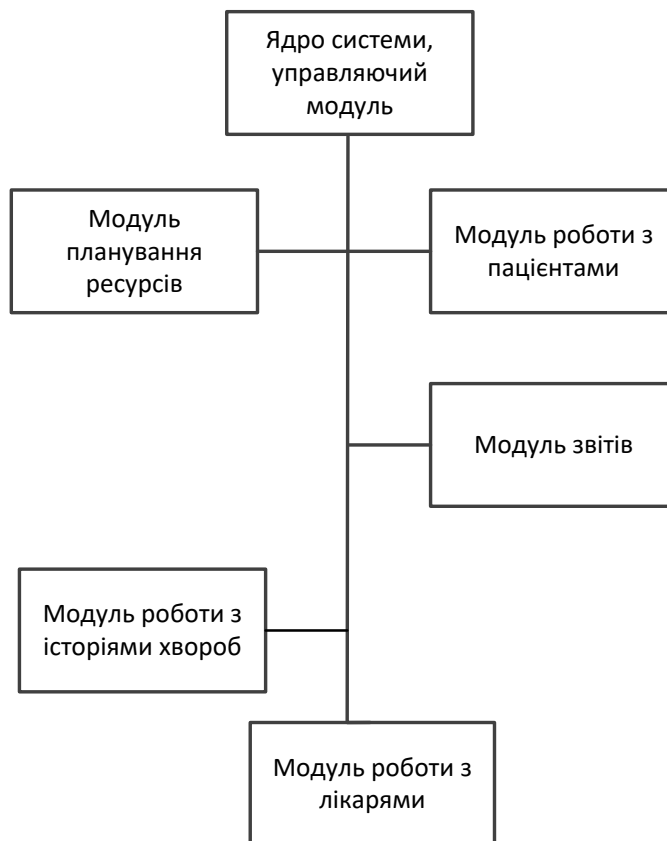


Рис.1.13. Структурна схема медичної інформаційної системи

Управляючий модуль – складає ядро системи та забезпечує взаємодію всіх її частин

Модуль планування ресурсів – забезпечує управління ресурсними потребами послуг, а також планування та перепланування медичних прийомів

Модуль роботи з пацієнтами – забезпечує роботу з паспортними картками пацієнтів

Модуль звітів – відповідає за роботу з управлінськими, оперативними, фінансовими, маркетинговими та іншими звітами.

Модуль роботи з історіями хвороб – дозволяє вести електронну медичну документацію

Модуль роботи з лікарями – забезпечує роботу лікарів

Задачі програмного модуля управління виробничими ресурсами є наступними:

Забезпечення процесу зручного планування та перепланування медичних прийомів пацієнтів. Для вирішення цієї задачі розроблено інтерфейс користувача, який прозоро та інтуїтивно дозволяє вести облік пацієнтів приватної клініки.

З точки зору предметної області список сутностей ПМУВР є наступним:

Прийом – центральне поняття планування та спосіб зарезервувати певну комбінацію виробничих ресурсів на певний проміжок часу. Прийом має посилання на пацієнта, який записується, виконавця прийому, кабінет та одну и декілька послуг, які мають бути виконані в процесі прийому. Ресурсами прийому, які підлягають бронюванню, є кабінет та виконавець, до того ж певним ресурсом вважається пацієнт, бо він теж не може бути записаний на два різні прийоми одночасно.

Пацієнт – сутність, яка відображає клієнтів клініки. В розрізі планування виробничих ресурсів пацієнт також може вважатися умовним ресурсом, оскільки запис на два різні прийоми одночасно є недоступним, а значить доступність пацієнта при записі його на прийом теж треба відслідковувати.

Послуга – сутність, яка відображає словник всіх можливих послуг. Важливими характеристиками послуги є два параметра тривалості послуг, з яких вираховується загальна тривалість послуги.

Ресурсні потреби послуг – сутність, яка відображає які виконавці можуть виконувати кожну послугу, а також у якому кабінеті можна виконувати кожну послугу. Тобто ресурсні потреби – це набір асоціацій між послугами та відповідними потребами у ресурсах цих послуг.

Екземпляр послуги – це є конкретно запланована послуга, яка є частиною прийому.

Виконавець – сутність, яка відображає ресурс (лікаря чи медсестру), який виконує послуги прийому.

Графік роботи – сутність, яка відображає робочий графік виконавців прийому.

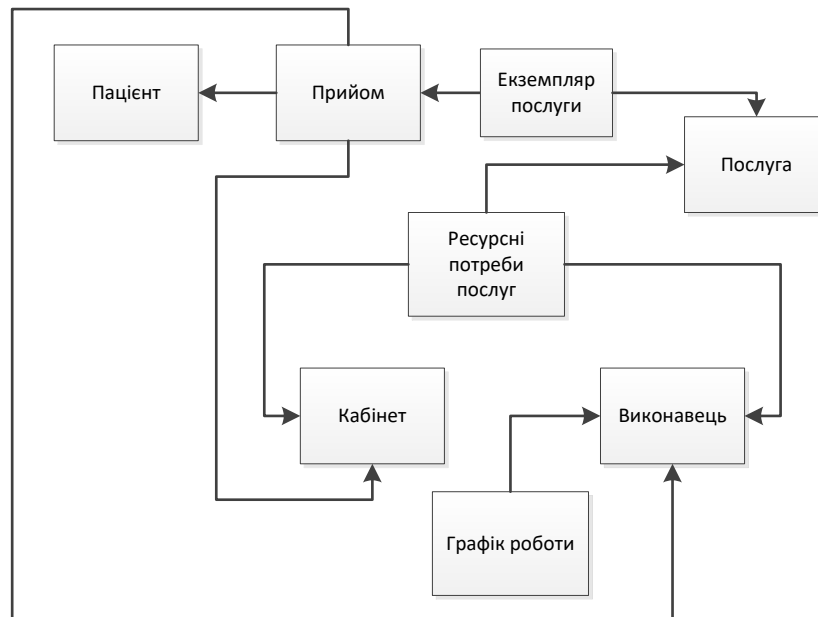


Рис. 2.14. Структура сутностей та їх зв'язків модуля управління виробничими ресурсами

Структура сутностей та зв'язки між ними зображені на Рис. . Напрямок стрілок відображає спрямованість посилань: наприклад прийом має посилання на пацієнта та лікаря і т.д.

Пакет (пакет) - це інструмент групування, який дозволяє взяти будь-яку конструкцію UML і об'єднати її елементи в одиниці високого рівня. В основному пакети служать для об'єднання класів у групи.

У моделі UML кожен клас може включатися тільки в один пакет. Пакети можуть також входити до складу інших пакетів, тому залишаємося в ієрархічній структурі, в якій пакети верхнього рівня

розпадаються на підпакекти зі своїми власними підпакетами, і так далі, до самого низу ієрархії класів. Пакет може містити й підпакекти, і класи.

У термінах програмування пакети в UML відповідають таким групуючим конструкціям, як пакети в Java і простору імен в C++ і .NET.

Кожен пакет являє простір імен (імен), а це означає, що кожен клас всередині власного пакета повинен мати унікальне ім'я. Щоб відрізнити один клас від іншого, можна використовувати повністю певне ім'я (повне ім'я), тобто ім'я, яке вказує на структуру, що володіє пакетом. У мові UML в іменах пакетів використовуються подвійні двокрапки, тому класи даної системи можуть мати імена Система :: DatenMartinFowler :: Util :: Дата.

На діаграмах пакети зображуються у вигляді папок із закладками. Можна показувати тільки ім'я пакета або ім'я разом з його вмістом. У будь-якому випадку можна використовувати або повністю певні імена, або звичайні імена. Зображення вмісту за допомогою значків класів дозволяє показати всі особливості класу, аж до зображення діаграми класів всередині пакета.

Цілоком можна зустріти клас, наприклад з ім'ям Дата (як у java.util), а не з повністю певним ім'ям. Цей стиль є угодою, в основному прийнятим в Rational Rose, яке не входить в стандарт мови.

UML дозволяє класам в пакетах бути відкритими (публічне) або закритими (приватне). Відкриті класи представляють частину інтерфейсу пакета і можуть бути використані класами з інших пакетів; закриті класи недоступні ззовні. У різних середовищах програмування діють різні правила щодо видимості їх конструкцій; необхідно дотримуватися правил свого програмного оточення, навіть якщо це йде врозріз з правилами UML.

У таких випадках корисно скоротити інтерфейс пакета, експортуючи тільки невелику підмножину операцій, пов'язаних з відкритими класами пакета. Можна зробити це, присвоївши всіх класах модифікатор видимості приватні (закритий), так щоб вони були доступні тільки класам даного

паketу, а також створивши додаткові відкриті класи для зовнішнього використання.

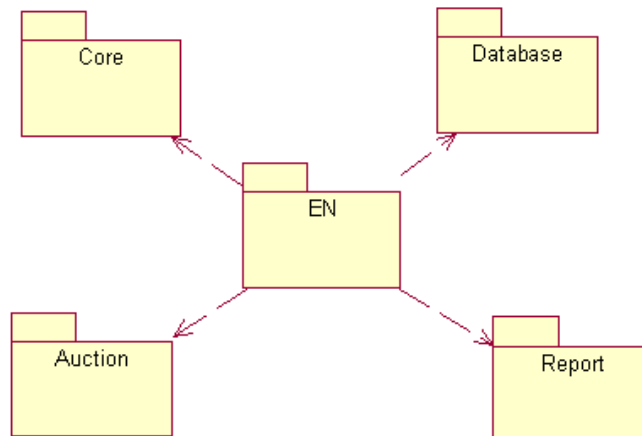


Рис. 2.15. Діаграма пакетів

Діаграма розгортання призначена для візуалізації елементів і компонентів програми, існуючих лише на етапі її виконання (runtime). При цьому подаються тільки компоненти-екземпляри програми, що є здійсними файлами або динамічними бібліотеками. Ті компоненти, які не використовуються на етапі виконання, на діаграмі розгортання не показуються. Так, компоненти з вихідними текстами програм можуть бути присутніми тільки на діаграмі компонентів. На діаграмі розгортання вони не вказуються.

Діаграма розгортання містить графічні зображення процесорів, пристроїв, процесів і зв'язків між ними. На відміну від діаграм логічного представлення, діаграма розгортання є єдиною для системи в цілому, оскільки повинна цілком відбивати особливості її реалізації. Ця діаграма, по суті, завершує процес ООАП для конкретної програмної системи та її розробка, як правило, є останнім етапом специфікації моделі.

Отже, перерахуємо цілі, переслідувані при розробці діаграми розгортання:

- Визначити розподіл компонентів системи по її фізичним вузлам.
- Показати фізичні зв'язки між усіма вузлами реалізації системи на етапі її виконання.
- Виявити вузькі місця системи і реконфігурувати її топологію для досягнення необхідної продуктивності.

Для забезпечення цих вимог діаграма розгортання розробляється спільно системними аналітиками, мережевими інженерами і системотехніками.

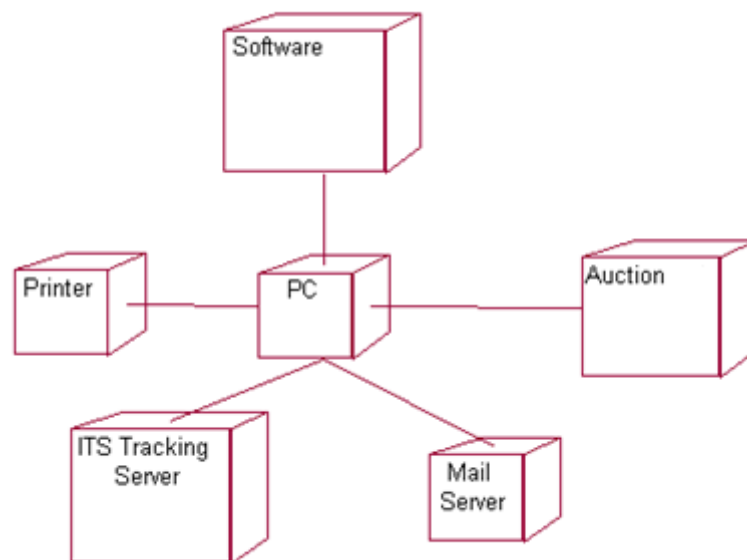


Рис. 2.16. Діаграма розгортання на одній машині

Наступним розділом проведемо практичну реалізацію результатів дослідження. Протестуємо розроблену систему обліку пацієнтів приватної клініки.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

Практична реалізація результатів дослідження ґрунтується на детальній розробці медичної системи обліку пацієнтів приватної клініки. Перелік режимів та структура діалогу розробленого програмного продукту визначає основні режими роботи та розкриває інтерфейс додатку.

3.1. Перелік та призначення режимів та структура діалогу

Програма працює у трьох режимах:

- ввід основних особистих даних пацієнтів та лікарів. Характеристика основних симптомів вичавлених у пацієнта. Після чого обраховується діагноз, сортуються дані і заносяться у базу даних;
- доповнення даних поточними симптомами, які можуть бути виявлені під час обстеження, після чого знову обчислюється діагноз, додається до попереднього і знову сортуються дані і записуються у базу даних;
- вивід результатів (особисті дані пацієнта, лікаря, поточні симптоми, діагноз).

Робота з програмою відбувається через зручний інтерфейс представлений вкладеннями:

- симптоми;
- пацієнти;
- лікарі;
- захворювання.

Також кожне окреме вкладення, має на головній сторінці свою анель управління.

Всі введені дані користувачем записуються у базу даних представлену окремим файлом, там всі дані і зберігаються. БД обрано Microsoft Office Access.

3.2. Характеристика та структура даних з якими працює програма

Дана програма працює з такими даними як структури, БД.

Структури це сукупність різнорідних даних, що описуються і обробляються як одне ціле. Структури використовують для зберігання комбінованої інформації, яка може складатися з чисел, символів, рядків, масивів [1].

База даних (англ. database) – це сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між їх елементами; ця сукупність підтримує щонайменше одну з областей застосування (за стандартом ISO/IEC 2382:2015). В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти. Дані у базі організовують відповідно до моделі організації даних. Таким чином, сучасна база даних, крім саме даних, містить їх опис та може містити засоби для їх обробки [2].

В загальному випадку базою даних можна вважати будь-який впорядкований набір даних. Наприклад, паперову картотеку з формулярами про працівників підприємства у відділі кадрів. Але дана стаття зосереджена на використанні баз даних в інформаційних системх. На даний час застосунки для роботи з базами даних є одними з найпоширеніших прикладних програм.

У сучасних інформаційних системах для забезпечення роботи з базами даних використовують системи керування базами даних (СКБД). Система

керування базами даних – це система, основана на програмних та технічних засобах, яка забезпечує визначення, створення, маніпулювання, контроль, керування та використання баз даних (за стандартом ISO/IEC 2382:2015). Застосунки для роботи з базою даних можуть бути частиною СКБД або автономними. Найпопулярнішими СКБД є MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird та IBM DB2. СКБД дозволяють ефективно працювати з базами даних, обсяг яких робить неможливим їх ручне опрацювання.

Через тісний зв'язок баз даних з СКБД під терміном «база даних» інколи необґрунтовано та неточно мають на увазі систему керування базами даних. Але варто розрізняти базу даних – сховище даних, та СКБД – засоби для роботи з базою даних. СКБД з інформаційної системи може бути видалена, але база даних продовжить існувати. І навпаки: СКБД може функціонувати без жодної бази даних [3].

В загальному базу даних неможливо просто перемістити з однієї СКБД до іншої. Але СКБД використовують стандарти (SQL, ODBC, JDBC), які уніфікують ряд операцій по роботі з даними і дозволяють різним застосункам працювати з базами даних різних СКБД. СКБД часто класифікують за моделлю організації даних. Найживаніші СКБД використовують реляційну модель, у якій дані подають у виді таблиць. Для кінцевого користувача (та прикладних програм) робота з базою даних напряму неможлива. Всі маніпуляції над даними здійснюють через спеціальні запити, які надсилають до СКБД. СКБД опрацьовує їх і повертає результат. Безпосередньо з базою даних працює виключно СКБД.

Головні таблиці БД наведено на рис. 3.1 – 3.7.

id	id_patient	id_doctor	AppealDate
2	Иванов	Кохан	14.03.2021
3	Иванов	Кохан	21.08.2016
4	Сидоров	Лебедев	10.01.2021
5	Иванов	Кохан	14.03.2020
7	Петров	Лебедев	#####
(№)			

Рис. 3.1. Таблица Звернення

id	id_illness	id_appeal	Добавить поле
1	аллергия	Иванов	
3	ангина	Сидоров	
4	аллергия	Иванов	
(№)			

Рис. 3.2. Таблица Діагнози

id	LastName	FirstName	SurName	Доб
1	Кохан	Иван	Иванович	
2	Лебедев	Сергей	Сергеевич	
3	Никифоров	Станислав	Львович	
(№)				

Рис. 3.3. Таблица Лікарі

id	Title	HealthGroup	Notes
1	пневмония	1	крупозная
2	гипертония	2	
4	эпилепсия	1	с детства
5	ангина	2	гнойная
6	аллергия	2	на цветение
7	артрит	2	
8	бронхиальная	1	
*	(№)		

Рис. 3.4. Таблица Захворювания

id	LastName	FirstName	SurName	Address	Tel
1	Иванов	Иван	Иванович	Ул.Радио, д.55	6455923
2	Сидоров	Сидор	Сидорович	ул. Ракетная, 1	7682234
3	Петров	Петр	Петрович	ул. Пакетная 4	8360233
*	(№)				

Рис.3.5. Таблица Пациенти

id	Title
1	температура
2	диарея
3	боль в животе
4	головная боль
5	хроническая у
6	боль в горле
7	язвы в горле
8	высыпания на
9	зуд на теле
10	боль в суставах
11	бледность ко
*	(№)

Рис. 3.6. Таблица Симптоми

id	id_appeal	id_symptom
1	Иванов	диарея
3	Иванов	хроническая усталость
4	Сидоров	высыпания на теле
5	Иванов	высыпания на теле
7	Петров	высыпания на теле
*	(№)	

Рис. 3.7. Таблица Симптоми по звернениям

Запити до БД наведено на рис. 3.8-3.9.

LastName	FirstName	SurName	AppealDate	Title	Appeals.id	Diagnoses.id	id_patient	id_doctor
Иванов	Иван	Иванович	21.08.2016	аллергия	2	1	Иванов	Кохан
Сидоров	Сидор	Сидорович	19.08.2016	ангина	4	3	Сидоров	Лебедев
Иванов	Иван	Иванович	21.08.2016	аллергия	3	4	Иванов	Кохан
*					(№)	(№)		

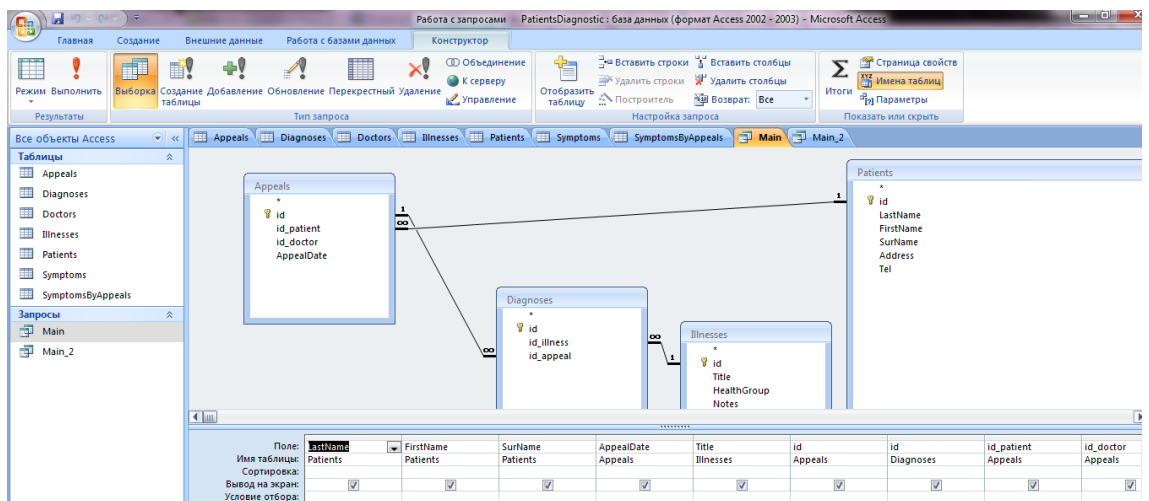


Рис. 3.8. Запит

LastName	FirstName	SurName	AppealDate	Title	id_patient	id_doctor	Doctor
Иванов	Иван	Иванович	17.08.2016		Иванов	Кохан	Кохан Иван И
Иванов	Иван	Иванович	21.08.2016	аллергия	Иванов	Кохан	Кохан Иван И
Петров	Петр	Петрович	#####		Петров	Лебедев	Лебедев Серг
Сидоров	Сидор	Сидорович	19.08.2016	ангина	Сидоров	Лебедев	Лебедев Серг

Field	Table
LastName	Q
FirstName	Q
SurName	Q
AppealDate	Q
Title	Q
id_patient	Q
id_doctor	Q
Doctor: Doctors.LastName	Q

Рис. 3.9. Запит 2

Схема БД наведена на рис. 3.10.

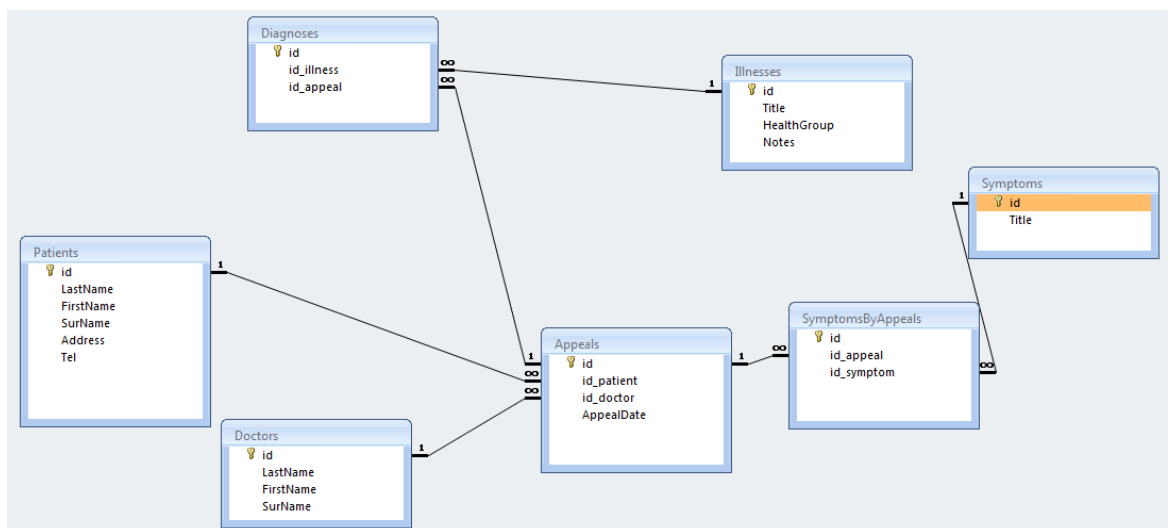


Рис. 3.10. Схема БД

У програмі БД записуємо у файл PatientsDiagnostic. Для доступу до БД використовуємо вказівник на файл (DM).

3.3. Опис медичної системи обліку пацієнтів приватної клініки

Програма складається з 7 функцій. А саме:

Функція встановлення діагноза

```
//функція для створення и открития формы в модальном режиме
bool StartAddDiagnosesForm(int x,int y, int id_appeal, int id_illness, int id)
{
    TAddDiagnosesForm *AddDiagnosesForm = new TAddDiagnosesForm(0);
    try
    {
        AddDiagnosesForm->Left = std::max(0,x - AddDiagnosesForm->Width / 2);
        AddDiagnosesForm->Top=std::max(0,y - AddDiagnosesForm->Height / 2);
        fDM->adoqDBLUCBillness->Close();
        fDM->adoqDBLUCBillness->Open();
        if(id == -1)
            AddDiagnosesForm->Caption="Добавление диагноза";
        else {
            AddDiagnosesForm->Caption="Изменение диагноза";
            AddDiagnosesForm->bAdd->Caption = "Сохранить";
            AddDiagnosesForm->dblucbDiagnoses->KeyValue = id_illness;
        }
    }
}
```

Функція введення симптомів

```
//функція для створення и открития формы в модальном режиме
bool StartAddSymptomsForm(int x,int y, int id_appeal)
{
    TAddSymptomsForm *AddSymptomsForm = new TAddSymptomsForm(0);
    try
```

```

{
    AddSymptomsForm->Left = std::max(0,x - AddSymptomsForm->Width / 2);
    AddSymptomsForm->Top=std::max(0,y - AddSymptomsForm->Height / 2);
    AddSymptomsForm->Caption="Добавление симптома";
    fDM->adoqDBLUCBSymptoms->Close();
    fDM->adoqDBLUCBSymptoms->Open();
    AddSymptomsForm->ShowModal();
    if(AddSymptomsForm->ModalResult == mrOk)
    {
        fDM->adoqTmp->Close();
        fDM->adoqTmp->SQL->Clear();
        fDM->adoqTmp->SQL->Add("INSERT INTO SymptomsByAppeals(id_appeal,
id_symptom) ");
        fDM->adoqTmp->SQL->Add("VALUES (:ID_Appeal, :ID_Symptom) ");
        fDM->adoqTmp->Parameters->ParamByName("ID_Appeal")->Value =
id_appeal;
        fDM->adoqTmp->Parameters->ParamByName("ID_Symptom")->Value =
AddSymptomsForm->dblucbSymptom->KeyValue;
        try
        {
            fDM->adoqTmp->ExecSQL();
        } catch(...) {
            MessageBox(0, "Не удалось добавить симптом!", "Ошибка",
MB_ICONERROR + MB_OK);
            return false;
        }
    }
}

```

Функція управління формою Лікарі

//функция для создания и открытия формы в модальном режиме

bool StartDoctorsForm(int x,int y)

```

{
    TDoctorsForm *DoctorsForm = new TDoctorsForm(0);
    try
    {
        DoctorsForm->Left = std::max(0,x - DoctorsForm->Width / 2);
        DoctorsForm->Top=std::max(0,y - DoctorsForm->Height / 2);
        DoctorsForm->Caption="Доктора";
        DoctorsForm->UpdateDoctors();
    }
}

```

```

DoctorsForm->FormatDoctors();
DoctorsForm->ShowModal();
if(DoctorsForm->ModalResult == mrOk)
{
    return true;
}
else
{
    return false;
}
}
__finally{
    delete DoctorsForm;
}
}

```

Функція звернення

//функція для створення і відкриття форми в модальному режимі

```
bool StartIllnessesForm(int x,int y)
```

```

{
    TIllnessesForm *IllnessesForm = new TIllnessesForm(0);
    try
    {
        IllnessesForm->Left = std::max(0,x - IllnessesForm->Width / 2);
        IllnessesForm->Top=std::max(0,y - IllnessesForm->Height / 2);
        IllnessesForm->Caption="Заболевания";
        IllnessesForm->UpdateIllnesses();
        IllnessesForm->FormatIllnesses();
        IllnessesForm->ShowModal();
        if(IllnessesForm->ModalResult == mrOk)
        {
            return true;
        }
        else
        {
            return false;
        }
    }
}

```

```

    }
    __finally{
        delete IllnessesForm;
    }
void TIllnessesForm::UpdateIllnesses(void)
{
    fDM->adoqIllnesses->Close();
    fDM->adoqIllnesses->Open();
}
void TIllnessesForm::FormatIllnesses(void)
{
    dbgIllnesses->Columns->Items[0]->Title->Caption = "Название";
    dbgIllnesses->Columns->Items[1]->Title->Caption = "Группа здоровья";
    dbgIllnesses->Columns->Items[2]->Title->Caption = "Примечания";
    dbgIllnesses->Columns->Items[1]->Width = dbgIllnesses->Width * 0.2;
    dbgIllnesses->Columns->Items[2]->Width = dbgIllnesses->Width * 0.3;
    dbgIllnesses->Columns->Items[0]->Width = dbgIllnesses->Width
        - dbgIllnesses->Columns->Items[1]->Width
        - dbgIllnesses->Columns->Items[2]->Width
        - 45;
    dbgIllnesses->Columns->Items[3]->Visible = false;
}

```

3.4. Тестування медичної системи обліку пацієнтів приватної клініки

Під час написання даної магістерської роботи, у рамках розв'язання поставленої задачі використано комбінований тип даних. Це дуже зручно, оскільки в них можна зберегти необмежену кількість різної інформації і опрацьовувати її по полях. Також використано функції для вводу даних, оновлення даних, запису даних у БД, зчитування даних з БД, сортування даних, виводу результатів.

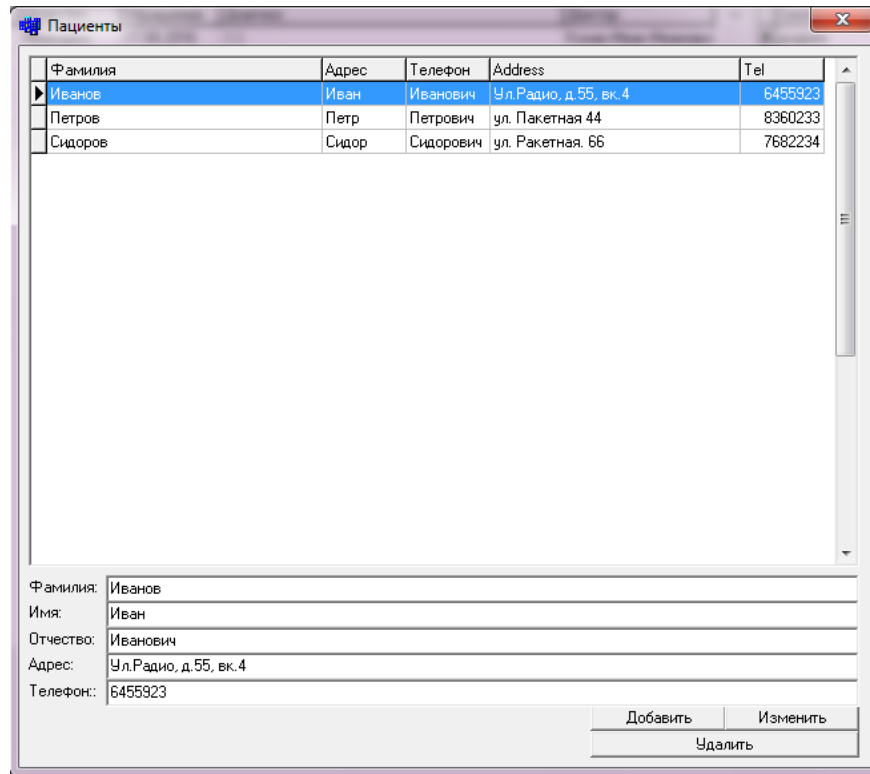


Рис. 3.11. Таблица «Пациент»

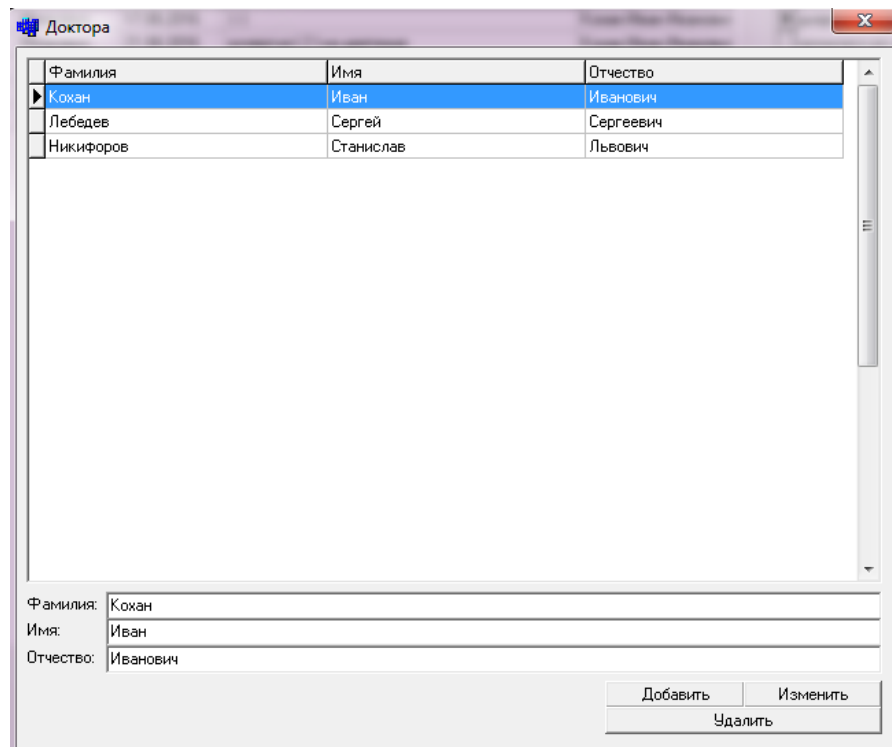


Рис. 3.12. Таблица «Лікарі»

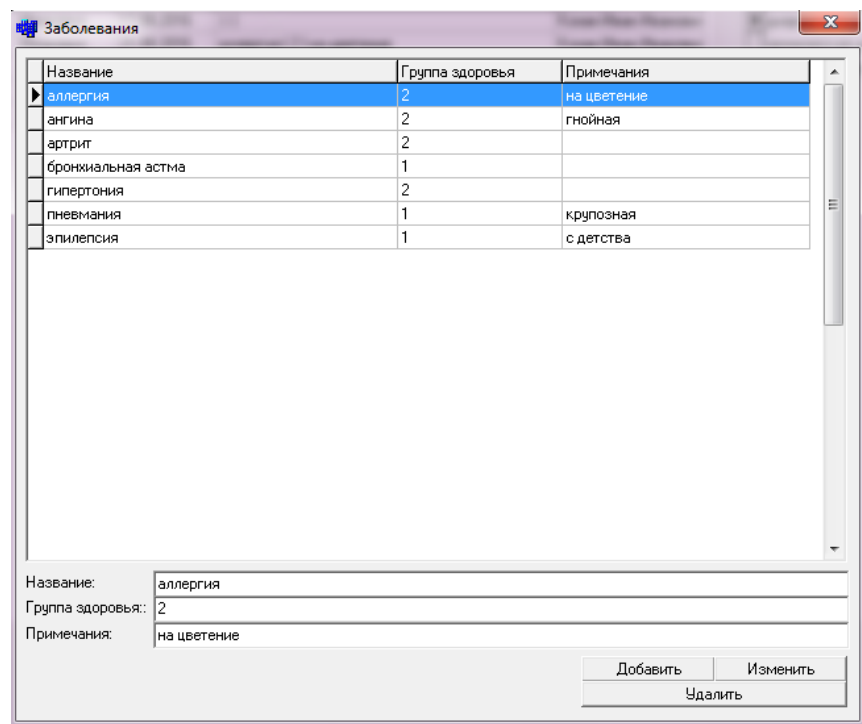


Рис. 3.13. Таблица «Захворювання»

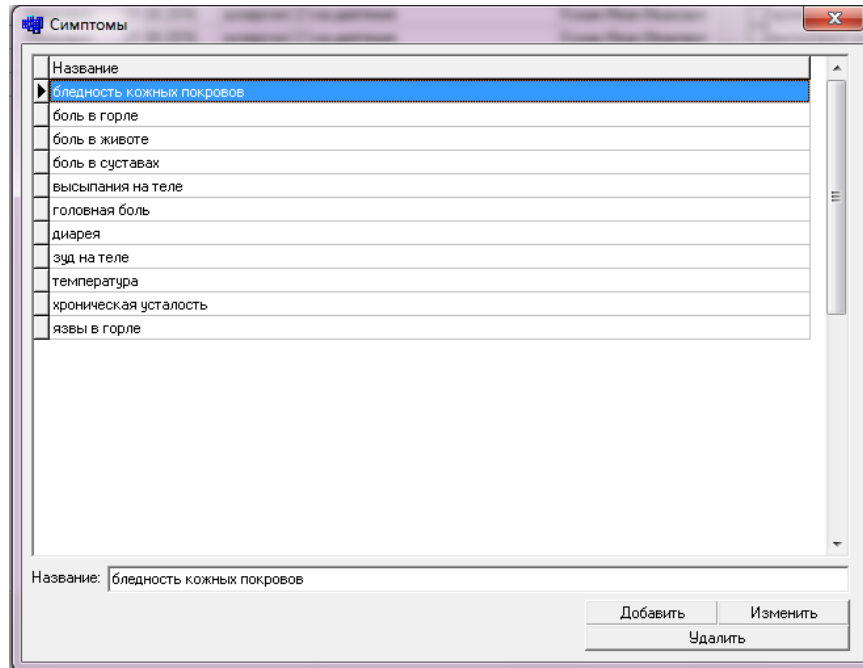


Рис. 3.14. Таблица «Симптомы»

Результат отримую таким чином:

У таблиці «Пацієнт» (рис. 3.11) додаємо новий рядок, вводимо ім'я, прізвище, по батькові, адресу проживання, телефон. Те саме здійснюємо у таблиці «Лікар» (рис. 3.12), за умови, що з'явився новий лікар. Потім встановлюється захворювання (рис. 3.13), згідно до симптомів (рис. 3.14) виявлених у пацієнта.

Дані по кожній таблиці підлягають сортуванню та всім можливим редагуванням (видалення, зміна, додавання)

Симптомы	Фамилия	Имя	Отчество	Обращение	Диагноз	Доктор	Симптомы
Пациенты	Иванов	Иван	Иванович	21.08.2016	аллергия 2 на цветение	Кохан Иван Иванович	
Доктора	Иванов	Иван	Иванович	14.03.2020		Кохан Иван Иванович	
Заболевания	Иванов	Иван	Иванович	14.03.2021	аллергия 2 на цветение	Кохан Иван Иванович	
	Петров	Петр	Петрович	14.03.2021		Лебедев Сергей Серг	
	Сидоров	Сидор	Сидорович	10.01.2021	ангина 2 гнойная	Лебедев Сергей Серг	

Рис. 3.15. Результуюча таблиця

Результуюча таблиця (рис. 3.15) виводиться на головний екран, та має наступні обов'язкові до перегляду рядки:

- ПІП пацієнта;
- дата звернення;
- діагноз;
- лікар, що встановив діагноз;
- СИМПТОМИ.

Під час тестування збоїв та недоліків у роботі розробленого програмного продукту не виявлено, що говорить про високу якість розробки та можливість впровадження на реальному підприємстві.

РОЗДІЛ 4

ОБҐРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ РОЗРОБКИ МЕДИЧНОЇ СИСТЕМИ ОБЛІКУ ПАЦІЄНТІВ ПРИВАТНОЇ КЛІНІКИ

4.1 Техніко-економічне обґрунтування витрат на виконання розробки програмного продукту обробки, формування та передачі звітів виконаних завдань

Метою економічної частини даної магістерської роботи є розрахунок витрат на створення та впровадження інформаційної системи обробки, формування та передачі звітів виконаних завдань. Рішення поставленої задачі неможливо без оцінки техніко-економічної ефективності розроблюваної системи.

У процесі розрахунку зіставляються дані по витратах на створення системи і дані, що відображають зростання ефективності інформаційної системи. Система буде економічно ефективною, якщо показники ефективності (термін окупності, розрахунковий коефіцієнт ефективності капітальних вкладень) відповідають галузевим нормам ефективності капіталовкладень.

Оцінка економічної ефективності системи – складна і трудомістка робота, що вимагає не тільки технічних, але і економічних навичок. І лише в поєднанні цих двох складових може призвести до достовірного результату проведеного аналізу.

Економічні оцінки впровадження системи:

Ефективність - виконання необхідних функцій при мінімальних витратах.

Економічний ефект - результат впровадження системи або технології, виражений у вартісній формі, а так само економія від впровадження.

Термін окупності - період часу, протягом якого окупаються витрати.

Джерела економії - економічна оцінка результатів впливу впровадженого на технологічні процеси обробки і використання даних: поліпшення показників діяльності організації; збільшення обсягів і скорочення термінів переробки інформації; зменшення чисельності персоналу; поява нових можливостей; підвищення продуктивності праці.

Для розрахунку економічної ефективності розробки використовуються наступні основні показники:

- зниження витрат на обробку інформації, річний приріст прибутку (річна економія);
- річна економічна ефективність;
- термін окупності [17, с. 89].

Розроблювана системи дозволяє знизити трудомісткість і витрати часу на обробку даних. Виходячи з цього річна економія (зростання прибутку) розраховується за рахунок економії живої праці. А інші показники ефективності (підвищення точності, оперативності розрахунків) враховуються на якісному рівні. Економічна доцільність розробки системи полягає в економії трудовитрат в порівнянні з ручною обробкою та отриманні більш достовірної інформації за більш короткий час. У нашому проекті задіяні такі фахівці: керівник проекту, фахівець з інформаційного забезпечення (ІЗ) і проектувальник. Матрична організаційна структура демонструє принцип організації роботи над автоматизованою системою (рис. 4.1).

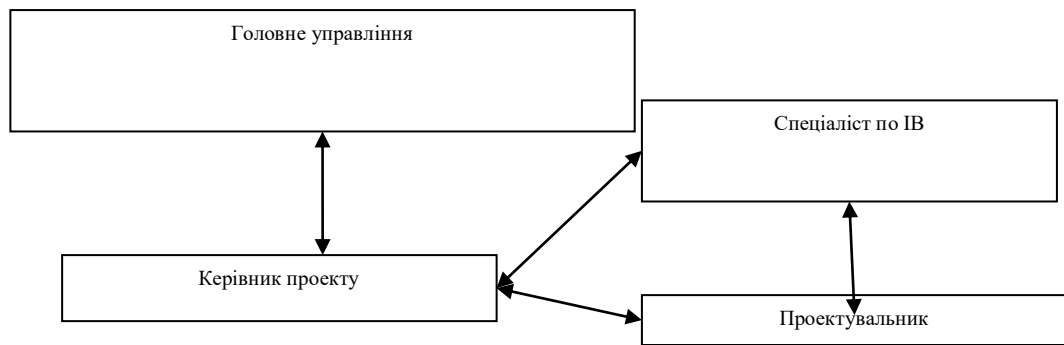


Рис. 4.1 Організаційна структура проектної групи

Розглянемо календарний план роботи над ІС.

Життєвим циклом програми вважається період з моменту прийняття рішення про проведення розробки до повного вилучення з експлуатації.

Етапи життєвого циклу:

Етап роботи над проектом склав близько трьох місяців.

Етап впровадження ІС склав один місяць.

Етап зрілості: перехід до автоматизованої системи - один місяць.

Етап занепаду: висновок недоліків, необхідність удосконалення, поява нових технологій і моральне старіння на даний момент не встановлено.

Складемо календарний графік робіт над проектом (рис.4.2).

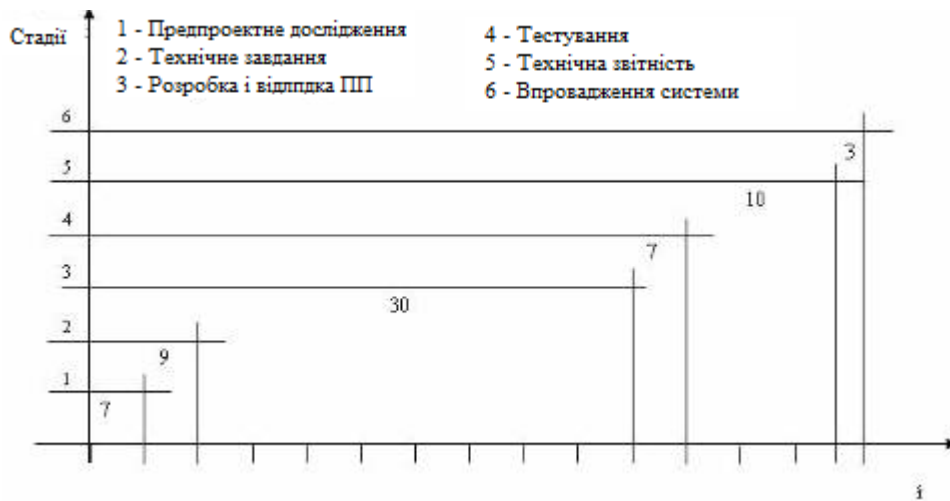


Рис. 4.2. Календарний графік роботи над проектом

Детально опишемо дані в таблиці: Витрати часу на розробку програми (табл.4.1).

Таблиця 4.1

Витрати часу на розробку програми

Найменування етапів	Виконавці	Період (дні)
Предпроектне дослідження (аналіз предметної області)	керівник проекту спеціаліст з ІВ	7
Технічне завдання (формулювання вимог до програми)	керівник проекту спеціаліст з ІВ	9
Розробка і налагодження ІС (розробка архітектури та структури програми, алгоритму, розробка інтерфейсу користувача)	керівник проекту спеціаліст з ІВ проектувальник	30
Тестування (перевірка та аналіз результатів при реальних можливостях зовнішнього середовища)	керівник проекту спеціаліст з ІВ проектувальник	7
Технічна звітність (оформлення програмної документації)	керівник проекту спеціаліст з ІВ проектувальник	10
Впровадження системи	керівник проекту спеціаліст з ІВ	3

За вище вказаним підрахунками термін розробки системи становить 66 днів (трохи більше 2-х місяців).

Теоретична спрямованість - є характерною рисою проведеної роботи. Кінцевим результатом може бути інформаційна система, що демонструє можливість застосування теоретичних розробок і не пропонує вихід на ринок науково-технічної продукції. Отже, основними джерелами витрат при роботі над темою, як частини етапу проектування життєвого циклу цілеспрямованої ІС можна вважати капітальні витрати, які можуть бути враховані і мінімізовані.

Таблиця 4.2

Витрати на витратні матеріали

Найменування матеріалу	Витрати, шт.	Ціна, грн. /шт.	Сума, грн.
флеш - накопичувач	1	60,0	60,0
допоміжна література	4	90	360
канцтовари	-	-	20,0
Разом:	440		

Розрахунок заробітної плати розробників представлений в таблиці 4.3. Враховуючи 22-х робочих днів на місяць.

Таблиця 4.3

Заробітна плата розробників

Найменування етапів	Виконавці	Трудомісткість чол. дн.	Денна з/п, грн
Передпроектне дослідження (аналіз предметної області)	керівник проекту	1	38,235
	Спеціаліст з ІВ	3	49,2
	проектувальник	8	23,667
Технічне завдання (формулювання вимог до програми)	керівник проекту	2	38,235
	Спеціаліст з ІВ	5	49,2
Розробка і налагодження ІС (розробка	керівник проекту	5	38,235

архітектури та структури про грами, алгоритму, розробка інтерфейсу користувача)	Спеціаліст з ІВ	6	49,2
	проектувальник	10	23,667
Тестування (перевірка та аналіз результатів при реальних можливостях зовнішнього середовища)	керівник проекту	2	38,235
	Спеціаліст з ІВ	3	49,2
	проектувальник	2	23,667
Технічна звітність (оформлення програмної документації)	керівник проекту	6	38,235
	Спеціаліст з ІВ	6	49,2
	проектувальник	10	23,667
Впровадження системи	керівник проекту	1	38,235
	Спеціаліст з ІВ	2	49,2

Скомпонувавши дані, що знаходяться в таблиці, в результаті отримуємо (табл.4.4):

Таблиця 4.4

Основна заробітна плата розробників

Виконавці	Робочі дні	Денна з/п, грн	Оклад, грн
Керівник проекту	17	38,235	650,00
Спеціаліст з ІВ	25	49,2	1 230,00
Проектувальник	30	23,667	710,00
Всього:			2 590,00

Додаткова заробітна плата розробників ІС становить 20% від основної заробітної плати: $2590,00 * 0,2 = 518,00$ грн.

Фонд заробітної плати являє собою суму основної та додаткової заробітної плати:

$$\Sigma \text{фзп} = 2590,00 + 518,00 = 3108,00 \text{ (грн)}$$

Отже, витрати з оплати праці розробникам склали 3108,0 гривен.

Відрахування на соціальні потреби становлять 22% від фонду оплати праці: $3108,00 * 0,22 = 1087,80$ (грн)

Накладні витрати становлять 250% від величини основної заробітної плати: $2590,00 * 2,5 = 6475,00$ (грн)

Інші витрати складаються з показників на машинний час (близько 2 - місяців на розробку, налагодження і тестування програми: враховуючи 8 годинний робочий день - 576 годин вартістю 5 грн. / год): $576 * 5 = 2880$ (грн)

На підставі даних результатів можна скласти таблицю: "Калькуляції теми" (табл.4.5).

Таблиця 4.5

Калькуляції теми

Найменування статей витрат	Витрати, грн.
Витратні матеріали	440
Основна заробітна плата розробників	2590,00
Додаткова заробітна плата розробників	518,00
Відрахування на соціальне страхування	1087,80
накладні витрати	6475,00
Інші витрати	2880,00
Разом витрат:	13990,80

У результаті ми отримуємо, витрати з оплати праці розробникам в сумі: 13990,80 гривен.

Розрахуємо загальногосподарські витрати.

Фонд часу, що виділяється на ремонт обладнання:

$$\Phi_{\text{ремо. об.}} = \Phi_{\text{дн}} + \Phi_{\text{міс}} \quad (4.1)$$

де $\Phi_{\text{дн}}$ - денний фонд часу на ремонт обладнання

Фонд робочого часу, що виділяється на щоденні профілактичні роботи:

$$\Phi_{\text{дн}} = t_{\text{смпроф}} \text{ Ч } (D_{\text{к}} - D_{\text{в}} - D_{\text{с}}), \quad (4.2)$$

де

$t_{\text{смпроф}}$ - це кількість годин, що виділяються на щоденні профілактичні роботи:

$$D_{\text{к}} = 62 - \text{дні календарні (2 місяці);}$$

$$D_{\text{в}} = 18 - \text{дні вихідні;}$$

$$D_{\text{с}} = 2 - \text{дні святкові}$$

$$\Phi_{\text{дн}} = 0,5 \text{ Ч } (62 - 18 - 2) = 21 \text{ (година)}$$

Фонд робочого часу, що виділяється на місячні профілактичні роботи:

$$\Phi_{\text{міс}} = t_{\text{міспроф}} * 12, \quad (4.3)$$

де

$t_{\text{міспроф}} = 2:00$ (кількість годин виділене на щомісячні профілактичні роботи)

$$\Phi_{\text{міс}} = 2 * 2 = 4 \text{ (годину)}$$

Дійсний фонд часу роботи обладнання:

$$\Phi_{\text{дійс}} = (D_{\text{к}} - D_{\text{в}} - D_{\text{с}}) * t_{\text{см}} * C, \quad (4.4)$$

де

$$C = 1 \text{ (кількість змін)}$$

$$t_{\text{см}} = 8 \text{ (годин в день)}$$

$$\Phi_{\text{дійс}} = (62 - 18 - 2) * 8 * 1 = 336 \text{ (годину)}$$

Витрати на електроенергію розраховується в основному на споживання енергії комп'ютером.

Витрати на освітлення одного робочого місця:

$$Z_{\text{осв}} = \Pi_{\text{ІКВт}} * N_{\text{л}} * N_{\text{осв}} * \Phi_{\text{дійс}}, \quad (4.5)$$

де

$C_{1\text{КВТ} / \text{ч}} = 0,99$ грн (ціна 1 КВт / год в)

$N_{\text{л}} = 2$ (кількість освітлювальних приладів)

$N_{\text{осв}} = 0,06$ (потужність одного освітлювального приладу)

$Z_{\text{осв}} = 0,99 * 2 \text{ ч} * 0,06 * 336 = 124,2$ (грн)

Витрати на електроенергію для монітора:

$$Z_{\text{елмон}} = C_{1\text{КВТ} / \text{ч}} * N_{\text{мон}} * \Phi_{\text{дійс}}, \quad (4.6)$$

де

$N_{\text{мон}} = 0,4$ (потужність монітора)

$Z_{\text{елмон}} = 0,99 * 0,4 * 336 = 414$ (грн)

Витрати на електроенергію системного блоку:

$$Z_{\text{елсист. бл}} = C_{1\text{КВТ} / \text{ч}} * N_{\text{сист. бл}} * \Phi_{\text{дійс}}, \quad (4.7)$$

де

$N_{\text{сист. бл}} = 0,3$ (потужність системного блоку)

$Z_{\text{елсист. бл}} = 0,99 * 0,3 * 336 = 310,5$ (грн)

Загальна сума витрат на електроенергію

$$\Sigma Z_{\text{ел}} = Z_{\text{осв}} + Z_{\text{елмон}} + Z_{\text{елсист. бл}} \quad (4.8)$$

$\Sigma Z_{\text{ел}} = 124,2 + 414 + 310,5 = 848,7$ (грн)

Отже, витрати на електроенергію складає сума в 848,7 гривень

Не можна не відзначити і витрати на послуги Інтернет - ресурсу. Щомісячна оплата становить 70 грн. А так як розробка ІС зажадала 2 місяці, то витрати склали 140 грн.

Розрахуємо амортизацію обладнання:

$$A_{\text{об}} = C_{\text{об}} * N_{\text{п. в / об}}, \quad (4.9)$$

Де $N_{\text{п. в / об}} = 24\%$ (норма амортизації по устаткуванню)

$C_{\text{об}} = 10\,000$ грн. (Вартість використаного обладнання)

$$A_{об} = 10\,000 * 0,24 = 2400 \text{ (грн)}$$

Отримана сума в 2 400 гривен - амортизація обладнання на протязі року.

$$A_{об1} = A_{об} / Ч_{рд}, \quad (4.10)$$

де

$$Ч_{рд} = 251 \text{ (робочі дні в плинні року)}$$

$$A_{об1} = 2\,400/251 = 9,56 \approx 10 \text{ (грн) - денна сума амортизації.}$$

За весь період розробки проекту витрати на амортизацію обладнання склало 620 гривен.

Проведемо розрахунки по обслуговуванню обладнання, вони становлять 25% амортизаційних відрахувань. Отримуємо $620 * 0,25 = 155$ гривен.

Складемо кошторис витрат на розробку програмного продукту.

Таблиця 4.6

Загальні витрати на розробку програми

Статті витрат	Числове значення (грн)
Витрати з оплати праці розробника програми	3263,40
Витрати на витратні матеріали	440,00
Витрати на електроенергію	848,70
Витрати на інтернет-трафік	140,00
Витрати на амортизацію устаткування	620,00
Витрати на обслуговування обладнання	155,00
Разом:	5467,10

Таким чином на розробку проекту знадобиться затратити 5467,10 гривні. Отже, на підставі отриманих даних розрахуємо економічну ефективність розробки програмного продукту (ПП).

Всі найкращі результати, показуваний ПП і є показник ефекту. Економічний ефект від використання ПП за розрахунковий період Т визначається за формулою, грн.:

$$ET = PT - ZT \quad (4.11)$$

де РТ - вартісна оцінка результатів застосування ПП протягом періоду Т, грн.;

ЗТ - вартісна оцінка витрат на створення і супровід ПП, грн. (Використовуємо ЗК).

Вартісна оцінка результатів застосування ПП за розрахунковий період Т визначається за формулою:

$$P_t = \sum_{t=0}^T P_t * \alpha_t \quad (4.12)$$

Т - розрахунковий період;

P_t - вартісна оцінка результатів року t розрахункового періоду, грн.;

α_t - функція, що дисконтуються, яка вводиться з метою приведення всіх витрат і результатів до одного моменту часу.

Функція, що дисконтуються має вигляд:

$$\alpha_t = 1 / (1 + p)^t, \quad (4.13)$$

p - коефіцієнт дисконтування ($p = E_n = 0.2$, E_n - нормативний коефіцієнт ефективності капітальних вкладень).

$$P_t = \sum_{t=0}^T P_t / 1,2^t \quad (4.14)$$

Таким чином, розроблена нами ІС замінює ручну працю, отже, набір корисних результатів у ній не змінюється. В якості оцінки результатів застосування ІС в рік береться різниця (економія) витрат, що виникає в результаті використання ІС, тобто $P_t = E_y$.

Економія від заміни ручної обробки інформації на автоматизовану утворюється в результаті зниження витрат на обробку інформації і визначається за формулою, грн.:

$$E_y = V_p - V_a \quad (4.15)$$

V_p - витрати на ручну обробку інформації, грн.;

V_a - витрати на автоматизовану обробку інформації, грн.

Витрати на ручну обробку інформації визначаються за формулою:

$$V_p = O_i * V * \Gamma_d / N_b \quad (4.16)$$

O_i - обсяг інформації, оброблюваної вручну, Мбайт;

V - вартість однієї години роботи, грн. / годину;

Γ_d - коефіцієнт, що враховує додаткові витрати часу на логічні операції при ручній обробці інформації;

N_b - норма виробітку, Мбайт / год.

В даному випадку: $O_i = 50$ Мбайт (загальний розмір оброблюваних даних, які вводяться для реєстрації за рік з подальшим підрахунком статистики),

$\Gamma_d = 450/22/8 * 2,55$ грн. / годину,

$\Gamma_d = 2,5$ (встановлений експериментально),

$N_b = 0,004$ Мбайт / год.

Виходить, витрати на ручну обробку інформації будуть рівні:

$$V_p = 50 * 2,55 * 2,5 / 0,004 = 79\,687,5 \text{ грн.}$$

Витрати на автоматизовану обробку інформації розраховуються за наступною формулою:

$$V_a = t_a * V_m + t_o * (V_m + V_o) \quad (4.17)$$

t_a - час автоматичної обробки, год.;

V_m - вартість однієї години машинного часу, грн. / годину;

t_o - час роботи оператора, год.;

V_o - вартість однієї години роботи оператора, грн. / годину.

Для нашої автоматизованої системи:

$$t_a = 18 \text{ год.},$$

$$V_m = 2 \text{ грн.},$$

$$t_o = 83,3 \text{ год.},$$

$$V_o = 450/22/8 * 2,55 \text{ грн.}$$

(Для введення даних оператором в систему знадобиться: (1000 випадків) * (5хв. реєстрації 1 випадку) = 5000 хв. = 83.3 години; Для автоматичної обробки введених даних, знадобиться 1080 хв. = 18 годин на рік).

Отже, витрати на автоматизовану обробку інформації будуть рівні:

$$V_a = 18 * 2 + 83,3 * (2 + 2,55) = 415,02 \text{ грн.}$$

Таким чином, річна економія від впровадження АС дорівнює:

$$E_y = 79\ 687,5 - 415,02 = 79\ 272,48 \text{ грн.}$$

Економічний ефект від використання ІС за рік визначається за формулою, грн.:

$$E_r = E_y - E_n * Z_k.$$

$$E_r = 79\ 272,48 - 0.2 * 37\ 610,2 \approx 71\ 750,44 \text{ грн.}$$

Ефективність розробки може бути оцінена за формулою:

$$E_p = E_r * 0.4/V_k.$$

$$E_p = 71\ 750,44 * 0.4/37\ 610,2 \approx 0.76$$

Оскільки $E_p > 0.20$, розробка інформаційної системи є економічно доцільною.

Здійснимо вартісну оцінку результатів застосування АС (економія) за розрахунковий період $T = 3$ роки (за умови відсутності внесених у програму доопрацювань і змін):

$$P_3 = \sum_{t=0}^3 79272,48/1,2^t = 79272,48 + 79272,48/1,2 + 79272,48/1,2^2 = 79272,48 + 66060,4 + 55050,3 = 200383,18$$

Економічний ефект від використання АС за розрахунковий період $T = 5$ років складе:

$$ET = 200\,383,18 - 37\,610,2 = 162\,772,98 \text{ грн.}$$

Таким чином, поставлена задача доказу економічної ефективності розробленої АС повністю підтвердилася.

Створювана система дає економічний ефект за рахунок, перш за все, скорочення часу і сил співробітника. Розширення сфери застосування АС і її подальше вдосконалення дозволить ще більше підвищити економічний ефект від застосування інформаційної системи.

Витрати на розробку, отримані методом калькуляції, складають 5467,10 грн.

Економічний ефект від використання даного ПП за розрахунковий період (3 роки) складе 162 772,98 грн.

Отже, можна зробити висновок, що отримані показники дають достатньо підстав для запровадження системи обліку пацієнтів приватної клініки.

4.2 Аналіз впровадження розробленого додатку

Впровадження сучасних інформаційних технологій - справа дорога. Функціонування компаній в ринковому середовищі вимагає як мінімум

аналізу економічних наслідків, а ще краще - оцінки економічної ефективності того чи іншого кроку перетворення системи.

У минулому параграфі ми розраховали економічну ефективність і виявили абсолютну окупність витрат на розробку і впровадження. Тепер ми опишемо сам процес впровадження інформаційних систем і проаналізуємо його з точки зору інформаційної системи.

Впровадження системи залежить від багатьох факторів і є нетривіальним практичним завданням. При впровадженні використовується багатоетапна процедура, на кожному з яких здійснюється тісна взаємодія фахівців виконавця і замовника, контроль виконання пропонованих вимог і навчання персоналу.

Слід виділити три етапи впровадження інформаційної системи:

Дослідження. Спочатку проводиться дослідження предметної області і бізнес процесів.

Доопрацювання системи. Програмісти налаштовують або доробляють необхідну функціональність системи.

Запуск системи. Початок реального використання системи, включає процеси навчання персоналу.

На етапі дослідження бізнес процесів відводиться певний час. Тут необхідно як можна більш точно описати, які процеси необхідно поліпшити.

Як правило, функціональність інформаційної системи дещо ширше, ніж реальні процеси. На даному етапі необхідно визначити як присутність тих чи інших функцій позначиться на кінцеву вартість системи, час впровадження, і що найважливіше, чи відповідає пропонована функціональність цілям компанії.

Важливо, щоб результати дослідження були надані в якості окремого документа, де відповідно до вимог компанії повинна бути детально описана вивчена предметна область.

Після стадії дослідження необхідно точно відповісти на питання про вартість і терміни впровадження інформаційної системи.

На етапі доопрацювання системи доцільно контролювати процес реалізації необхідних функцій в інформаційній системі. Розумно, щоб з боку замовника виступала людина, добре знайома із завданнями компанії та її бізнес процесами.

На фазі запуску системи важливо переключити бізнес процеси компанії на використання впровадженої системи. Основне завдання швидко навчити і мотивувати персонал використовувати нову інформаційну систему.

Запроваджена система, як правило, не починає працювати відразу. Потрібно проаналізувати наскільки впровадження було успішно, чи досягнуто основні цілі впровадження.

Впровадження можна вважати успішним, тільки якщо система дозволяє отримувати вигоду, а саме оптимізує роботу служб, дозволяє виконувати роботу швидше, підвищує якість процесів. Необхідно постійно аналізувати показники роботи системи, а також ступінь зацікавленості персоналу у використанні цієї системи.

Процес впровадження інформаційної системи займає як мінімум кілька місяців. Протягом цього часу важливо фокусуватися на цілях, які компанія хоче досягти, впроваджуючи систему. Логічно пам'ятати про можливі ризики і фінансові витрати.

Перш ніж приймати рішення про впровадження інформаційної системи, необхідно визначити те коло завдань, які повинна підтримувати інформаційна система.

Для успішної роботи з інформаційною системою необхідно мати у розпорядженні наступні пристрої [23]. Системні вимоги:

операційна система: Windows 2000, Windows XP, Windows 2003, Vista, Windows 7 і пакетом програм MS Office;

процесор: Pentium 4, 3000 МГц;

оперативна пам'ять: не менше 3 Гб;

вільне місце на жорсткому диску: не менше 1 Гб;

маніпулятор типу "миша";

CD (DVD) – привід.

Інформаційна система успішно пройшла випробування і готова до експлуатації.

РОЗДІЛ 5

ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

У даному розділі магістерської роботи розглядаються питання охорони праці пов'язані з роботою оператора ПК, який буде працювати з розробленим програмним модулем. Розділ включає характеристику приміщення, аналіз шкідливих та небезпечних факторів, опис мікроклімату, шуму, освітлення, а також вимоги до електробезпеки та пожежної безпеки приміщення, яке буде використовуватися для роботи із розробленим програмним забезпеченням. Також розділом передбачено висвітлення питань безпеки в надзвичайних ситуаціях.

5.1. Характеристика приміщення

Для аналізу було обрано приміщення, яке зображене на Рис. 5.1 та має наступні характеристики:

Загальна площа – 30 м² (довжина 6 м, ширина 5 м)

Висота приміщення – 3 м

Об'єм - 90 м³

Висота вікон – 1,5 м, площа вікон – 3,75 кв. м.

Стіни пофарбовані у світло-голубий колір. Стеля пофарбована у білий колір.

Вікна направлені на схід.

Параметри столів: довжина – 2 м, ширина – 1,25 м, висота – 0,75 м.

Розміри стільців: встановлена висота посадки – 45 см, є спинка, підлокітники, регулятор висоти посадки

LED монітори LG IPS226/

Підлога вкрита лінолеумом тілесного матового кольору

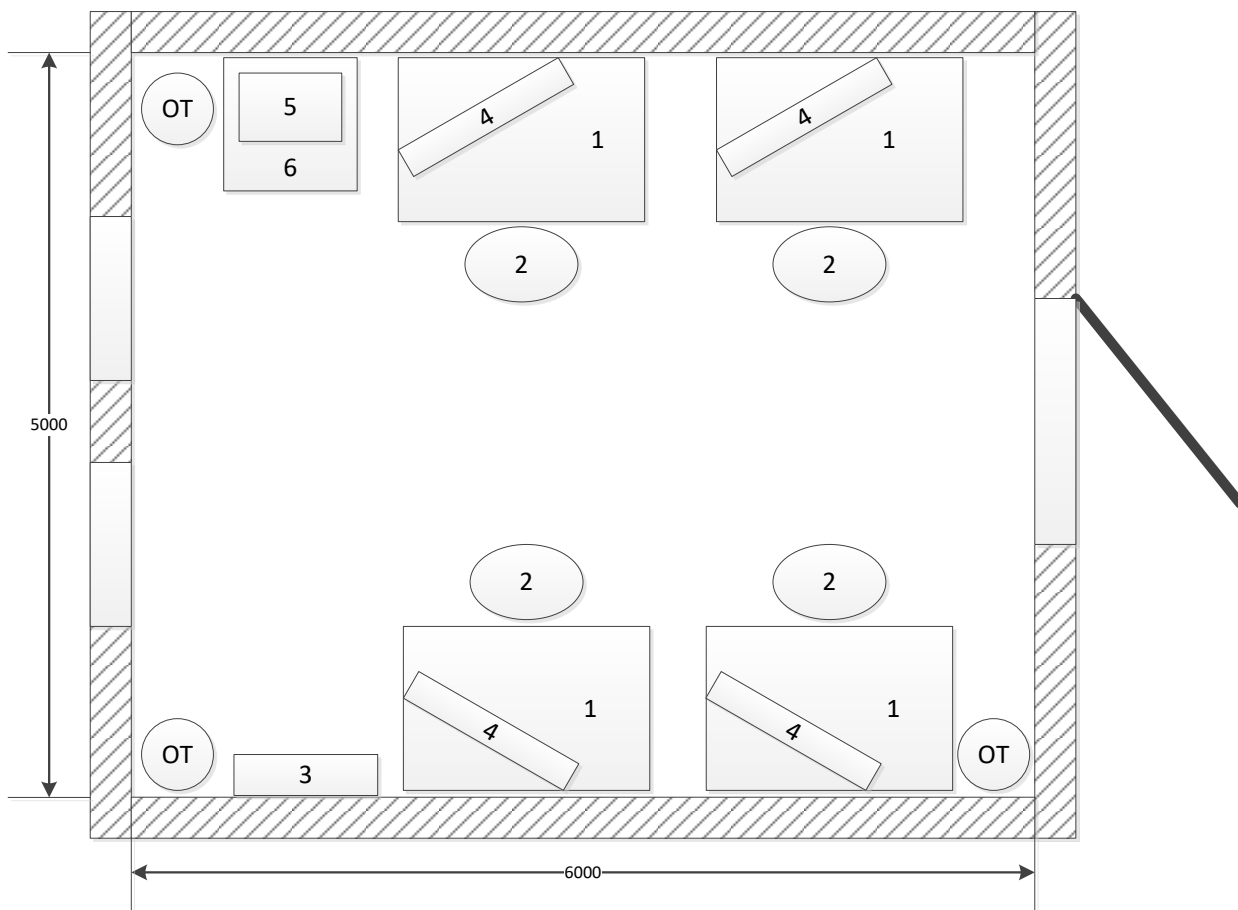


Рис. 5.1. План робочого місця

1 – робочий стіл; 2 – робочий стілець; 3 – кондиціонер; 4 – монітор;
5 – принтер; 6 – загальний стіл; (ОТ) – вогнегасник

Згідно з вимогами НПАОП 0.00-1.28-10 на одну людину повинно виділятися не менше 6 м² і 20 м³. У даному приміщенні працює чотири оператора ПК. Площа на одну людину складає 7,5 м², а об'єм - 22,5 м³, що задовольняє вимогам. Монітор повинен знаходитися під кутом 45-50° до вікна.

5.2. Аналіз шкідливих та небезпечних факторів

На робочому місці оператора ПК виникають небезпечні та шкідливі фактори: підвищений рівень шуму, несприятливі мікрокліматичні умови, недостатній рівень освітленості, висока напруга електричної мережі, статична електрика та інші. Робота з ПК супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі виробничих факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи. Таким чином, виявлення та аналіз небезпечних та шкідливих факторів виробництва оператора ПК є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

Мікрокліматичні умови виробничих приміщень характеризуються такими показниками:

- - температура повітря,
- - відносна вологість повітря,
- - швидкість руху повітря.

За ступенем впливу на тепловий стан людини мікрокліматичної умови поділяють на оптимальні та допустимі.

Роботи, які виконується операторами ПК відносяться до категорії «легка 1а». Таблиця 5.1 містить оптимальні величини температури, які повинні забезпечуватися в офісах при роботі з ПК, згідно ДСН 3.3.6.042-99.

Таблиця 5.1

Оптимальні параметри мікроклімату при виконанні робіт
операторського типу на ЕОМ

Період року	Температура повітря, град С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний період року	22-24	40-60	0,1
Теплий період року	23-25	40-60	0,1

Для дотримання цих параметрів приміщення повинно мати наступне обладнання:

- кондиціонер: спліт-система потужністю по холоду 4,74 кВт, для зниження температури у теплий період року;
- система водного опалення низького тиску – для підвищення температури у холодний період року;
- місцева система вентиляції типу «витяжна шафа», для забезпечення руху повітря.

Регулярне вологе прибирання забезпечує чистоту повітря від пилу. Природний обмін повітря здійснюється через віконні прорізи та кватирки.

За ДБН В.2.5-28-2006 робота оператора ПК класифікується як робота високої точності, розряд «Ш», під розряд «в», так як мінімальним об'єктом розпізнання є піксель, розмір якого складає 0,3-0,5 мм.

В приміщенні використовується комбіноване загальне штучне та природне освітлення.

Природне освітлення складають вікна загальною площею 3,75 м², які виходять на схід і створюють одностороннє бічне освітлення.

Для загального освітлення приміщення використовуються п'ять растрових світильників із дзеркальною решіткою типу PRBLUX 236. Кожен світильник містить 2 люмінесцентні лампи, кожна потужністю 36 Вт, чого достатньо для досягнення нормативної освітленості в 300 лк.

Для забезпечення нормованих значень освітленості у приміщеннях з ЕОМ слід чистити шибки і світильники принаймні двічі на рік і вчасно замінювати лампи, що перегоріли.

Згідно з ДСН 3.3.6.037-99 допустимий шум на постійних робочих місцях має бути 50 дБА.

В розрахунку загального рівня шуму беруть участь 4 комп'ютери, які працюють постійно, та не беруть участь принтер та кондиціонер, які

працюють меншу частину часу. Відповідно, розрахунок загального рівня шуму L для джерел шуму однакової інтенсивності L_n та кількості n проводиться за наступною формулою:

$$L = L_n + 10 \lg n$$

Середній рівень шуму для кожного з комп'ютерів складає близько 23 дБА. Отже загальний рівень шуму складає близько 29 дБА. Наявність принтера вимагає вібраційні підкладки. Для ізоляції від зовнішнього шуму застосовуються спеціальні віконні профілі та звукоізоляція зовнішніх стін плитами зі звукоізоляційними наповнювачами.

Згідно з НПАОП 0.00-1.28-10 робоче місце підпадає під категорію «без підвищеної небезпеки» за наступними критеріями:

- відносна вологість у приміщенні не перевищує 75%;
- у приміщенні відсутній струмопровідний пил;
- підлога покрита лінолеумом, що є струмоізолятором;
- температура не перевищує 30°C;

Живлення електрообладнання здійснюється через однофазну електричну мережу змінного струму напругою 220 В з частотою 50 Гц та з глухо заземленою нейтраллю. Розподільчий щит розташований за межами приміщення.

Згідно з НПАОП 0.00-1.28-10 у робочому приміщенні все електрообладнання відповідає нормам електробезпеки. Розетки знаходяться на висоті 0,4 м від підлоги, вимикачі знаходяться на висоті 1 м від підлоги. ПЕОМ мають заземлення та якісну ізоляцію, і підключені до електромережі через спеціальні мережні фільтри, також для забезпечення безпеки виконується регулярна перевірка електрообладнання та ізоляції кваліфікованими працівниками на предмет можливого порушення.

Пожежа – це неконтрольований процес горіння, що поширюється у часі і просторі надзвичайно швидко та веде до майнових та людських втрат. Для

забезпечення пожежної безпеки у приміщенні встановлені два вуглекислотні вогнегасники, а також встановлена система теплових датчиків.

Згідно з ДСТУ Б В.1.1-36:2016 приміщення відноситься до категорії В – «пожежонебезпечна».

При нормальному режимі роботи можливість виникнення пожежі мінімальна. Можливість виникнення вибухів повністю відсутня.

Можливими причинами загоряння можуть бути пошкодження та замикання в електромережі та електрообладнанні, а також порушення правил безпеки при роботі.

На робочому місці наявні наступні пожежонебезпечні матеріали: папір, пластик, віконні рами, дерев'яні шафи, корпуси техніки, меблі.

Робоче приміщення обладнане трьома вуглекислотними вогнегасниками ВВ-5.

Технічні заходи щодо зниження пожежної небезпеки на підприємстві:

- застосування засобів пожежогасіння (вогнегасники ВВ-5);
- використання засобів автоматичної пожежної сигналізації
- проведення пожежотехнічних обстежень;
- використання інформуючих знаків;
- використання пожежних оповіщувачів.

Сигнал від автоматичної пожежної сигналізації виведений на пульт централізованого спостереження пожежної охорони. В приміщенні використовуються один аспіраційний димовий сповіщувач FAS-420 який покривають зону до 400 кв. м.

5.3. Розробка заходів з охорони праці

Шум – сукупність звуків різної інтенсивності і частоти. Під шумом розуміють усі неприємні та небажані звуки чи їх сукупність, які заважають

нормально працювати, сприймати потрібні звукові сигнали, відпочивати. Шум підвищує втомленість робітника, знижує його працездатність і увагу до небезпеки. Шум як стрес-чинник є загально біологічним подразником, негативно впливає на всі органи і системи організму. В разі тривалого систематичного впливу шуму може виникнути патологія з переважним ураженням слуху, центральної нервової і серцевосудинної систем. Вплив шуму на організм умовно поділяють на специфічний, що спричиняє зміни в органі слуху, і не специфічний – з боку інших органів і систем. Шум є однією з найбільш частих причин зниження слуху.

Основними методами боротьби з виробничим шумом і вібрацією є:

- зменшення шуму в джерелі;
- звукопоглинання і вібропоглинання;
- звукоізоляція і віброізоляція;
- акустична обробка приміщень;
- зменшення шуму на шляху його поширення;
- раціональне планування підприємства і цехів;
- установка глушників шуму;
- вживання засобів індивідуального захисту.

Гранично припустимі значення вібрацій (11-2800 Гц) при роботі з інструментами, механізмами та й устаткуванням, що безупинно впливають на працюючих протягом робочого дня (8 годин).

Електробезпека – це система організаційних та технічних заходів і засобів, що забезпечують захист людей від шкідливого та небезпечного впливу електричного струму, електричної дуги, електромагнітного поля і статичної електрики.

Аналіз виробничого травматизму показує, що кількість травм, які спричинені дією електричного струму є незначною і складає близько 1%,

однак із загальної кількості смертельних нещасних випадків частка електротравм вже складає 20-40% і займає одне з перших місць.

З метою зменшення небезпечного впливу електричного струму і ефективного виконання прямих службових обов'язків працівник ОВС повинен орієнтуватися, у яких місцях він може зустріти небезпечну напругу.

Величини напруги залежать від робочого устаткування:

- до 42 В – напруга використовується для індивідуального освітлення і ручних електроінструментів при роботі в небезпечних зонах;
- 127, 220 В – напруга використовується для освітлення і ручного інструмента на виробництві й у побуті;
- 380 В – напруга використовується при експлуатації промислових установок;
- понад 380 В – напруга використовується для передачі електроенергії на відстань (лініями електропередач – ЛЕП).

Пожежі в обчислювальних центрах (ОЦ) становлять особливу небезпеку, тому що пов'язані з великими матеріальними втратами. Характерна риса ОЦ – невеликі площі приміщень. Як відомо пожежа може виникнути при взаємодії горючих речовин, окиснення й джерел запалювання. У приміщеннях ОЦ присутні всі три основні фактори, необхідні для виникнення пожежі.

Горючими компонентами на ОЦ є : персональна ЕОМ, принтер, дисплей, меблі, книги, документи, ізоляція кабелів і ін.

Джерелами запалювання в ОЦ можуть бути електронні схеми від ЕОМ, прилади, що застосовуються для технічного обслуговування, обладнання електроживлення, кондиціонування повітря, де в результаті різних порушень утворюються перегріті елементи, електричні іскри й дуги, здатні спричинити загоряння горючих матеріалів.

Одним з найбільш важливих завдань протипожежного захисту є захист будівельних приміщень від руйнувань і забезпечення їх достатньої міцності в умовах впливу високих температур при пожежі. Враховуючи високу вартість електронного обладнання ОЦ, а також категорію його пожежної небезпеки, будівля для ОЦ і частини будівлі іншого призначення, у яких передбачено розміщення ЕОМ, повинні бути 1 і 2 ступеня вогнестійкості.

Для виявлення початкової стадії загоряння й оповіщення служби пожежної охорони використовують системи автоматичної пожежної сигналізації (АПС). Крім того, вони можуть самостійно пускати вхід установки пожежогасіння, коли пожежа ще не досягла великих розмірів. Системи АПС складаються з пожежних оповіщувачів, ліній зв'язку й приймальних пультів (станцій).

Ефективність застосування систем АПС визначається правильним вибором типу оповіщувачів і місць їх установки. При виборі пожежних оповіщувачів необхідно враховувати конкретні умови їх експлуатації: особливості приміщення й повітряного середовища, наявність пожежних матеріалів, характер можливого горіння, специфіку технологічного процесу й т.п.

До засобів гасіння пожежі, призначених для локалізації невеликих загорянь, належать пожежні стовбури, внутрішні пожежні водопроводи, вогнегасники, сухий пісок, азбестові ковдри й т.п..

Газові вогнегасники застосовуються для гасіння рідких і твердих речовин, а також електроустановок, що перебувають під напругою.

У виробничих приміщеннях ОЦ застосовуються головним чином вуглекислотні вогнегасники, достоїнством яких є висока ефективність гасіння пожежі, схоронність електронного встаткування, діелектричні властивості вуглекислого газу, що дозволяє використовувати ці вогнегасники навіть у тому випадку, коли не вдається знеструмити електроустановку відразу.

Якщо у інформаційному центрі не вдалося уникнути пожежі, необхідно слідувати твердо установленому порядку дій при пожежі.

Керівник установи, співробітники і обслуговуючий персонал у разі виникнення пожежі або її ознак (дим, запах горіння або тління різних матеріалів і т. п.), а також кожен громадянин зобов'язані:

- негайно повідомити про пожежу за телефоном в пожежну охорону (при цьому необхідно назвати адресу об'єкта, місце виникнення пожежі, а також повідомити своє прізвище);
- прийняти по можливості заходів з евакуації людей, гасіння пожежі та збереження матеріальних цінностей.

Прибулі до місця пожежі зобов'язані:

- продублювати повідомлення про виникнення пожежі в пожежну охорону, чітко назвавши адресу установи, по можливості місце виникнення пожежі, що горить і чому пожежа загрожує (в першу чергу - яка загроза для людей), а також повідомити свою посаду і прізвище, номер телефону, дати сигнал тривоги місцевій добровільній пожежній дружині, повідомити черговому по установі або керівнику (у робочий час);
- вжити негайних заходів по організації евакуації людей, починаючи евакуацію з приміщення, де виникла пожежа, а також з приміщень, яким загрожує небезпека поширення вогню і продуктів горіння, використовуючи для цього наявні сили і засоби;
- перевірити включення в роботу (або привести в дію) автоматичні системи протипожежного захисту (оповіщення людей про пожежу, пожежогасіння, протидимного захисту);
- при необхідності відключити електро-і газопостачання (за винятком систем протипожежного захисту), зупинити роботу транспортувальних пристроїв, агрегатів, апаратів, перекрити сировинні, газові, парові і водяні комунікації, зупинити роботу систем вентиляції в аварійному та суміжному з

ним приміщеннях, виконати інші заходи, що сприяють запобіганню поширення пожежі і задимлення приміщень будівлі;

- припинити всі роботи в будівлі (якщо це допустимо по технологічному процесу виробництва), крім робіт, пов'язаних із заходами щодо ліквідації пожежі;

- видалити за межі небезпечної зони всіх працівників, які беруть участі у гасінні пожежі;

- здійснити загальне керівництво з гасіння пожежі (з урахуванням специфічних особливостей об'єкта) до прибуття підрозділу пожежної охорони;

- забезпечити дотримання вимог безпеки працівниками, які беруть участь у гасінні пожежі;

- одночасно з гасінням пожежі організувати евакуацію і захист матеріальних цінностей;

- організувати зустріч підрозділів пожежної охорони і надати допомогу у виборі найкоротшого шляху для під'їзду до осередку пожежі.

Після прибуття пожежного підрозділу керівник об'єкта (або особа, яка його заміщає) зобов'язаний чітко проінформувати керівника гасіння пожежі про те, чи всі евакуйовані з палаючої чи задимленої будівлі і в яких приміщеннях ще залишилися люди; про конструктивні і технологічні особливості об'єкта, прилеглих будівель та споруд; про наявність та місця зберігання отруйних і вибухових речовин, установок, що не підлягають відключенню за спеціальними вимогами, для чого він повинен мати списки із зазначенням кількості цих речовин і числа установок для кожного приміщення, і т.д., а також організувати залучення сил і засобів об'єкта до здійснення необхідних заходів, пов'язаних із ліквідацією пожежі та попередженням її поширення.

План евакуації – документ, в якому зазначені евакуаційні шляхи і виходи, встановлені правила поведінки людей, а також порядок і послідовність дій обслуговуючого персоналу на об'єкті при виникненні надзвичайної ситуації. План евакуації, знаки безпеки та покажчики напрямку дозволяють вжити необхідних заходів з евакуації людей з місць масового скупчення при виникненні надзвичайних ситуацій.

Призначення плану евакуації:

- чітко позначити шляхи евакуації, евакуаційні виходи, що забезпечують безпеку процесу організованого самостійного руху людей назовні з приміщень, в яких є можливість впливу на них небезпечних факторів пожежі, без урахування вживаних в них засобів пожежогасіння та захисту від диму;
- вказати розташування пожежного обладнання та засобів оповіщення про пожежу;
- нагадати про першочергові дії, які необхідно вжити кожній людині, яка виявила пожежу, що почалася.

У межах даного розділу подано характеристику робочого місця, де проводиться розробка алгоритму. Розглянуто та проаналізовано вплив шкідливих та небезпечних виробничих факторів, до яких відносяться показники мікроклімату, шум, освітлення та випромінювання. Приміщення відповідає встановленим нормам, що регламентовані законодавством, щодо вищезазначених факторів. Також було розглянуто пожежну безпеку та електробезпеку приміщення, в результаті чого було виявлено, що приміщення також відповідає необхідними вимогам безпеки. Аналіз усіх розрахованих у даному розділі факторів показав результати, які дають всі підстави вважати, що розглянуте виробниче приміщення повністю відповідає всім нормативним документам і вимогам.

РОЗДІЛ 6

ЕКОЛОГІЯ

6.1. Аналіз впливу забруднення атмосфери на організм людини

Наша планета оточена повітряною оболонкою – атмосферою, яка поширюється над Землею на 1500 - 2000 км. Однак ця межа умовна, оскільки сліди атмосферного повітря виявлені і на висоті 20000 км.

Наявність атмосфери – необхідна умова існування життя на Землі, оскільки атмосфера регулює клімат Землі, а також згладжує добові коливання температур на планеті. В даний час середня температура поверхні Землі дорівнює 140С⁰. атмосфера пропускає випромінювання Сонця і пропускає тепло. У ній утворюються хмари, дощ, сніг, вітер. Вона є переносником вологи на Землі і середовищем, у якому поширюється звук.

Атмосфера служить джерелом кисневого дихання, вмістилищем газоподібних продуктів обміну речовин, впливає на теплообмін і інші функції живих організмів. Основне значення для життєдіяльності організму мають кисень і азот, вміст яких у атмосферному повітрі відповідно дорівнює 21 і 78% [3].

Кисень необхідний для дихання більшості живих істот (виняток становить лише невелика кількість анаеробних мікроорганізмів). Азот входить до складу білків і азотистих сполук. Вуглекислий газ –о джерело вуглецю органічних речовин – найважливішого компонента цих сполук.

За добу людина вдихає близько 12 - 15 м³ кисню, а виділяє приблизно 580 л вуглекислого газу. Тому атмосферне повітря є одним з основних життєво важливих елементів навколишнього середовища. Необхідно відзначити, що у видаленні від джерел забруднення хімічний склад атмосфери досить стабільний. Проте в результаті господарської діяльності

людини з'явилися осередки вираженого забруднення повітряного басейну в тих районах, де розміщені великі промислові центри. Тут у атмосфері відзначається наявність твердих і газоподібних речовин, що роблять несприятливий вплив на умови життя і здоров'я населення.

До теперішнього часу накопичилося багато наукових даних про те, що забрудненість атмосфери, особливо у великих містах, досягла небезпечних для здоров'я людей розмірів. Відомо чимало випадків захворювань і навіть смерті жителів міст індустріальних центрів в результаті викидів токсичних речовин промисловими підприємствами та транспортом за певних метеорологічних умовах.

Двоокис кремнію і вільний кремній, що міститься в летючій золі, є причиною важкого захворювання легень - силікозу, розвиваючогося у робітників «курних» професій, наприклад у гірників, працівників коксохімічних, вугільних, цементних і ряду інших підприємств. Тканина легенів покривається сполучною тканиною, і ці ділянки перестають функціонувати. У дітей, що проживають поблизу потужних електростанцій, не обладнаних пиловловлювачами, виявляють зміни в легенях, подібні з формами силікозу [4]. Велика забрудненість повітря димом і кіптявою, що триває протягом декількох днів, може викликати отруєння людей зі смертельними наслідками.

Особливо згубно діє на людину забруднення атмосфери в тих випадках, коли метеорологічні умови сприяють застою повітря над містом. Шкідливі речовини, що містяться в атмосфері впливають на людський організм при контакті з поверхнею шкіри або слизовими оболонками. Поряд з органами дихання забруднювачі вражають органи зору і нюху, а також впливаючи на слизову оболонку гортані, можуть викликати спазми голосових зв'язок. Вдихувані тверді і рідкі частки розмірами 0,6 - 1,0 мкм досягають альвеол і абсорбуються у крові, деякі накопичуються в лімфатичних вузлах.

Забруднене повітря дратує здебільшого дихальні шляхи, викликаючи бронхіт, емфізему, астму. До подразників, що викликають ці хвороби, відносяться сірчистий (SO_2) і сірчаний (SO_3) ангідриди, оксиди азоту, хлороводень (HCl), сірководень (H_2S), фосфор та його сполуки [2].

Ознаки та наслідки дій забруднювачів повітря на організм людини проявляються здебільшого у погіршенні загального стану здоров'я: з'являються головні болі, нудота, відчуття слабкості, знижується або втрачається працездатність. Окремі забруднюючі речовини викликають специфічні симптоми отруєння. Наприклад, хронічне отруєння фосфором супроводжується болями у шлунково-кишковому тракті і пожовтінням шкірного покриву. Ці симптоми пов'язані з втратою апетиту і уповільненням обміну речовин. Надалі отруєння фосфором призводить до деформації кісток, які стають все більш крихкими. Знижується опірність організму в цілому.

Оксид вуглецю (II), (CO) – це безбарвний і не маючий запаху газ, який впливає на нервову і серцево-судинну системи, викликаючи задуху. Первинні симптоми отруєння чадним газом (поява головного болю) виникають у людини через 2 - 3 години його перебування в атмосфері, що містить 200 - 220 мг / м³ CO . При більш високих концентраціях чадного газу з'являються відчуття пульсації крові в скронях, запаморочення. Токсичність чадного газу зростає при наявності у повітрі азоту, в цьому випадку концентрацію CO в повітрі необхідно знижувати в 1,5 рази.

Оксиди азоту (NO , N_2O_3 , NO_2 , N_2O). У атмосферу викидаються в основному діоксид азоту NO_2 - безбарвний, такий, що не має запаху отруйний газ, дратівливо діючий на органи дихання. Особливо небезпечні оксиди азоту в містах, де вони взаємодіють з вуглеводнями вихлопних газів і утворюють фотохімічний туман – зміг. Перші симптом отруєння оксидами азоту – легкий кашель. При підвищенні концентрації NO_2 виникає сильний кашель, блювота, іноді головний біль. При контакті з вологою поверхнею

слизових оболонок оксиди азоту утворюють азотну та азотисту кислоти (HNO_3 і HNO_2), що призводить до набряку легенів [3].

Сірчистий ангідрид (SO_2) - безбарвний газ з гострим запахом – вже в малих концентраціях (20 - 30 мг / м³) створює неприємний смак у роті, подразнює слизові оболонки очей і дихальні шляхи. Вдихання SO_2 викликає хворобливі явища в легенях і дихальних шляхах, іноді призводять до набряку легенів, глотки і паралічу дихання.

Вуглеводні (пари бензину, метану і т. д.) мають наркотичну дію, у малих концентраціях викликають головний біль, запаморочення і т. д. Так, при вдиханні протягом 8 годин парів бензину в концентрації 600 мг / м³ виникають головні болі, кашель, неприємні відчуття в горлі. Особливо небезпечні поліциклічні ароматичні вуглеводні типу 3, 4 - бензопірену ($\text{C}_{20}\text{H}_{12}$), що утворюються при неповному згорянні палива. За даними ряду вчених, вони мають канцерогенні властивості.

Альдегіди. При тривалому впливі альдегіди викликають подразнення слизових оболонок очей і дихальних шляхів, а при підвищенні концентрації – головний біль, слабкість, втрату апетиту, безсоння.

З'єднання свинцю. В організм через органи дихання надходить приблизно 50% сполук свинцю. Вплив свинцю порушує синтез гемоглобіну, призводить до захворювання дихальних шляхів, сечостатевої системи, нервової системи. Особливо небезпечні сполуки свинцю для дітей молодшого віку. У великих містах вміст свинцю в атмосфері досягає 5 - 38 мг / м³, що перевищує природний фон в 10000 разів.

Дисперсний склад пилу і туманів визначає загальну проникаючу здатність в організм людини шкідливих речовин. Особливу небезпеку становлять токсичні тонкодисперсні пилинки з розміром частинок 0,5 - 1,0 мкм, які легко проникають в органи дихання.

Нарешті, різні прояви дискомфорту у зв'язку із забрудненням повітря – неприємні запахи, зниження освітленості та ін. – психологічно діють на людей.

Шкідливі речовини, що знаходяться в атмосфері і випадають вражають і тварин. Вони накопичуються в тканинах тварин і можуть стати джерелом отруєнь, якщо м'ясо цих тварин використовується в якості харчових продуктів.

6.2. Нормування вмісту забруднюючих речовин в повітрі

Нормування якості навколишнього природного середовища – це діяльність по встановленню нормативів гранично допустимих впливів людини на природу. Під впливом розуміється антропогенна діяльність, пов'язана з реалізацією економічних, рекреаційних, культурних та інших інтересів людини, що вносить зміни в природне середовище. Найбільш поширеним видом негативного впливу є забруднення, що заподіює шкоду життю і здоров'ю людини, рослинному і тваринному світу і екосистемам. Гігієнічні нормативи – найбільш розроблена система норм, правил і регламентів для оцінки якості навколишнього природного середовища [2].

У систему екологічних нормативів і стандартів входять:

- нормативи якості навколишнього середовища;
- нормативи гранично допустимого шкідливого впливу на стан навколишнього середовища;
- нормативи використання природних ресурсів;
- екологічні стандарти;
- нормативи санітарних і захисних зон.

Методологічною основою для розробки санітарно-гігієнічних і санітарно-епідеміологічних стандартів послужили методи, використовувані в медичній і ветеринарній токсикології.

Залежно від ступеня токсичності отруйних речовин виділяють 4 класи небезпеки (найбільш небезпечний 1-й клас). Впливаючи на організм, шкідливі речовини викликають гострі та хронічні захворювання: гострі виникають після одноразового впливу і можуть призводити до смертельного результату, хронічні розвиваються в результаті систематичного впливу доз, що не приводять до гострого отруєння.

Нормативи якості навколишнього середовища встановлюються у формі нормативів гранично допустимих концентрацій (ГДК) шкідливих речовин, а також шкідливих мікроорганізмів та інших біологічних речовин, що забруднюють навколишнє середовище, і нормативів гранично допустимих рівнів (ГДР) шкідливих фізичних впливів на неї.

Такі нормативи служать також для оцінки стану атмосферного повітря, вод, ґрунтів по хімічним, фізичним і біологічним характеристикам. Встановлені відповідно до вимог законодавства нормативи якості навколишнього середовища служать одним з юридичних критеріїв для визначення її сприятливого стану.

Для більшості речовин встановлюють дві порогові концентрації: мінімальну при гострому отруєнні (ГДК мін.остр.) і мінімальну при хронічному отруєнні (ГДК мін.хрон).

Однак необхідно враховувати, що одні й ті ж концентрації шкідливих речовин по-різному впливають на організми залежно від того, де вони знаходяться: в повітрі, воді або ґрунті. Тому ГДК шкідливих речовин в різних середовищах можуть сильно відрізнятися.

Повітря – середовище, яке безпосередньо оточує людину і тому прямо впливає на її здоров'я. Ще в 20-і рр. ХХ століття почали вводити ГДК шкідливих речовин в робочих приміщеннях. Зазвичай вміст домішок у

повітрі робочого приміщення більше, ніж на майданчику підприємства і тим більше за його межами.

Тому для кожної шкідливої речовини в повітрі встановлюють принаймні два нормативних значення: ГДК в повітрі робочої зони (ГДКр.з) і ГДК в атмосферному повітрі найближчого населеного пункту (ГДКа.п).

ГДК р.з - це концентрація, яка при роботі не більше 41 години на тиждень протягом усього робочого стажу не може викликати захворювань у працюючих та їхніх дітей.

Отже, при нормуванні шкідливих речовин в повітрі виробничих приміщень враховується час перебування людей в зоні забруднення. На території підприємства вміст домішок приймається рівним 0,3 ГДКр.з., так як це повітря використовується для припливної вентиляції.

ГДК а.п - це гранична концентрація, яка впродовж усього життя людини не повинна чинити на нього шкідливого впливу, включаючи віддалені наслідки на навколишнє середовище в цілому.

З урахуванням часу впливу у атмосферному повітрі встановлюють ГДК максимально-разову (ГДКм.р.) і ГДК середньодобову (ГДКс.д.).

Максимально-разова ГДК встановлюється для запобігання рефлекторних реакцій організму людини при короткочасному (20 хвилин) впливі шкідливої речовини.

Середньодобова ГДК – встановлена для попередження загальнотоксичної, канцерогенної, мутагенної та іншого впливу речовини на організм людини. Речовини, що оцінюються за цим нормативом мають здатність тимчасово або постійно накопичуватися в організмі людини.

Необхідність такого роздільного нормування визначається тим, що на підприємстві протягом робочого дня працюють практично здорові люди, а в населених пунктах цілодобово знаходяться не тільки дорослі, а й діти, літні і хворі люди, вагітні та жінки та ін. Тому ГДКр.з > ГДКа.в. Наприклад, для

діоксиду сірки (SO_2) $\text{ПДК}_{\text{рз}} = 10 \text{ мг / м}^3$, а $\text{ПДК}_{\text{а.п}} = 0,5 \text{ мг / м}^3$. Для метилмеркаптану ці показники відповідно рівні 0,8 і $9 \cdot 10^{-6} \text{ мг / м}^3$ [5].

При проектуванні або будівництві підприємств в районах, де повітря вже забруднене, викиди підприємств нормувати з урахуванням присутніх домішок, тобто фонові концентрації ($C_{\text{ф}}$). Якщо у атмосферному повітрі присутні викиди декількох речовин, то сума відносин концентрацій забруднюючих речовин до їх ГДК (з урахуванням $C_{\text{ф}}$) не повинна перевищувати одиниці.

6.3. Засоби контролю за чистотою повітря у виробничому приміщенні

Відомі засоби санітарно-хімічного аналізу повітря можна розділити на три основні групи: лабораторні, експресні та автоматичні (останні забезпечують постійний контроль повітря виробничих приміщень) При розробці всіх типів засобів застосовують різні аналітичні методи: хімічні, фізичні, фізико-хімічні та біохімічні.

Аналітичні та лабораторні методи контролю шкідливих речовин включають відбір проб з наступною доставкою та проведення їх аналізу у лабораторних умовах.

Лабораторні методи аналізу не завжди є достатньо оперативними, але вони забезпечують високу точність визначення наявних у повітрі хімічних речовин. До лабораторних відносяться фотохімічні, люмінесцентні, електрохімічні, хроматографічні, спектрофотометричні, полярографічні та інші методи.

Експресні методи визначення концентрацій у повітрі виробничих приміщень є простими і оперативними, крім того, вони не вимагають джерел електричної і теплової енергії, у практиці експресного аналізу застосовується

індикаційні метод, який передбачає вимірювання концентрації шкідливих речовин індикаторними трубками. В основі індикаційного методу аналізу повітряного середовища лежать коло метричні реакції, що відбуваються на твердих носіях (папірцях, крейді, порошках), просочених індикаторними реактивами.

Експресні методи також полягають у застосуванні спеціальних приладів-газоаналізаторів різних конструкцій. Наприклад, газоаналізатор типу УГ-2 - універсальний переносний прилад, призначений для експресного кількісного визначення різних шкідливих речовин (аміаку, ацетилену, ацетону, бензину, бензолу, оксидів азоту та вуглецю, сірководню, вуглеводнів нафти, хлору та ін.) в повітрі виробничих приміщень [2].

Для експресного аналізу органічних і неорганічних речовин в різних галузях промисловості успішно застосовуються індикаторні трубки, що випускаються іноземними фірмами - \ "Drager \ " (Німеччина), \ "Kitagawa a \ " (Японія), \ "Хігітест \ " (Болгарія).

В умовах сучасних виробництв різних галузей промисловості лабораторні методи і прилади з індикаторними трубками не завжди забезпечують ефективний контроль стану повітряного середовища, оскільки небезпеки безпечної концентрації газів і парів у повітрі робочої зони можуть створюватися за короткий час і процес виникнення небезпечної ситуації носить випадковий характер. Тому автоматичний контроль загазованості повітря за допомогою автоматичних газоаналізаторів стає необхідним елементом контролю та управління технологічними процесами.

Автоматичні газоаналізатори забезпечують: швидкість вимірювання та реєстрації концентрації шкідливої речовини в повітрі; звукову та світлову сигналізацію про перевищення санітарних норм вмісту шкідливих речовин у повітрі на місці вимірювання або у диспетчерських пунктах з включенням в необхідних випадках вентиляції; економію витрат робочого часу при

контролі стану повітряного середовища; можливість їх влаштування у важкодоступних і небезпечних місцях, а також у пересувних лабораторіях.

Промислові автоматичні газоаналізатори, що залежать від принципу дії (методу аналізу) поділяють на механічні, звукові, теплові, магнітні, електрохімічні, іонізаційні, оптичні, оптико-акустичні та ін.

Для встановлення концентрації сірководню, аміаку, фосгену застосовують фотоколометричні автоматичні газоаналізатори \ "Сирена \ " у вибухозахищеному виконанні широко використовуються електрохімічні автоматичні газоаналізатори типу \ "Атмосфера \ ", \ "Мигдаль \ ", \ "Паладій-М \ ", призначені для визначення оксиду вуглецю, діоксиду сірки, сірководню, озону, синильної кислоти у великому діапазоні вимірювань.

За кордоном провідні приладобудівні фірми (в основному Японії та Німеччини) розробляють і випускають автоматичні газоаналізатори, сигналізатори й системи газового аналізу різних типів для контролю вмісту хімічних речовин в повітря.

Для оцінки запиленості повітряного середовища визначають масову концентрацію пилу ($\text{мг} / \text{м}^3$) прямим (гравіметричним) методом, а також його дисперсний склад, кількість пилинок в одиниці об'єму повітря і їх форму рахунковим методом за допомогою мікроскопу [4].

Для встановлення вмісту пилу в повітрі часто використовують непрямі методи, засновані на закономірності зміни фізичних властивостей запиленого повітря в залежності від концентрації пилу – зміни значень поглинання світлових, теплових та іонізуючих випромінювань і т.п. Найчастіше в цьому випадку застосовують радіоізотопні та оптичні методи. Наприклад, для експресного визначення масової концентрації пилу в призначені: фотопиломіри Ф-1, Ф-2; вимірювач концентрації пилу ІКП-ЗД у іскробезпечному виконанні; радіоізотопні пиломіри ПРИЗ-2, ІЗВ-3, тощо.

ВИСНОВКИ

У рамках даної роботи здійснено дослідження сучасних методів та засобів обліку пацієнтів приватної клініки та розробити програмний засіб обліку пацієнтів приватної клініки.

Область використання: для використання в приватних поліклінічних медичних закладах.

Програма забезпечує такі основні функції:

Додавання, редагування, копіювання та видалення даних по пацієнтах, які поступили до приватної клініки

Додавання, редагування, копіювання та видалення даних по лікарях, які працюють у приватній клініці

Додавання, редагування та видалення медичних прийомів як способу резервації виробничих ресурсів.

Формування даних по захворюваннях та діагнозах поставлених лікарями, які працюють у приватній клініці

Метою існування комерційних медичних закладів є отримання прибутку, як і будь-якої комерційної структури. Через це медичний заклад є зацікавленим, щоб тривалість прийому була не більшою за таку тривалість, яка може забезпечити прийнятний рівень собівартості витрачених ресурсів (час лікаря, кабінету). З іншого боку, через існування конкуренції та наявність певних стандартів надання медичних послуг, медичний заклад є зацікавленим, щоб тривалість прийому була не меншою від такої тривалості, яка може забезпечити достатній рівень медичної якості.

Розробка сучасного засобу обліку пацієнтів приватної клініки ґрунтується на окреслених недоліках виявлених у рамках функціонування аналогів, а також на завданні поставленому у рамках даного магістерського дослідження. Розробка якісного програмного додатку обліку пацієнтів

приватної клініки дозволить медичним закладам функціонувати більш ефективно, знизити кількість помилок та проблем в обслуговуванні, а також скоротити людські ресурси та час, що витрачається на ведення документації.

Програма працює у трьох режимах:

– ввід основних особистих даних пацієнтів та лікарів.

Характеристика основних симптомів вичавлених у пацієнта. Після чого обраховується діагноз, сортуються дані і заносяться у базу даних;

– доповнення даних поточними симптомами, які можуть бути виявлені під час обстеження, після чого знову обчислюється діагноз, додається до попереднього і знову сортуються дані і записуються у базу даних;

– вивід результатів (особисті дані пацієнта, лікаря, поточні симптоми, діагноз).

Всі введені дані користувачем записуються у базу даних представлену окремим файлом, там всі дані і зберігаються. БД обрано Microsoft Office Access.

Під час тестування збоїв та недоліків у роботі розробленого програмного продукту не виявлено, що говорить про високу якість розробки та можливість впровадження на реальному підприємстві.

СПИСОК ЛИТЕРАТУРЫ

1. Исакович Н. Н. Концепция развития службы клинической лабораторной диагностики [Текст] / Н. Н. Исакович. – Ейск: ПАО ЕЦРБ, 2015. – 18 с.
2. Болгович М. Ю. Проблема формализации текстовых данных в универсальных медицинских информационных системах [Текст] / М. Ю. Болгович, Д. А. Мурта // Украинский журнал телемедицины и медицинской телематики. – 2016. – Т. 4, № 2. – С. 171-176.
3. Лапрун, И. Автоматизация клинико-диагностических лаборатории [Электронный ресурс] / И. Лапрун // PC Week Doctor. – Сентябрь 2008. – № 3(3). – Режим доступа: \www/URL: <http://www.pcweek.ru/idea/article/detail.php?ID=114686> (Станом на 27.02.2021)
4. Медицинская информационная система «АРИАДНА» [Электронный ресурс]. – Режим доступа: \www/URL: <http://www.reshenie.spb.ru/ariadna/system/system.htm> (Станом на 27.02.2021)
5. Талисманов, В. С. ЛИМС STARLIMS – необходимый компонент современной лаборатории контроля качества [Электронный ресурс] / В. С. Талисманов, Р. А. Андрющенко, А. В. Юрдик, П. В. Карпов // АВРОРА-ИТ – информационные технологии в управлении качеством. – Режим доступа: \www/URL: <http://www.avrora-it.ru/content/articles/index.php?article=2230> (Станом на 27.02.2021)
6. Лабораторная информационная система LabTrak [Электронный ресурс]. – Режим доступа: http://intersystems.ru/partners/catalog/sparm_labtrak.html. – (Станом на 27.02.2021)
7. Назначение модуля «Лаборатория» [Электронный ресурс] // Медиалог — медицинская информационная система. – Режим доступа: \www/URL: http://www.medialog.ru/?tree_id=63#topic1 (Станом на 27.02.2021)

8. Индасофт – промышленная автоматизация [Электронный ресурс]. – Режим доступа: \www/URL: <http://indusoft.com.ua/page.php?id=710> (Станом на 27.02.2021)
9. Bika 3 User manual [Electronic resource] // Bika Lab Systems. – Available at: \www/URL: <http://demo.bikalabs.com/knowledge-centre/manual/bika-3-user-manual> (Станом на 27.02.2021)
10. ISO/IEC 17025. General requirements for the competence of testing and calibration laboratories [Electronic resource]. – publ. 29.06.2015. – Available at: \www/URL: <http://dx.doi.org/10.3403/02033502u>
11. Вейскас Дж. "Эффективная работа с MS Access 2007". [Электронный ресурс]. – URL: <http://www.piter.com>.
12. Голіцина О.Л. Бази даних / Голіцина О.Л., Максимов Н.В., – К.: Форум, 2009. – 352 с.
13. Гончаров А.Ю. Access 2007. Самоучитель із прикладами – Х.: КУДИЦ – ОБРАЗ, 2011. – 272с.
14. Грекул В.И. "Проектирование информационных систем". [Электронный ресурс]. – URL: <http://www.intuit.ru>.
15. Дейт К. Дж. Введение в системы баз данных (седьмое издание). Издательство "Вильямс", 2001. –1072 с.
16. Джеффри Д. Ульман, Дженнифер Уидом. Основы реляционных баз данных. Издательство "Лори", 2006. – 382 с.
17. Джон Кауфельд. Access 2002 для "чайников". – Издательство "Диалектика", 2002. – 304 с.
18. Діго С.М. Бази даних: проектування і управління: Підручник. – К.: Фінанси і статистика, 2011. – 592 с.
19. Діго С.М. Навчальний посібник з курсу "Бази даних" – М., 2010. – 177 с.
20. Єфімов О.М., Патрушіна СМ., Панфьорова Л.Ф., Хашиєв Л.І. Інформаційні системи в економіці. – Львів: ІКЦ "МарТ", 2011. – 352 с.

21. Житкова О., Журина М., Кудрявцева Є. Проектування в СУБД Access. К.: "Інтелект-Центр", 2012. – 64 с.

22. Захаров, В.П. Інформаційні системи: навч. посібник / В.П. Захаров. – Х., 2012. – 187 с.

23. Зірка М.О., Бухтоярова В., Коробкіна Є., Кручинін Р., Бази даних. Розрахунково – графічні роботи з інформаційних систем / За заг. ред. Зірка М.О. Навчально – методичний посібник для студентів вищих навчальних закладів (Спеціальність 080801 "Прикладна інформатика в економіці"), 2011. – 69 с.

24. Игумнов Е. "Спецификация каркаса информационной системы с распределенной архитектурой". [Электронный ресурс]. – URL: <http://www.citforum.ru>.

25. Избачков Ю.С., Петров В.Н. "Информационные системы: Учебник для вузов". [Электронный ресурс]. – URL: <http://www.fictionbook.ru>.

26. Информатизация образования в Киеве. [Электронный ресурс]. – URL: <http://ikt.kiev.ua>.

27. Информационные технологии. "Основы информационной культуры". [Электронный ресурс]. – URL: <http://www.ssti.ru>.

28. Кареева Е.Д. Реляционная модель данных: Методические указания / Краснояр. гос. ун-т; Сост. Е.Д. Кареева. – Красноярск, 2011. – 45 с.

29. Ким М.Ш. "Экономическая эффективность разработки и использования ПО". [Электронный ресурс]. – URL: <http://www.kimmsh.ru>.

30. Кириллов В.В. Учебное пособие. "Основы проектирования реляционных баз данных". [Электронный ресурс]. – URL: <http://www-sbras.nsc.ru>.

31. Глушаков, С. В. Практикум по С++: учебное пособие / С.В. Глушаков, А.В. Коваль, С.В. Смирнов. – Х.: Фолио, 2006. – 525 с.

32. Прата С. Язык программирования С++. М. – СПб. – К.: Diasoft, 2003. – 233 с.

33. Кораблев В. С и С++. К.: ВНУ, 2002. – 453 с.
34. Спірідонов В.І., Войтков В.Г., Обчислювальна техніка і програмування. Хмельницький: ХТІБ 1992. – 344 с.
35. Єжова Л.Ф. Алгоритмізація і програмування процедур обробки інформації. К.: КНАУ, 2000. – 245 с.

Лістинг програми

```
-----  
//-----  
-----  
#include <vc1.h>  
#pragma hdrstop  
//-----  
-----  
USEFORM("MainUnit.cpp", MainForm);  
USEFORM("DM.cpp", fDM); /* TDataModule: File Type */  
USEFORM("IllnessesUnit.cpp", IllnessesForm);  
USEFORM("SymptomsUnit.cpp", SymptomsForm);  
USEFORM("PatientsUnit.cpp", PatientsForm);  
USEFORM("DoctorsUnit.cpp", DoctorsForm);  
USEFORM("NewAppealsUnit.cpp", AppealsForm);  
USEFORM("AddDiagnosUnit.cpp", AddDiagnosesForm);  
USEFORM("AddSymptomUnit.cpp", AddSymptomsForm);  
//-----  
-----  
WINAPI WinMain(HINSTANCE, HINSTANCE, LPSTR, int)  
{  
    try  
    {  
        Application->Initialize();  
        Application->CreateForm(__classid(TfDM), &fDM);  
&MainForm);  
        Application->CreateForm(__classid(TMainForm),  
        Application->Run();  
    }  
    catch (Exception &exception)  
    {  
        Application->ShowException(&exception);  
    }  
    catch (...)  
    {  
        try  
        {  
            throw Exception("");  
        }  
        catch (Exception &exception)  
        {  
            }  
        }  
    }  
}
```

```

        Application->ShowException(&exception);
    }
}
return 0;
}
//-----
//-----

#include <vcl.h>
#pragma hdrstop

#include "PatientsUnit.h"
//-----

#pragma package(smart_init)
#pragma resource "*.dfm"
TPatientsForm *PatientsForm;
//-----

__fastcall TPatientsForm::TPatientsForm(TComponent* Owner)
    : TForm(Owner)
{
}

//функция для создания и открытия формы в модальном режиме
bool StartPatientsForm(int x,int y)
{
    TPatientsForm *PatientsForm = new TPatientsForm(0);
    try
    {
        PatientsForm->Left = std::max(0,x - PatientsForm->Width / 2);
        PatientsForm->Top=std::max(0,y - PatientsForm->Height / 2);
        PatientsForm->Caption="Пациенты";
        PatientsForm->UpdatePatients();
        PatientsForm->FormatPatients();

        PatientsForm->ShowModal();
        if(PatientsForm->ModalResult == mrOk)
        {
            return true;
        }
    }
    else
    {
        return false;
    }
}

```



```

        }
    }
    __finally{
        delete PatientsForm;
    }
}

void TPatientsForm::UpdatePatients(void)
{
    fDM->adoqPatients->Close();
    fDM->adoqPatients->Open();
}

void TPatientsForm::FormatPatients(void)
{
    dbgPatients->Columns->Items[0]->Title->Caption = "Фамилия";
    dbgPatients->Columns->Items[1]->Title->Caption = "Имя";
    dbgPatients->Columns->Items[2]->Title->Caption = "Отчество";
    dbgPatients->Columns->Items[3]->Title->Caption = "Адрес";
    dbgPatients->Columns->Items[4]->Title->Caption = "Телефон";

    dbgPatients->Columns->Items[1]->width = dbgPatients->width * 0.1;
    dbgPatients->Columns->Items[2]->width = dbgPatients->width * 0.1;
    dbgPatients->Columns->Items[3]->width = dbgPatients->width * 0.3;
    dbgPatients->Columns->Items[4]->width = dbgPatients->width * 0.1;
    dbgPatients->Columns->Items[0]->width = dbgPatients->width
>Items[1]->width                -   dbgPatients->Columns-
>Items[2]->width                -   dbgPatients->Columns-
>Items[3]->width                -   dbgPatients->Columns-
>Items[4]->width                -   dbgPatients->Columns-
                                - 45;
    dbgPatients->Columns->Items[5]->Visible = false;
}
//-----
void __fastcall TPatientsForm::dbgPatientsColEnter(TObject *Sender)
{
    if(!dbgPatients->Fields[0]->IsNull)
    {
        eLastName->Text = dbgPatients->Fields[0]->AsString;
        eFirstName->Text = dbgPatients->Fields[1]->AsString;
        eSurName->Text = dbgPatients->Fields[2]->AsString;
    }
}

```

```

        eAddress->Text = dbgPatients->Fields[3]->AsString;
        eTel->Text = dbgPatients->Fields[4]->AsString;
    }
    else
        eLastName->Text = "";
}
//-----
void __fastcall TPatientsForm::dbgPatientsEnter(TObject *Sender)
{
    if(!dbgPatients->Fields[0]->IsNull)
    {
        eLastName->Text = dbgPatients->Fields[0]->AsString;
        eFirstName->Text = dbgPatients->Fields[1]->AsString;
        eSurName->Text = dbgPatients->Fields[2]->AsString;
        eAddress->Text = dbgPatients->Fields[3]->AsString;
        eTel->Text = dbgPatients->Fields[4]->AsString;
    }
    else
        eLastName->Text = "";
}
//-----
void __fastcall TPatientsForm::dbgPatientsCellClick(TColumn *Column)
{
    dbgPatientsEnter(0);
}
//-----
void __fastcall TPatientsForm::dbgPatientsKeyPress(TObject *Sender,
    char &Key)
{
    dbgPatientsEnter(0);
}
//-----
void __fastcall TPatientsForm::dbgPatientsKeyUp(TObject *Sender, WORD
&Key,
    TShiftState Shift)
{
    dbgPatientsEnter(0);
}
//-----
void __fastcall TPatientsForm::bAddClick(TObject *Sender)

```

```

    {
        if(!dbgPatients->Fields[1]->IsNull) {
            fDM->adoqTmp->Close();
            fDM->adoqTmp->SQL->Clear();
            fDM->adoqTmp->SQL->Add("SELECT Count(id) FROM Patients ");
            fDM->adoqTmp->SQL->Add("WHERE LastName = :LastName AND FirstName
= :FirstName AND SurName = :SurName ");
            fDM->adoqTmp->SQL->Add("          AND Address = :Address AND Tel =
:Tel;");
            fDM->adoqTmp->Parameters->ParamByName("LastName")->Value =
eLastName->Text;
            fDM->adoqTmp->Parameters->ParamByName("FirstName")->Value =
eFirstName->Text;
            fDM->adoqTmp->Parameters->ParamByName("SurName")->Value =
eSurName->Text;
            fDM->adoqTmp->Parameters->ParamByName("Address")->Value =
eAddress->Text;
            fDM->adoqTmp->Parameters->ParamByName("Tel")->Value = eTel->Text;
            try
            {
                fDM->adoqTmp->Open();
            } catch(...){
                MessageBox(0, "Не удалось проверить данные!", "Ошибка",
MB_ICONERROR + MB_OK);
                return;
            }

            if(fDM->adoqTmp->Fields->Fields[0]->AsInteger != 0 ) {
                fDM->adoqTmp->Close();
                MessageBox(0, "Такой пациент уже есть в списке!", "Ошибка",
MB_ICONERROR + MB_OK);
                return;
            }

        }
        fDM->adoqTmp->Close();
        fDM->adoqTmp->SQL->Clear();
        fDM->adoqTmp->SQL->Add("INSERT INTO Patients(LastName, FirstName,
SurName, Address, Tel) ");
        fDM->adoqTmp->SQL->Add("VALUES (:LastName, :FirstName, :SurName,
:Address, :Tel);");
        fDM->adoqTmp->Parameters->ParamByName("LastName")->Value =
eLastName->Text;
        fDM->adoqTmp->Parameters->ParamByName("FirstName")->Value =
eFirstName->Text;
        fDM->adoqTmp->Parameters->ParamByName("SurName")->Value = eSurName-
>Text;
        fDM->adoqTmp->Parameters->ParamByName("Address")->Value = eAddress-
>Text;
        fDM->adoqTmp->Parameters->ParamByName("Tel")->Value = eTel->Text;
    }

```

```

        try
        {
            fDM->adoqTmp->ExecSQL();
        } catch(...){
            MessageBox(0, "Не удалось добавить данные!", "Ошибка",
MB_ICONERROR + MB_OK);
            return;
        }
        UpdatePatients();
        FormatPatients();
    }
    //-----
void __fastcall TPatientsForm::bChangeClick(TObject *Sender)
{
    if(!dbgPatients->Fields[1]->IsNull) {
        fDM->adoqTmp->Close();
        fDM->adoqTmp->SQL->Clear();
        fDM->adoqTmp->SQL->Add("SELECT Count(id) FROM Patients ");
        fDM->adoqTmp->SQL->Add("WHERE LastName = :LastName AND FirstName
= :FirstName AND ");
        fDM->adoqTmp->SQL->Add("          SurName = :SurName AND Address <>
:Address AND Tel <> :Tel AND id<>:ID ");
        fDM->adoqTmp->Parameters->ParamByName("LastName")->Value =
eLastName->Text;
        fDM->adoqTmp->Parameters->ParamByName("FirstName")->Value =
eFirstName->Text;
        fDM->adoqTmp->Parameters->ParamByName("SurName")->Value =
eSurName->Text;
        fDM->adoqTmp->Parameters->ParamByName("Address")->Value =
eAddress->Text;
        fDM->adoqTmp->Parameters->ParamByName("Tel")->Value = eTel->Text;
        fDM->adoqTmp->Parameters->ParamByName("ID")->Value = dbgPatients-
>Fields[5]->AsInteger;
        try
        {
            fDM->adoqTmp->Open();
        } catch(...){
            MessageBox(0, "Не удалось проверить данные!", "Ошибка",
MB_ICONERROR + MB_OK);
            return;
        }
        if(fDM->adoqTmp->Fields->Fields[0]->AsInteger != 0 ) {
            fDM->adoqTmp->Close();
            MessageBox(0, "Такой пациент уже есть в списке!", "Ошибка",
MB_ICONERROR + MB_OK);
            return;
        }
    }
}

```

```

        fDM->adoqTmp->Close();
        fDM->adoqTmp->SQL->Clear();
        fDM->adoqTmp->SQL->Add("UPDATE Patients ");
        fDM->adoqTmp->SQL->Add("SET LastName = :LastName, FirstName =
:FirstName, SurName = :SurName, Address = :Address, Tel = :Tel WHERE id =
:ID;");
        fDM->adoqTmp->Parameters->ParamByName("LastName")->Value =
eLastName->Text;
        fDM->adoqTmp->Parameters->ParamByName("FirstName")->Value =
eFirstName->Text;
        fDM->adoqTmp->Parameters->ParamByName("SurName")->Value = eSurName-
>Text;
        fDM->adoqTmp->Parameters->ParamByName("Address")->Value = eAddress-
>Text;
        fDM->adoqTmp->Parameters->ParamByName("Tel")->Value = eTel->Text;
        fDM->adoqTmp->Parameters->ParamByName("ID")->Value = dbgPatients-
>Fields[5]->AsInteger;
        try
        {
            fDM->adoqTmp->ExecSQL();
        } catch(...){
            MessageBox(0, "Не удалось обновить данные!", "Ошибка",
MB_ICONERROR + MB_OK);
            return;
        }
        UpdatePatients();
        FormatPatients();

    }
    //-----
    void __fastcall TPatientsForm::bDelClick(TObject *Sender)
    {
        if(dbgPatients->Fields[5]->IsNull) {
            MessageBox(0, "Данные для удаления не выбраны", "Ошибка",
MB_ICONERROR + MB_OK);
            return;
        }

        fDM->adoqTmp->Close();
        fDM->adoqTmp->SQL->Clear();
        fDM->adoqTmp->SQL->Add("DELETE FROM Patients ");
        fDM->adoqTmp->SQL->Add("WHERE id = :ID;");

        fDM->adoqTmp->Parameters->ParamByName("ID")->Value = dbgPatients-
>Fields[5]->AsInteger;
        try
        {
            fDM->adoqTmp->ExecSQL();
        } catch(...){

```

```
        MessageBox(0, "Не удалось удалить данные!", "Ошибка",
MB_ICONERROR + MB_OK);
        return;
    }
    UpdatePatients();
    FormatPatients();

}
//-----
-----
```