ВСТУП

Покращення навчального процесу проводиться постійно. Кожен навчальний заклад робить усе можливе для підвищення якості матеріалу, за яким навчаються його студенти. І це проявляється не тільки в оновленні застарілої інформації на актуальну, закупці нового обладнання до своїх лабораторних приміщень, а й у змінах у способах подання цих даних. У еру діджіталізації всі знання переходять у цифровий формат для полегшення взаємодії з ними. Адже, не дивлячись на те, що з давніх давен людство використовувало паперові книжки, як основне джерело інформації, погодьтеся цей варіант представлення даних відходить на другий план у нашому сучасному світі. Саме тому потрібно приділяти особливу увагу проблемі подання електронного матеріалу студентам. Те, як буде зберігатися та відображатися користувачеві інформація залежить її доступність, зацікавленість у ній та розуміння прочитаного.

Державний університет телекомунікацій не стоїть в стороні і має свій web-сайт для електронного навчання – Moodle. Це мережевий організаційно-методичний центр, що забеспечує матеріалами студентів для дистанційного навчання. Але, працюючи з ним, я зіткнувся з проблемою. Це недоступність до сайту в зв’язку з відсутністю з’єднання з мережею Інтернет. Траплялося, що в лабораторіях, з певних причин, на деяких комп’ютерах не було з’єднання з мережею і це не давало змогу завантажити навчальні матеріали та успішно засвоїти знання. Саме це неприємне явище і підштовхнуло мене до вивчення даної проблемної ситуації.

Отже, об’єктом дослідження являється розробка десктопного додатку для застосування у навчальному процесі. Навчальний процес – взаємодія складових освітньо-виховної діяльності у єдину організовану структуру, що використовується у певних навчальних закладах.

Предметом дослідження є апаратно-програмний комплекс, що використовується для подачі навчального матеріалу.

Мета роботи полягає в удосконаленні функціонування апаратно-програмного комплексу для покращення навчального процесу спеціальності Комп'ютерної інженерії.

В процесі дослідження вирішувалися питання:

- вибір мови програмування для написання додатка;

- вибір середовища розробки для створення графічного інтерфейсу користувача;

- визначення основного функціоналу додатку;

- реалізація програмним кодом та імплементованими графічними конструкторами;

- систематизація та структуризація даних.

Робота виконувалася слідуючи наступному плану:

- дослідження поточного стану навчального процесу;

- аналіз способів покращення функціонування апаратно-програмного комплексу;

- організоване упорядкування даних навчальних дисциплін;

- створення початкових об'єктів конфігурації графічними конструкторами;

- написання коду мовою програмування 1С;

- імплементація систематизованих даних у додаток.

Наукова новизна справжнього дослідження полягає в удосконаленні функціонування апаратно-програмного комплексу для покращення навчального процесу на базі клієнт-серверного підходу організації інформаційної системи в супротив вже існуючому варіанту.

Практична значущість отриманих в ході роботи результатів полягає в розширенні форм подання інформації студентам Державного університету телекомунікацій при підготовці їх як фахівців.

1 Шляхи покращення функціонування апаратно-програмного комплексу

Перед тим як розпочати створення нашого додатку варто детально дослідити теоретичну базу, з якою нам потрібно буде працювати.

1.1 Вибір оптимальної мови програмування для створення додатку

Мова програмування – система позначень для написання алгоритмів і структур даних, специфічна, штучно виготовлена формальна система, засобами якої можна описувати алгоритми. Мова програмування визначає набір лексичних, синтаксичних та семантичних положень, які встановлюють зовнішній вигляд програми і дії, які виконує виконавець, тобто комп'ютер під її управлінням. Мова програмування забеспечує зручний опис конкретних проблем, формулюємих людиною й розрахованих комп'ютером [1].

Програма, створена мовою програмування, дозволяє при її успішному виконанні обчислювальною системою отримати конкретні результати.

На перших обчислювальних машинах доводилося програмувати двійковими машинними кодами. Однак писати код таким чином доволі трудомістка і складна задача. Для полегшення цієї задачі почали з'являтися мови програмування низького рівня, які давали змогу задавати машинні команди у зрозумілому для людині вигляді. Для декодування їх у двійковий код були розроблені спеціальні програми – транслятори.

Транслятори діляться на компілятори та інтерпретатори. Компілятори змінюють текст програми на машинний код, який можна зберегти і потім використати вже без їхного задійствування. Прикладом цього можуть слугувати виконувані файли з роширенням .exe. Інтерпретатори, в свою чергу, перетворюють лише певну частину програми в машинний код, виконують її і після цього переходять до слідуючої частини. При цьому кожного разу, як виконується програма використовується і інтерпретатор.

Напевно, найбільш відомою мовою низького рівня являється асемблер. Ці мови орієнтовані на конкретний тип процесора, беручи у розрахунок його характеристики. Через це, щоб транспортувати програму написану на асемблері на платформу з іншою апаратною частиною її необхідно майже повністю переписати. Деякі відмінності присутні і в синтаксисі програм під різні компілятори. На щастя, центральні процесори для комп'ютерів фірм Intel та AMD майже всі сумісні і відрізняються один від одного лише деякими специфічними командами. В свою чергу, процесори, призначені для інших пристроїв, таких як відеокарти чи мобільні телефони включають в собі суттєві відмінності.

Головною перевагою мов низького рівня являється ефективність і компактність створеної програми, тому що програміст одержує доступ до всіх можливостей процесора.

Серед недоліків можна виділити значні часові затрати на створення габаритних і комплексних програм; неможливість транспортування результуючої програми на комп'ютер або пристрій з типом процесора, що відрізняється від того, на якому від розроблювався, а також той факт, що програміст, який використовує мову низького рівня має бути висококваліфікованим спеціалістом, гарно розбиратися в пристрої мікропроцесорної системи, для якої розроблюється програма, тобто необхідно знати пристрій комп'ютера і, особливо, пристрій і особливості його процесора.

Мови низького рівня, зазвичай, використовують для створення малих за обсягом системних програм, драйверів пристроїв, програмування спеціалізованих мікропроцесорів, коли найважливішими вимогами є компактність, швидкодія і можливість прямого доступу до апаратної частини.

У свою чергу, мови програмування високого рівня можна сказати більш зрозуміліші людині, ніж обчислювальній машині [2]. Специфікації конкретних комп'ютерних архітектур в них не беруться в розрахунок, тому створені програми легко транспортуються між комп'ютерами. Зазвичай, не виникає ніяких проблем у перекомпілюванні програми під певну комп'ютерну архітектуру і операційну

систему. Розробляти додатки за таких умов набагато простіше і помилок допускається значно менше. Суттєво скорочується час, затрачений на розробку програми, що особливо важливо при роботі над великими програмними проектами [3].

Недоліком мов високого рівня є більший розмір програм в порівнянні з програмами, написаними мовами низького рівня. Тому, зазвичай, мови високого рівня використовуються для створення програмного забеспечення комп'ютерів і пристроїв, в яких обсяг пам'яті не є критичним фактором.

Очевидно, що програма, яку ми маємо на меті створити не потребує тісного зв'язку з процесором, тому ми не будемо використовувати мови низького рівня, а звернемо увагу на більш зрозумілу людському мозку мову високого рівня. Серед великої кількості варіантів я віддаю перевагу такій мові як 1С. Це мова програмування, що використовується у сімействі програм «1С:Підприємство». У версії 7.х була інтерпретованою мовою надвисокого рівня. Інтерпретація тексту програмного модуля в байт-код виконувалася в момент звернення до цього модуля в процесі роботи, таким чином, зазвичай інтерпретувалася лише частина текстів програмних модулів. Взагалі, є дуже багато речей, у назві яких фігурує 1С і це викликає певні непорозуміння. Тому давайте розберемося, що собою являє 1С.

Програма 1С — це продукт фірми «1С», призначений для автоматизації підприємств та організацій найрізноманітніших сфер. Усі програмні рішення розробника створені з урахуванням єдиної технологічної платформи і функціонують за загальними принципами.

Існує поширена думка, що рішення 1С призначені виключно для бухгалтерів. Така думка склалася через те, що першим і найпопулярнішим досі продуктом 1С являється «1С:Бухгалтерія».

Фірма «1С» випустила свою першу конфігурацію ще 1991 року, показавши, що однотипні та рутинні завдання можна перекласти на автоматизовану систему. Це дозволило фахівцям зосередитися на найважливіших та цікавішихх справах.

Майже за 30 років існування фірми «1С» IT-рішення пройшли великий шлях розвитку від простої програми для ведення обліку та складання звітності до розрахованих на багато користувачів систем класу ERP (англ. Enterprise Resource Planning – планування ресурсів організації).

Сьогодні продукти 1С – це широкий асортимент типових рішень та галузевих конфігурацій, які дозволяють автоматизувати будь-які бізнес-процеси в компаніях різних галузей і масштабів.

Дамо відповідь на питання, що таке «1С:Підприємство 8.3». Під терміном "програма "1С" мається на увазі поєднання технологічної платформи "1С" та прикладного рішення (конфігурації) [4]. Прикладне рішення встановлюється на платформу. Конфігурацій дуже багато (наприклад, Бухгалтерія, Зарплата та управління персоналом, Управління торгівлею та ін.), а платформа одна. Кожна платформа має свою версію. І кажучи про «1С:Підприємство 8.3» мове йдеться саме про платформу.

У чому тоді різниця між платформою та конфігурацією. Платформа "1С:Підприємство 8.3" - це основа, без якої неможливо використовувати жодне прикладне рішення лінійки 1С. Фактично це фундамент для встановлення однієї чи кількох конфігурацій. Цей принцип є загальним для всіх версій 1С: як найактуальнішою — програма 1С версії 8.3, так і для попередніх — систем «1С:Підприємство 8.2», 8.0, 7.7 та ін.

На основі платформи програмісти 1С пишуть програми (конфігурації) для користувачів. Платформа «1С» дуже функціональна, вона включає широкий список найрізноманітніших можливостей.

У свою чергу, конфігурація - це весь функціонал, який забезпечує працездатність платформи. Таким чином, перший елемент не може існувати окремо від другого. Будь-який програміст може доопрацювати прикладне рішення під індивідуальні завдання компанії, але платформа залишиться незмінною.

Небагато про версії програм 1С 8.3. Найсучасніша та найактуальніша версія платформи на момент написання магістерської роботи – 1С 8.3. Працювати можна і на ранніх релізах - 8.2 або 8.0. Деякі користувачі все ще залишаються вірними вдалому релізу платформи 7.7. Однак для повноцінної та комфортної роботи з використанням всього спектра можливостей рекомендується встановлювати версію 1С 8.3.

У чому переваги версії програми 1С 8.3 порівняно з ранніми релізами.

Перше, це хмарні технології. У програмах 1С 8.3 можна працювати з будь-якої точки світу, де є Інтернет, через потужні та безпечні хмарні сервери. Для цього у версії 1С 8.3 покращено показники розподілу навантаження на сервер, підвищено відмовостійкість, а також створено профілі безпеки та багато іншого.

Друге, це мобільна платформа. Прикладне рішення 1С 8.3 ще більше пристосоване до роботи на мобільних пристроях. Це дозволяє користувачам використовувати софт 1С на планшетах або смартфонах під управлінням ОС Android і iOS.

Наступною перевагою являється режими тонкого та веб-клієнта. Тонкий клієнт та веб-клієнт дозволяють підключатися до бази даних системи 1С через веб-сервер. Для використання тонкого клієнта необхідно встановити невелику програму. Для роботи через веб-клієнт нічого не потрібно встановлювати, достатньо відкрити браузер, ввести адресу сервера та розпочати роботу з віддаленою базою.

Великим покращенням також є інтерфейс "Таксі" [5]. Особливістю оновленого інтерфейсу «Таксі» є великий шрифт, покращена навігація, адаптація під стандарти веб-додатків, а також максимізація робочого простору. Користувачі програм 1С 8.3 зможуть налаштувати зовнішній вигляд системи під себе та розташувати панелі так, як зручно.

Невід'ємною частиною являється відмовостійкий кластер серверів. Підвищена стійкість платформи до непередбачених збоїв або несподіваного відключення системи завдяки кластеру серверів. Вся інформація у програмах 1С 8.3 зберігається в основі

кластера серверів, а непередбачені збої не призведуть до її втрати. Робочі процеси буде відновлено саме з того місця, де стався збій.

Адміністрація прав користувачів також покращилася в порівнянні із попередніми версіями платформи. Можливість розділення прав доступу користувачів до інформації також є перевагою програм 1С 8.3. Платформа 1С дозволяє максимально гнучко налаштовувати доступ співробітників до документів та іншої корпоративної інформації.

Програми «1С:Підприємство» працюють на основних відомих ОС:

- Windows;
- Linux;
- iOS;
- Android;
- Mac OS.

Версія для Windows була найпершою. Це функціональний, потужний та знайомий усім інструмент, який постійно покращується та розвивається, щоб працювати стабільно та без проблем. Версія під Linux вважається відносно свіжою, в ній поки що можуть зустрічатися помилки.

Платформа для Mac OS знаходиться на стадії бета-тестування. Однак, ґрунтуючись на відгуках фахівців у сфері, працює вона стабільно та швидко. За більш ніж рік експлуатації 1С на Mac не було виявлено будь-яких зависань, помилок та інших факторів, що перешкоджають повноцінному функціонуванню системи. Операційну систему для роботи на Apple відрізняють надійність, безпека, продуктивність, а також, безперечно, чудовий дизайн.

Для того, щоб забезпечити такі можливості, система «1С:Підприємство» має різні режими роботи: 1С:Підприємство та Конфігуратор.

Режим 1С: Підприємство є основним та служить для роботи користувачів системи. У цьому режимі користувачі вносять дані, обробляють їх та отримують підсумкові результати.

Режим Конфігуратор використовується розробниками та адміністраторами інформаційних баз. Саме цей режим надає інструменти, необхідні для модифікації існуючої або створення нової конфігурації.

1.2 Режими роботи бази

Бази 1С можуть працювати у двох режимах роботи: файловому та клієнт-серверному.

Файловий варіант роботи – один із варіантів роботи системи «1С:Підприємство 8». Файловий варіант роботи розрахований на персональну роботу одного користувача або невелику кількість користувачів в локальній мережі [7].

У цьому варіанті всі дані інформаційної бази (конфігурація, база даних, адміністративна інформація) розміщуються в одному файлі - файлової бази даних (рис. 1.1). Роботу з цією базою даних здійснює файлова СУБД, розроблена фірмою «1С» і є частиною платформи.

Такий варіант роботи забезпечує легкість встановлення та експлуатації системи. При цьому для роботи з інформаційною базою не потрібні додаткові програмні засоби, достатньо мати операційну систему та «1С:Підприємство 8».

Рисунок 1.1 –Файловий режим роботи бази

Файловий варіант роботи забезпечує цілісність інформаційної бази та просте створення резервних копій. Виключена ситуація, коли користувач може помилково (наприклад, при копіюванні інформаційної бази) переплутати різні файли інформаційної бази та привести таким чином систему в непрацездатний стан.

Резервне копіювання може здійснюватися на файловому рівні шляхом простого копіювання файлу інформаційної бази.

При роботі у файловому варіанті, за рахунок використання механізму транзакцій, платформа «1С:Підприємства 8» мінімізує ризик порушення цілісності даних при збоях комп'ютерів та локальної мережі. Однак, зрозуміло, використання клієнт-серверного варіанта забезпечує більшу надійність, тому що в ньому забезпечується повна незалежність запису даних на сервері від збоїв клієнтських комп'ютерів та локальної мережі.

Робота у файловому варіанті можлива як безпосередньо, безпосередньо з файлом бази даних, так і через веб-сервер, якщо використовуються клієнтські підключення за протоколом HTTP або HTTPS.

Робота з файловою базою даних можлива за допомогою тонкого клієнта або товстого клієнта.

Якщо використовується товстий клієнт, він сам реалізує всю функціональність файлової СУБД.

Якщо використовується тонкий клієнт, то комп'ютері, де запущений сам тонкий клієнт, організується спеціалізоване серверне середовище.

В рамках цього спеціалізованого середовища виконуються завантаження необхідних для роботи системи серверних компонентів завантаження конфігурації, інші дії, необхідні організації нормальної роботи системи з інформаційною базою.

З погляду тонкого клієнта, це середовище виступає у ролі сервера. З точки зору операційної системи, дане спеціалізоване середовище не виділено в окремий процес і виконується в рамках процесу тонкого клієнта.

Робота з файловою базою даних через веб-сервер можлива за допомогою тонкого клієнта чи веб-клієнта.

І тут модуль розширення веб-сервера створює аналогічне серверне середовище на веб-сервері кожної інформаційної бази.

Прямо в адресний простір веб-сервера завантажується компонент для роботи з файловою базою даних та дані інформаційної бази. При цьому навантаження на веб-сервер значно зростає, а користувачі однієї бази не мають можливості працювати паралельно. Усі їх запити до бази даних вибудовуються в одну чергу.

З цієї причини такий варіант роботи є скоріше тестовим. Щоб, наприклад, спробувати, як інформаційна база працює через веб-сервер, із веб-клієнтом. Як робітник такий варіант, напевно, можна використовувати в особливих випадках для дуже невеликих робочих груп.

Клієнт-серверний варіант роботи – один із варіантів роботи системи «1С:Підприємство 8». Клієнт-серверний варіант роботи призначений для використання у робочих групах або масштабі підприємства. Він реалізований на основі трирівневої архітектури клієнт-сервер (рис. 1.2).

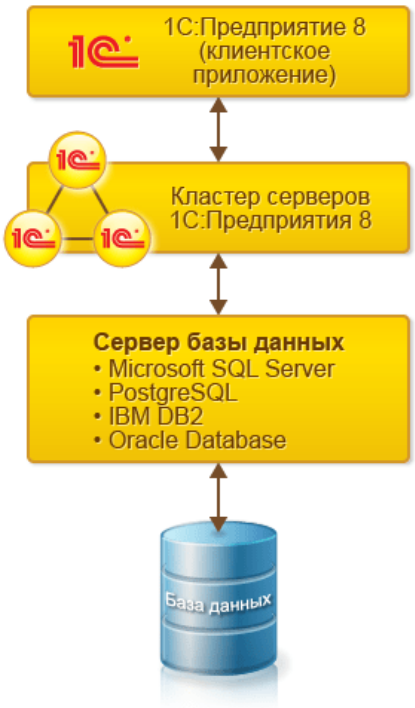

Рисунок 1.2 – Клієнт-серверний режим роботи бази

Клієнт-серверна архітектура поділяє всю працюючу систему на три різні частини, які певним чином взаємодіють між собою:

- клієнтська програма;
- кластер серверів «1С:Підприємства 8»;
- сервер бази даних.

Клієнтська програма — це програма, що працює на комп'ютері користувача та забезпечує інтерактивну взаємодію системи «1С:Підприємство 8» з користувачем, на відміну від інших компонентів системи (програм та робочих процесів), призначених виключно для програмної взаємодії з іншими частинами системи або з іншими програмними об'єктами.

Кластер серверів «1С:Підприємства 8» - основний компонент платформи, що забезпечує взаємодію між користувачами та системою управління базами даних у клієнт-серверному варіанті роботи. Наявність кластера дозволяє забезпечити безперебійну, стійку до відмови, конкурентну роботу великої кількості користувачів з великими інформаційними базами.

Кластер серверів «1С:Підприємства 8» є логічним поняттям і є сукупністю робочих процесів, що обслуговують один і той же набір інформаційних баз.

Система управління базами даних (СУБД) - спеціалізована програма (найчастіше комплекс програм), призначена для організації та ведення бази даних.

Платформа підтримує роботу із п'ятьма СУБД.

Одна з цих СУБД - файлова СУБД - розроблена фірмою «1С» і є частиною платформи.

Інші — це СУБД сторонніх постачальників. Як сервер баз даних можуть використовуватися:

- Microsoft SQL Server;
- PostgreSQL;
- IBM DB2;
- Oracle Database.

Підтримуючи PostgreSQL, Фірма «1С» випускає власні релізи PostgreSQL від 1С, що забезпечують сумісність із платформою 1С:Підприємство та значно підвищують продуктивність PostgreSQL при роботі в типових сценаріях використання продуктів фірми «1С».

Програма, що працює у користувача, (клієнтська програма) взаємодіє з кластером серверів «1С:Підприємства 8», а кластер, при необхідності, звертається до сервера баз даних.

При цьому фізично кластер серверів «1С:Підприємства 8» та сервер баз даних можуть розташовуватися як на одному комп'ютері, так і на різних. Це дозволяє адміністратору за необхідності розподіляти навантаження між серверами.

Використання кластера серверів "1С:Підприємства 8" дозволяє зосередити на ньому виконання найбільш об'ємних операцій з обробки даних. Наприклад, при виконанні навіть дуже складних запитів програма, що працює у користувача, отримуватиме необхідну їй вибірку, а вся проміжна обробка виконуватиметься на сервері. Зазвичай збільшити потужність кластера серверів набагато простіше, ніж оновити парк клієнтських машин.

Іншим важливим аспектом використання 3-х рівневої архітектури є зручність адміністрування та впорядкування доступу користувачів до інформаційної бази. У цьому варіанті користувач не повинен знати про фізичне розташування конфігурації або бази даних. Весь доступ здійснюється через кластер серверів "1С:Підприємства 8". При зверненні до тієї чи іншої інформаційної бази користувач повинен вказати лише ім'я кластера та ім'я інформаційної бази, а система запитує відповідно ім'я та пароль користувача.

«1С:Підприємство 8» використовує можливості системи управління базами даних для ефективної вибірки інформації: механізм запитів орієнтований на максимальне використання СУБД для виконання розрахунків та складання звітів та перегляд великих динамічних списків забезпечується без виконання великої кількості

звернень до бази; при цьому користувачеві надаються можливості ефективного пошуку, а також налаштування відбору та сортування.

Розгортання клієнт-серверного варіанта та його адміністрування виконується досить просто. Наприклад, створення бази даних виробляється у процесі запуску конфігуратора (як і, як й у файлового варіанта).

Робота в клієнт-серверному варіанті можлива як безпосередньо з кластером, так і через веб-сервер. При цьому у разі безпосереднього підключення до кластера товстий клієнт та тонкий клієнт використовують протокол TCP/IP. При підключенні через веб-сервер тонкий клієнт та веб-клієнт використовують протокол HTTP або HTTPS.

Вся робота з прикладними об'єктами, читання та запис бази даних виконується лише на сервері. Функціональність форм та командного інтерфейсу також реалізована на сервері.

На сервері виконується підготовка даних форм, розташування елементів, запис даних форм після зміни. На клієнті відображається вже підготовлена на сервері форма, виконується введення даних та виклики сервера для запису введених даних та інших необхідних дій.

Аналогічно командний інтерфейс формується на сервері та відображається на клієнті. Також звіти формуються повністю на сервері і відображаються на клієнті.

При цьому механізми платформи орієнтовані мінімізацію обсягу даних, переданих на клієнтський комп'ютер. Наприклад, дані списків, табличних частин та звітів передаються із сервера не відразу, а в міру перегляду їх користувачем [8].

На сервері виконуються:

- запити до бази даних;
- запис даних;
- проведення документів;
- різні розрахунки;
- виконання обробок;

- формування звітів;
- підготовка форм для відображення.

На клієнті виконується:

- отримання та відкриття форм;
- відображення форм;
- «спілкування» з користувачем (попередження, питання, тощо);
- невеликі розрахунки у формах, що вимагають швидкої реакції ;
- робота з локальними файлами;
- робота із торговим обладнанням.

Керувати функціональністю форм можна як на сервері, а й у клієнті. На клієнті підтримується робота вбудованої мови. Він використовується в тих випадках, коли необхідно провести розрахунки, пов'язані з відображеною на екрані формою, наприклад швидко (без звернення до сервера) підрахувати суму рядка документа на основі ціни та кількості; поставити користувачеві питання та опрацювати відповідь; прочитати файл із файлової системи комп'ютера та відправити його на сервер.

Однак робота вбудованої мови на клієнті підтримується в обмеженому обсязі. Клієнтські процедури в модулях явно відокремлюються від серверних, і в них використовується обмежений склад об'єктної моделі вбудованої мови.

На клієнті не допускається безпосередня робота з базою даних. Не допускається робота безпосередньо з прикладними об'єктами, наприклад, недоступні такі типи вбудованої мови, як Довідник Об'єкт. <ім'я>. Не допускається використання запитів. При необхідності виклику дій з даними в коді клієнта потрібно викликати серверні процедури, які вже будуть звертатися до даних.

Переваги файлового режиму роботи:

- оптимальний для невеликої кількості користувачів (до 5-ти);
- простота встановлення та експлуатації системи;

- для роботи з інформаційною базою не потрібні додаткові програмні засоби, крім операційної системи та 1С Підприємство;
- просте створення резервних копій шляхом простого копіювання файлу інформаційної бази;
- невисока ціна.

Мінуси:

- розмір бази до 10 ГБ в одній таблиці;
- менш надійна ніж клієнт-серверний варіант;
- без активних користувачів не працюють регламентні завдання.
- немає відмовостійкості. База 1С - це папка в мережі, будь-яке пошкодження файлів у цій папці через мережу може пошкодити інформаційну базу;

Переваги клієнт-серверного варіанту роботи:

- розподіл навантаження між серверами;
- найкраща надійність даних від збоїв клієнтських комп'ютерів та локальної мережі;
- оптимально для виробничих компаній;
- зручність адміністрування;
- розмір бази не обмежений;
- регламентні завдання працюють без користувачів;
- відмовостійкість (перемикання на інший сервер).

Мінуси:

- вищі витрати.

Поговоримо про основні об'єкти конфігурації.

Будь-яке прикладне рішення складається з кількох складових елементів чи об'єктів конфігурації. У тому числі розробник збирає певну структуру, та описує

зв'язок між її частинами за допомогою специфічних алгоритмів. Перед створенням прикладного рішення виконуються такі кроки:

- проводиться аналіз робочих процесів;
- вивчаються предметні галузі;
- у кожній області виділяються робочі сутності;
- на основі зібраних даних вибираються об'єкти для нової конфігурації.

Серед об'єктів можуть бути:

- документи - облік господарських операцій організації;
- регістри - інформація про зміну об'єктів;
- довідники – списки можливих значень атрибутів;
- константи - значення за замовчуванням, налаштування;
- інші компоненти.

Розберемося, які є області застосування продуктів 1С. Програми «1С:Підприємство 8.3» створені для вирішення конкретних обліково-управлінських завдань та мають свою функціональну та галузеву спрямованість. У кожному рішенні поєднуються стандартні функції, загальні більшості систем, і навіть галузеві можливості з урахуванням індивідуальних завдань організації. Будь-яка програма лінійки «1С:Підприємство 8» може бути адаптована для конкретної галузі.

Крім цього, ви можете використовувати системи як окремо, так і спільно з іншими програмами лінійки. Також можна інтегрувати «1С:Підприємство» зі сторонніми програмами. Наприклад, продуктами MS Office або внутрішнім програмним забезпеченням організації.

Можливості платформи «1С:Підприємство» дозволяють використовувати програми лінійки не тільки в рамках офісу, але й віддалено в «хмарі». Ви зможете підключитися до програм через веб-браузер у будь-який час та з будь-якої точки земної кулі.

Які переваги це дає? Користувачі перестають бути прив'язаними до робочого місця. Крім цього, за такої організації роботи ви знижуєте витрати на покупку

програмного забезпечення, роботу сервера та обслуговуючого персоналу, отримуєте віддалений доступ з будь-якого робочого місця, безпечне зберігання даних.

1.3 Огляд можливостей платформи

Платформа «1С:Підприємство 8.3» враховує багаторічний досвід застосування системи програм «1С:Підприємство» попередніх версій, які використовують десятки тисяч розробників та на якій працюють сотні тиражних та сотні тисяч замовних прикладних рішень. Завдяки цьому нова версія «1С:Підприємство 8.3» зберегла ідеологічну наступність із попередніми версіями.

Гнучкість платформи дозволяє використовувати систему програм «1С:Підприємство 8.3» для автоматизації обліку та управління на виробничих підприємствах, у бюджетних та фінансових організаціях, підприємствах оптової та роздрібної торгівлі, сфери обслуговування тощо.

Новий сучасний дизайн інтерфейсу забезпечує легкість освоєння для початківців та високу швидкість роботи для досвідчених користувачів:

- можливість масового введення інформації завдяки функції "введення по рядку" та ефективному використанню клавіатури;
- швидке освоєння системи непідготовленими користувачами;
- зручність роботи з великими динамічними списками, управління видимістю та порядком колонок, настроювання відбору та сортування;
- максимальне використання доступного простору екрана для відображення інформації;
- застосування стилів оформлення;
- створення багатомовних прикладних рішень;
- механізм повнотекстового пошуку даних.

Масштабованість системи «1С:Підприємство 8.3» дозволяє працювати як в однокористувальному режимі, так і забезпечує можливість паралельної роботи великої кількості користувачів.

Система «1С:Підприємство 8.3» дозволяє також ефективно працювати зі збільшенням кількості розв'язуваних завдань та обсягу оброблюваних даних. Технологічна платформа «1С:Підприємства 8.3» містить низку механізмів, що оптимізують швидкість роботи прикладних рішень та підтримує такі режими роботи, серед яких однокористувацький варіант для використання у невеликих організаціях або в домашніх умовах; файловий варіант для розрахованої на багато користувачів роботи, що забезпечує простоту установки і експлуатації; клієнт-серверний варіант роботи на основі трирівневої архітектури з використанням кластера серверів «1С:Підприємства 8.х» та СУБД, таких як Microsoft SQL Server, IBM DB2, PostgreSQL. Забезпечується надійне зберігання та ефективна обробка даних за одночасної роботи великої кількості користувачів. Також є варіант використання механізму розподілених інформаційних баз для використання в територіально розподілених системах. Механізм забезпечує ідентичність конфігурацій інформаційних баз і дозволяє виконувати обмін даних без додаткового програмування.

Система «1С:Підприємство 8.3» надає зручні інструменти для адміністрування:

- конфігуратор;
- механізми аутентифікації;
- список користувачів;
- механізм завдань;
- списки загальних інформаційних баз;
- журнал реєстрації та технологічний журнал;
- вивантаження, завантаження, тестування та виправлення інформаційної бази;
- налаштування параметрів інформаційної бази;
- оновлення конфігурації;
- адміністрування клієнт-серверного режиму роботи;

- відновлення файлової бази даних;
- центр управління продуктивністю;
- центр контролю якості.

Конфігуратор входить у стандартне постачання системи «1С:Підприємство 8.3» і є спеціальним режимом запуску системи. Цей режим дозволяє виконувати зміну існуючих прикладних рішень та створення нових, а також виконувати дії щодо адміністрування інформаційної бази.

Однак є ряд механізмів, що не мають безпосереднього відношення до конфігуратора, які покликані також полегшити працю розробника: стандартизація технології розробки прикладних рішень, дерево об'єктів конфігурації, механізм підсистем, групова розробка і т.д.

Відкритість системи «1С:Підприємство 8.х» дозволяє здійснювати інтеграцію практично з будь-якими зовнішніми програмами та обладнанням на основі загальновизнаних відкритих стандартів та протоколів передачі даних:

- текстові документи;
- текстові файли;
- XML документ;
- DBF файли;
- зовнішнє з'єднання;
- Automation Client/Server;
- HTML-документи;
- робота із файлами;
- технологія зовнішніх компонентів;
- макети ActiveDocument;
- робота з Інтернетом;
- Web-розширення;
- механізми обміну даними;

– механізм Web-сервісів;
– механізм XDTO.

Відмітимо ряд механізмів платформи.

Повнотекстовий пошук дозволяє швидко знаходити потрібну інформацію у програмі. Так, пошук на ім'я користувача знайде самого користувача в довіднику користувачів, а також всі документи, де цей користувач фігурував.

Історія даних — механізм, що компактно зберігає історію зміни прикладних даних програми користувачами. За допомогою цього механізму можна гнучко аналізувати зміни даних, порівнювати різні версії даних та відновлювати дані в той стан, який вони мали у вибраній версії.

Механізм копій бази даних призначений прискорення роботи з великими обсягами даних. Він дозволяє налаштувати «1С:Підприємство 8» таким чином, що для необхідних даних (всіх даних додатка або їх підмножини) буде створюватись та оновлюватись копія в інших фізичних базах даних. На копії бази даних програми можна запускати складну аналітичну звітність, не навантажуючи при цьому робочу базу та уникаючи блокувань даних у робочій базі.

Дата акселератора призначена для прискорення роботи з великими обсягами даних. У дата акселераторі база даних повністю розміщена в оперативній пам'яті робочого сервера кластера серверів «1С:Підприємства 8». Це дозволяє часом на порядки прискорити роботу аналітичної звітності. Дата акселератора працює спільно з механізмом копій бази даних.

Технологія зовнішніх компонентів дозволяє створювати програми (зовнішні компоненти) практично будь-якою мовою програмування в ОС Windows, Linux, macOS, які динамічно підключатимуться і тісно взаємодіятимуть із системою «1С:Підприємство 8», розширюючи її можливості.

Об'єкти конфігурації — це складові елементи, «деталі», у тому числі складається будь-яке прикладне рішення.

Вони є проблемно-орієнтовані об'єкти, що підтримуються лише на рівні технологічної платформи. За великим рахунком, завдання розробника полягає в тому, щоб зібрати з цих об'єктів, як з конструктора, необхідну структуру прикладного рішення і потім описати специфічні алгоритми функціонування та взаємодії цих об'єктів, що відрізняються від їх типової поведінки.

Склад об'єктів, що підтримуються технологічною платформою, є результатом аналізу предметних областей використання 1С:Підприємства, та виділення та класифікації використовуваних у цих галузях бізнес-сутностей. Внаслідок цього аналізу розробник може оперувати такими об'єктами як довідники, документи, регістри відомостей, плани рахунків та ін.

Для того щоб стандартизувати та спростити процес розробки та модифікації прикладних рішень, розробнику надається графічний інтерфейс, за допомогою якого він має можливість описати склад об'єктів, що використовуються у конкретному прикладному рішенні.

На підставі цього опису технологічна платформа створить у базі даних відповідні інформаційні структури, і певним чином працюватиме з даними, що зберігаються в цих структурах. Розробнику немає необхідності дбати про те, в яких таблицях, наприклад, повинні розміщуватись дані, яким чином вони будуть модифікуватися або представлятися користувачеві. Всі ці дії платформа буде виконувати автоматично, виходячи з типової поведінки об'єктів, що використовуються.

Таким чином, розробник оперує метаданими - "даними про дані", або об'єктами конфігурації. Додаючи в структуру прикладного рішення черговий об'єкт конфігурації, розробник, по суті, додає опис того, як розміщуватимуться відповідні дані, і як вони взаємодіятимуть з іншими даними, що зберігаються в інформаційній базі.

Склад об'єктів, які може використовувати розробник, фіксовано та визначено на рівні платформи. Розробник неспроможна створювати власні види об'єктів, може

оперувати лише тим набором об'єктів, який є. Подібний підхід до розробки прикладних рішень дозволяє, по-перше, стандартизувати процес розробки, а по-друге, забезпечити просту та швидку модифікацію прикладних рішень іншими розробниками або користувачами.

Поговоримо про вбудовану мову, що використовується при розробці конфігурацій.

Вбудована мова є важливою частиною технологічної платформи "1С:Підприємства 8", оскільки дозволяє розробнику описувати власні алгоритми функціонування прикладного рішення.

Вбудована мова має багато спільних рис з іншими мовами, такими як Pascal, Java Script, Basic, що полегшує його освоєння розробниками-початківцями. Однак він не є прямим аналогом будь-якої з перелічених мов.

Перелічимо найбільш значущі особливості вбудованої мови.

Попередня компіляція – перед виконанням модулі, що містять текст вбудованою мовою, перетворюються на внутрішній код. Кешування скомпілованих модулів у пам'яті. М'яка типізація – тип змінної визначається типом значення, яке вона містить, і може змінюватися у процесі роботи. Відсутність програмного опису об'єктів конфігурації – розробник може використовувати або вбудовані в платформу об'єкти, або об'єкти, створені системою внаслідок візуального конструювання прикладного рішення. Подієва орієнтованість вбудованої мови.

Призначення вбудованої мови у системі 1С:Підприємство визначається ідеологією створення прикладних рішень. Прикладні рішення в 1С:Підприємстві 8 не кодуються повністю. Більшість прикладного рішення створюється розробником шляхом візуального конструювання — створення нових об'єктів конфігурації, завдання їх властивостей, форм представлення, взаємозв'язків та інші. Вбудована мова використовується лише для того, щоб визначити поведінку об'єктів прикладного рішення, відмінну від типового, та створити власні алгоритми обробки даних .

Тому модулі, що містять текст вбудованою мовою, використовуються системою в конкретних, заздалегідь відомих ситуаціях, які можуть виникнути в процесі роботи прикладного рішення. Такі ситуації називаються подіями. Події можуть бути пов'язані з функціонуванням об'єктів прикладного рішення або самим прикладним рішенням, як таким.

Наприклад, з функціонуванням об'єкта прикладного рішення Довідник пов'язаний ряд подій, серед яких є подія.

Ця подія виникає безпосередньо перед тим, як дані елемента довідника мають бути записані до бази даних. Розробник, використовуючи вбудовану мову, може описати алгоритм, який, наприклад, перевірятиме коректність даних, введених користувачем. Розмістивши цей алгоритм у відповідному модулі, розробник забезпечить те, що кожного разу, як користувач виконуватиме запис елемента довідника, система виконуватиме створений розробником алгоритм і перевірятиме, чи не забув користувач заповнити обов'язкові реквізити довідника [9].

Таким чином можна сказати, що вбудована мова є скриптовою мовою для програмування бізнес-логіки, а використання модулів вбудованою мовою є подієво-залежним, тобто виконання модулів відбувається при виникненні певних подій у процесі функціонування прикладного рішення.

2 Огляд вбудованої мови

Об'єкти конфігурації — це складові елементи, «деталі», у тому числі складається будь-яке прикладне рішення.

Вони є проблемно-орієнтовані об'єкти, що підтримуються лише на рівні технологічної платформи. За великим рахунком, завдання розробника полягає в тому, щоб зібрати з цих об'єктів, як з конструктора, необхідну структуру прикладного рішення і потім описати специфічні алгоритми функціонування та взаємодії цих об'єктів, що відрізняються від їх типової поведінки.

Склад об'єктів, що підтримуються технологічною платформою, є результатом аналізу предметних областей використання 1С:Підприємства, та виділення та класифікації використовуваних у цих галузях бізнес-сутностей. Внаслідок цього аналізу розробник може оперувати такими об'єктами як довідники, документи, регістри відомостей, плани рахунків та ін.

Для того щоб стандартизувати та спростити процес розробки та модифікації прикладних рішень, розробнику надається графічний інтерфейс, за допомогою якого він має можливість описати склад об'єктів, що використовуються у конкретному прикладному рішенні.

На підставі цього опису технологічна платформа створить у базі даних відповідні інформаційні структури, і певним чином працюватиме з даними, що зберігаються в цих структурах. Розробнику немає необхідності дбати про те, в яких таблицях, наприклад, повинні розміщуватись дані, яким чином вони будуть модифікуватися або представлятися користувачеві. Всі ці дії платформа буде виконувати автоматично, виходячи з типової поведінки об'єктів, що використовуються.

Таким чином, розробник оперує метаданими - "даними про дані", або об'єктами конфігурації. Додаючи в структуру прикладного рішення черговий об'єкт конфігурації, розробник, по суті, додає опис того, як розміщуватимуться відповідні

дані, і як вони взаємодіятимуть з іншими даними, що зберігаються в інформаційній базі.

Склад об'єктів, які може використовувати розробник, фіксовано та визначено на рівні платформи. Розробник неспроможна створювати власні види об'єктів, може оперувати лише тим набором об'єктів, який є. Подібний підхід до розробки прикладних рішень дозволяє, по-перше, стандартизувати процес розробки, а по-друге, забезпечити просту та швидку модифікацію прикладних рішень іншими розробниками або користувачами.

2.1 Основні об'єкти вбудованої мови

Вбудована мова підтримує роботу з великою кількістю різноманітних об'єктів. Безперечно, основну групу об'єктів складають прикладні об'єкти, що дозволяють описувати алгоритми функціонування бізнес-логіки.

Однак, не менш важливою групою є об'єкти, призначені для зберігання тимчасових наборів даних протягом сеансу роботи користувача. Як правило, вони служать для допоміжного збору, угруповання, аналізу та обробки інформації [10].

Коротко їх перерахуємо.

Масив. Є пронумерованою колекцією значень довільного типу. До елемента масиву можна звертатися за його індексом. Як елементи масиву можуть виступати, зокрема, інші масиви. Це дозволяє створювати багатовимірні масиви.

Структура. Являє собою названу колекцію, що складається з пар «ключ – значення». Ключ може бути лише рядковим, значення – довільного типу. До елементу структури можна звертатися за значенням його ключа, тобто на ім'я. Зазвичай використовується для зберігання невеликої кількості значень, кожне з яких має унікальне ім'я.

Відповідність. Так само як і структура, є колекцією пар ключ — значення. Проте, на відміну структури, ключ може бути практично будь-якого типу.

Список значень. Використовується, зазвичай, на вирішення інтерфейсних завдань. Дозволяє будувати динамічні набори значень та маніпулювати ними (додавати, редагувати, видаляти елементи, сортувати). Він може містити значення будь-якого типу, крім того, в одному списку типи збережених значень можуть бути різними. Наприклад, список значень можна використовувати для вибору конкретного документа зі списку можливих документів, сформованого за складним алгоритмом.

Таблиця значень. Таблиця значень дозволяє будувати динамічні набори значень та маніпулювати ними. Вона може бути наповнена значеннями будь-якого типу, і в одній таблиці типи значень, що зберігаються, можуть бути різними. Одним із прикладів використання таблиці значень може бути організація уявлення у формі списку елементів довідника, відібраних за складним алгоритмом.

Дерево значень. При його створенні динамічно формується набір значень будь-якого типу, схожий на таблицю значень. На відміну від таблиці значень, рядки дерева значень можуть утворювати ієрархічні структури: кожен рядок дерева може мати набір підлеглих рядків, кожен із підлеглих рядків, у свою чергу, також може мати набір підлеглих рядків і так далі. При цьому пошук значень, сортування, отримання результатів можуть здійснюватися або за поточним рівнем ієрархії або включаючи всі підлеглі.

COMSafeArray. Є об'єктною оболонкою над багатовимірним масивом SAFEARRAY з COM. Дозволяє створювати та використовувати SAFEARRAY для обміну даними між COM-об'єктами.

Фіксований масив. Незмінний масив. Масив заповнюється системою при ініціалізації об'єктів даного типу або розробником за допомогою конструктора.

2.2 Огляд редактора програмного модуля

Для створення та зміни текстів вбудованою мовою розробник може використовувати редактор тексту та модуля, що володіє зручними засобами створення, редагування та синтаксичної перевірки модулів [11].

Редактор тексту та модуля – це один із інструментів розробки. Він використовується для редагування текстових документів та для редагування програмних модулів конфігурації.

Редактор текстів та модулів надає користувачеві всі основні функції, необхідні при редагуванні як простих, так і текстів програмних модулів. Конфігуратор 1С:Підприємства 8 використовує цей редактор у двох режимах:

– для редагування текстових документів;
– для редагування текстів модулів (як складника редактора форми).

У режимі редагування текстових документів редактор підтримує всі стандартні функції редагування тексту:

– створення нового документа чи відкриття одного з існуючих документів;
– введення та редагування тексту;
– збереження відредагованого тексту;
– друк тексту.

У процесі редагування тексту можна переходити до певного рядка документа, зрушувати блоки тексту на позицію табуляції, шукати та замінювати та використовувати закладки. Закладки можуть бути розміщені на будь-якому рядку тексту (рис. 2.1):

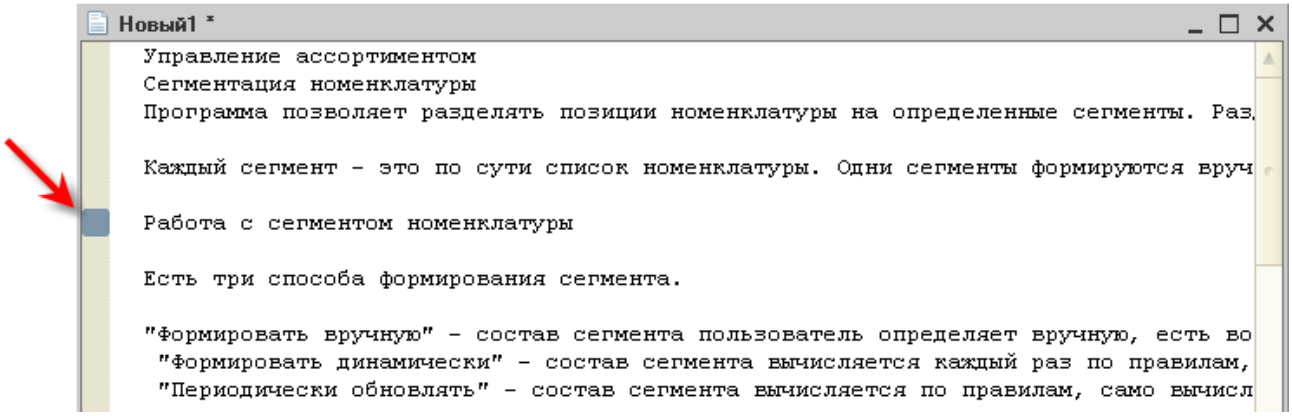

Рисунок 2.1 – Розміщення закладок

Редагування текстів модулів може виконуватися в процесі створення форми об'єкта прикладного рішення, безпосередньо при розробці модулів об'єктів або всієї програми, і при редагуванні зовнішнього текстового файлу, що містить текст модуля.

Крім стандартних дій, властивих будь-якому текстовому редактору, редактор текстів та модулів має низку специфічних особливостей:

Виділення кольором синтаксичних конструкцій

Для зручності редагування текстів модулів редактор виділяє кольором елементи вбудованої мови: ключові слова, мовні константи, оператори, коментарі та ін (рис. 2.2).

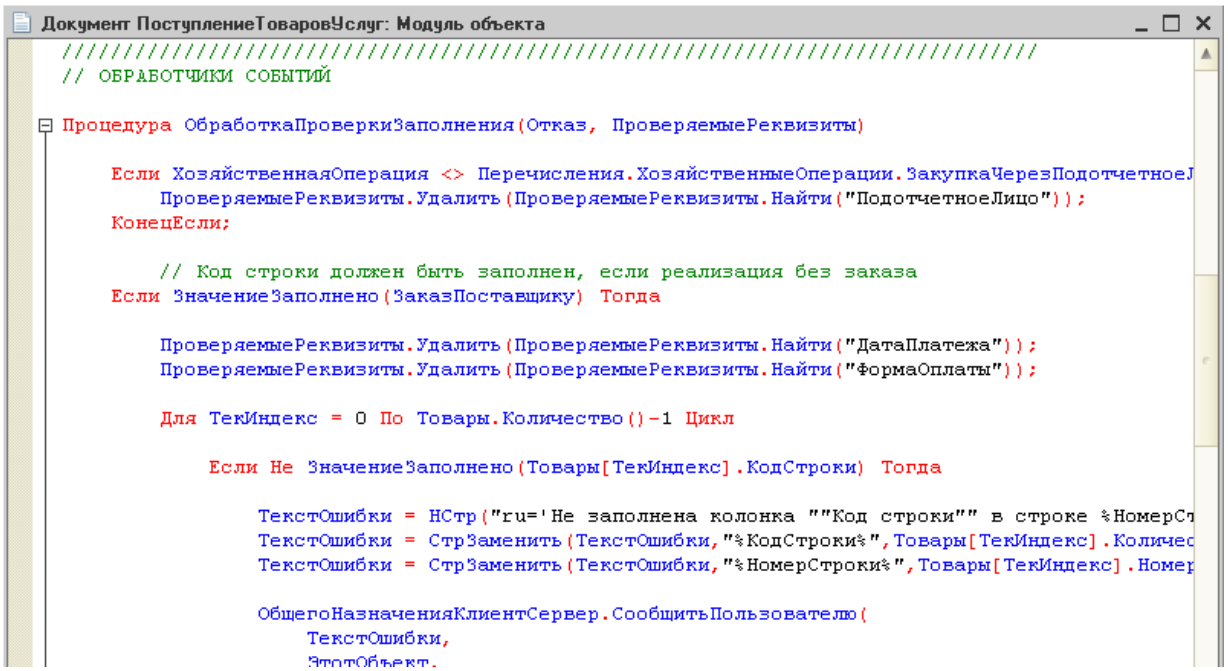

Рисунок 2.2 - Виділення кольором синтаксичних конструкцій

Розробник може використовувати кольори виділення, встановлені за замовчуванням, або налаштувати їх самостійно. У випадку система сама відслідковує необхідність включення режиму виділення кольором. Однак у ситуації, коли система не знає про те, що редагується текст модуля (наприклад, якщо редагується зовнішній текстовий файл, що містить текст модуля), розробник може включити режим виділення кольором вручну, використовуючи меню конфігуратора (рис. 2.3):

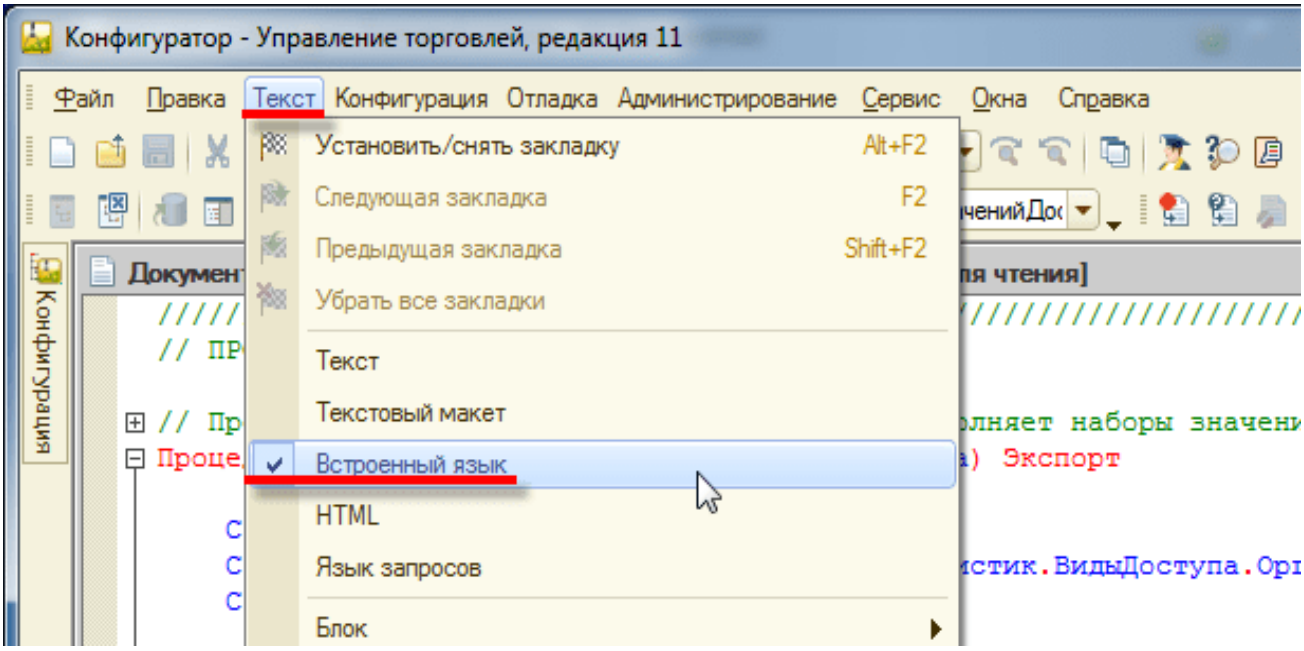

Рисунок 2.3 – Ввімкнення режиму виділення синтаксичних конструкцій

При перегляді модулів редактор дозволяє об'єднувати деякі синтаксичні конструкції мови групи, згортати і розгортати їх. Використання групування синтаксичних конструкцій дозволяє краще сприймати різні частини тексту, а також переносити та копіювати групи повністю (рис. 2.4):

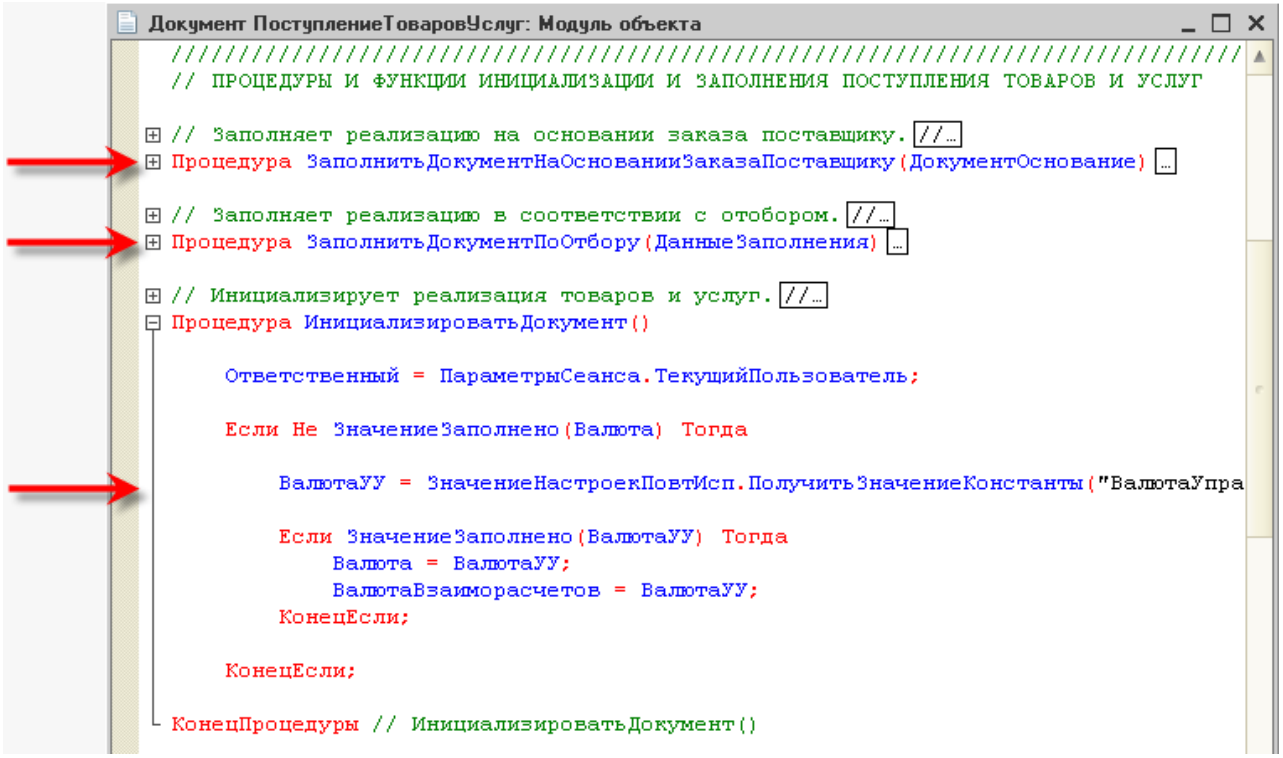

Рисунок 2.4 - Угрупування

Згорнутий текст заміщається спеціальним маркером, який дозволяє переглянути вміст згорнутої групи як підказки (рис. 2.5):

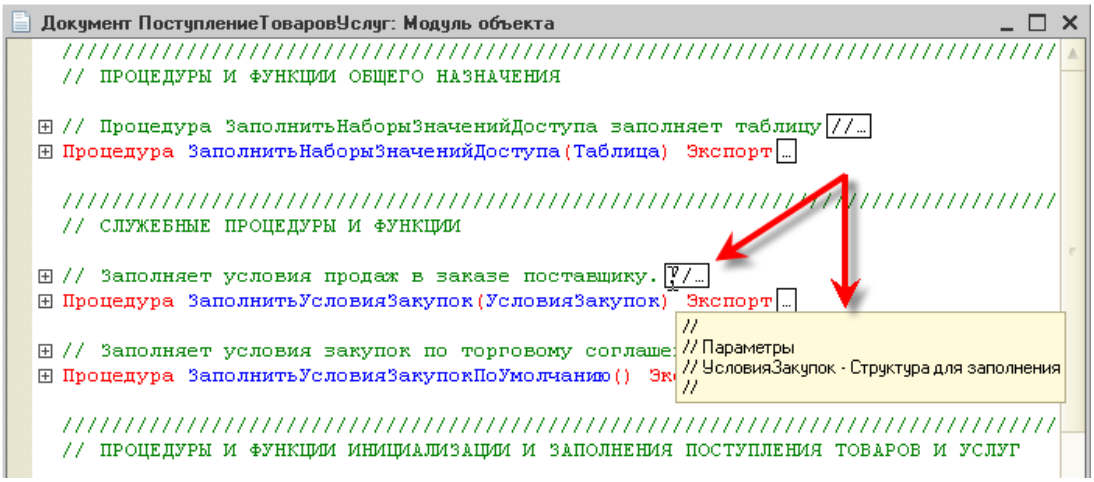

Рисунок 2.5 – Підказка згорнутої групи

Розробнику надається можливість налаштовувати режим угруповання, вказуючи, які синтаксичні конструкції можуть групуватися, і яким має бути вихідний стан угруповання (згорнутий або розгорнутий) при відкритті документа. Таким чином, він може налаштувати, наприклад, використання угруповань «максимум» (рис. 2.6):

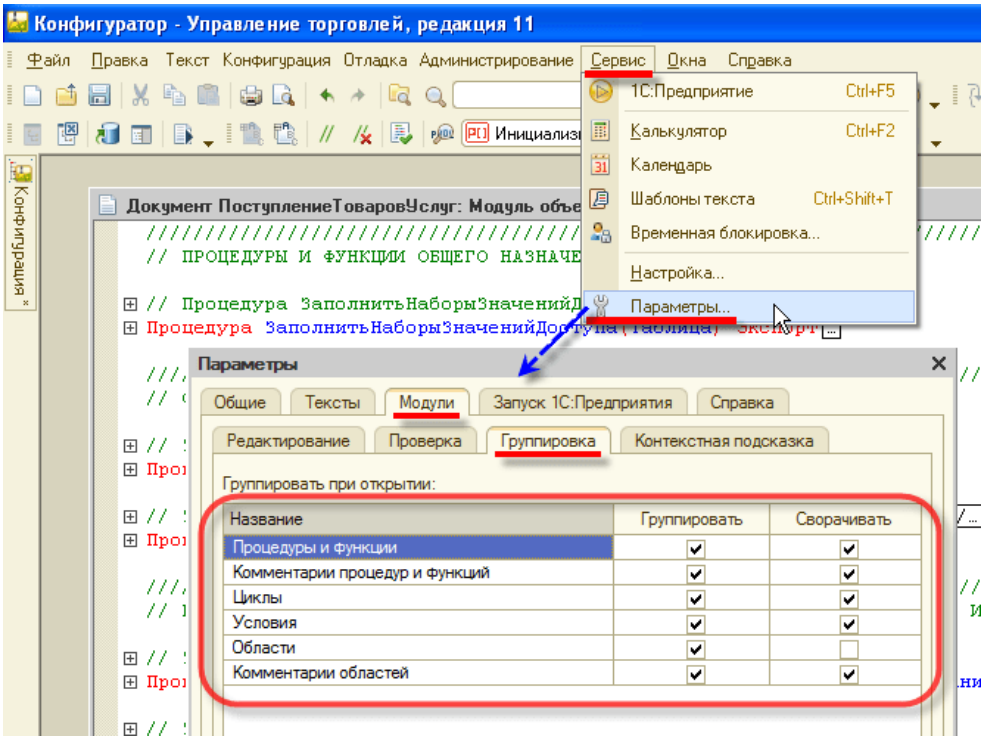

Рисунок 2.6 – Налаштування групувань

Розробник може виділяти довільні області тексту, групувати та згортати їх подібно до того, як згортаються інструкції циклів, умов, процедур та функцій.

Кожній області тексту, яку виділяє розробник, може дати власне ім'я. Це дозволяє простим і зрозумілим чином виділяти частини модуля, що мають схожий зміст (рис. 2.7).

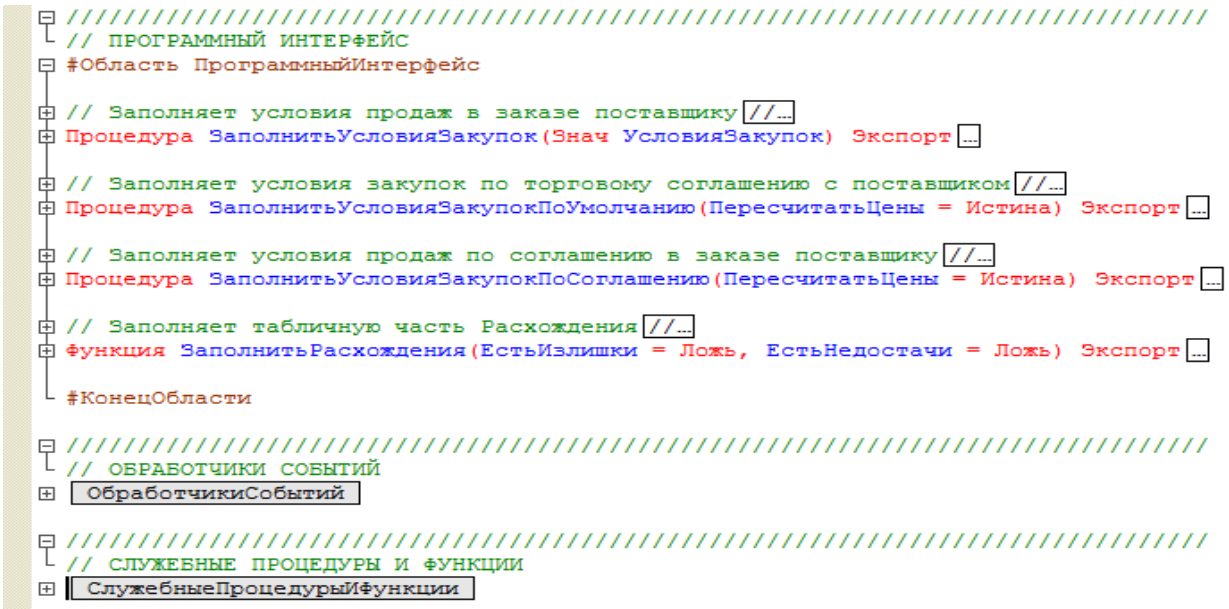

Рисунок 2.7 – Використання областей модулів

Області виділяються за допомогою двох інструкцій препроцесора: #Область та #КонецОбласти. Єдине призначення цих інструкцій - позначити рядки модуля, що групуються і згортаються.

Області можуть бути вкладені одна в одну або в інші конструкції мови, що групуються.

Редактор дозволяє виконувати ряд операцій над виділеними блоками тексту модуля: автоматичне форматування, зміна відступу, додавання/видалення коментарів та перенесення рядків (рис. 2.8).

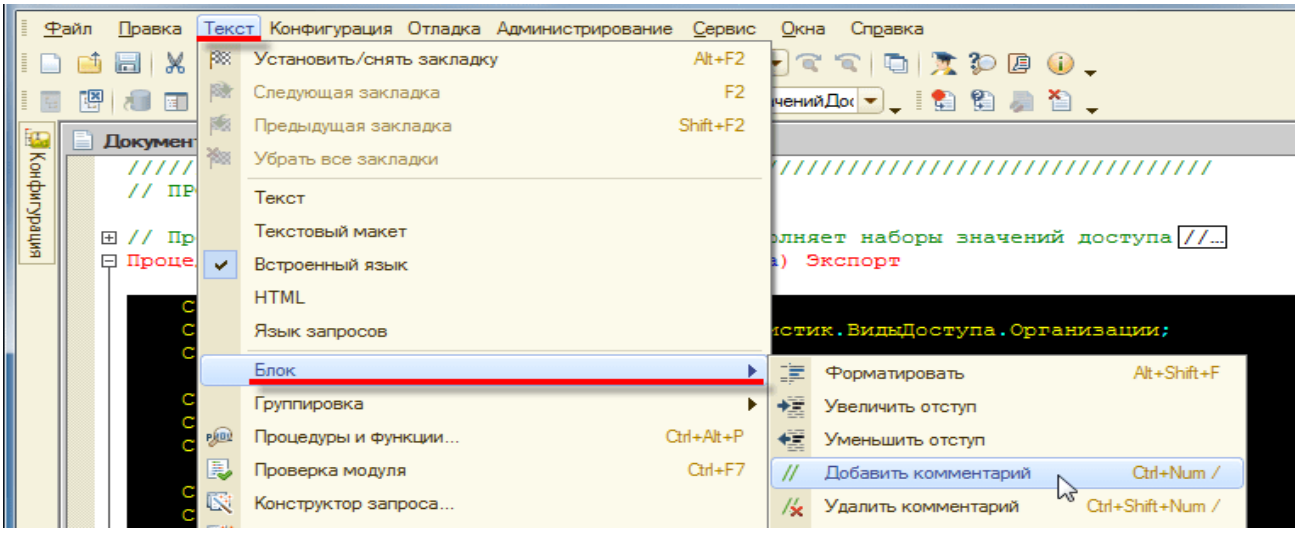

Рисунок 2.8 – Операції із блоками

Хорошим стилем написання модулів вважається використання синтаксичного відступу виділення управляючих конструкцій вбудованої мови. Редактор дозволяє автоматично форматувати текст під час його введення, а також виконувати автоматичне форматування вже введеного тексту.

Початковий текст (рис. 2.9):

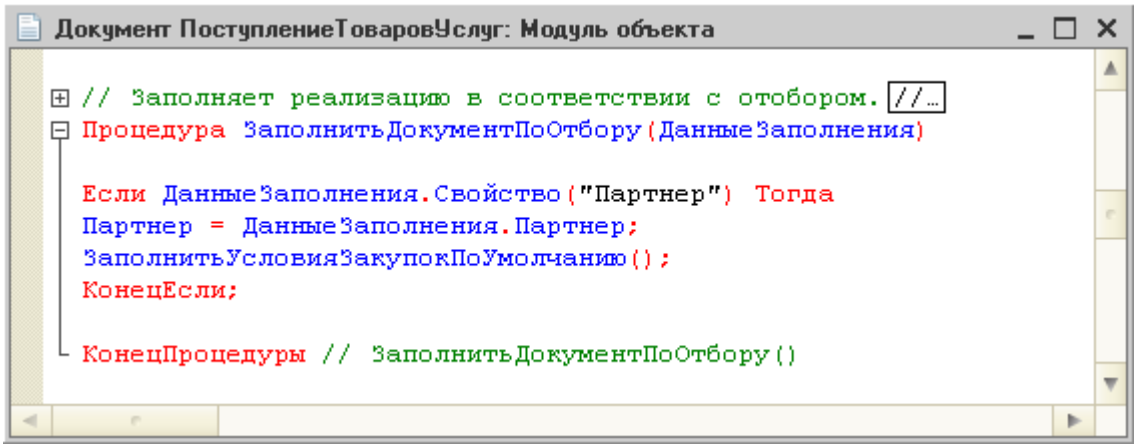

Рисунок 2.9 – Вихідний текст

Результат автоматичного форматування (рис. 2.10):

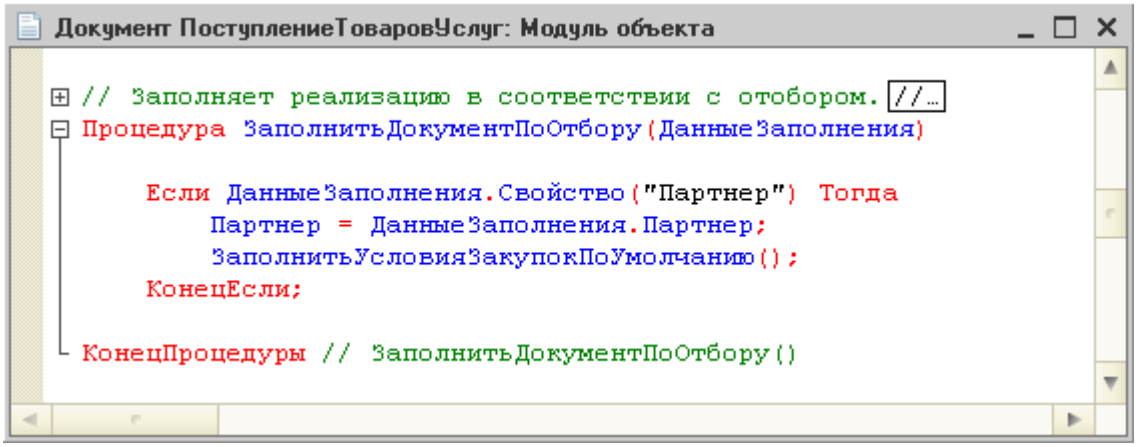

Рисунок 2.10 – Відформатований текст

Поряд з автоматичним форматуванням всього виділеного тексту редактор підтримує також операції зсуву виділеного блоку вправо або вліво на крок табуляції. Це полегшує ручне форматування великих фрагментів коду.

Також редактор містить дуже зручну для розробника функцію автоматичної (одним натисканням миші) встановлення та зняття коментарів на виділений текст. Така можливість часто використовується при налагодженні модулів (рис. 2.11):

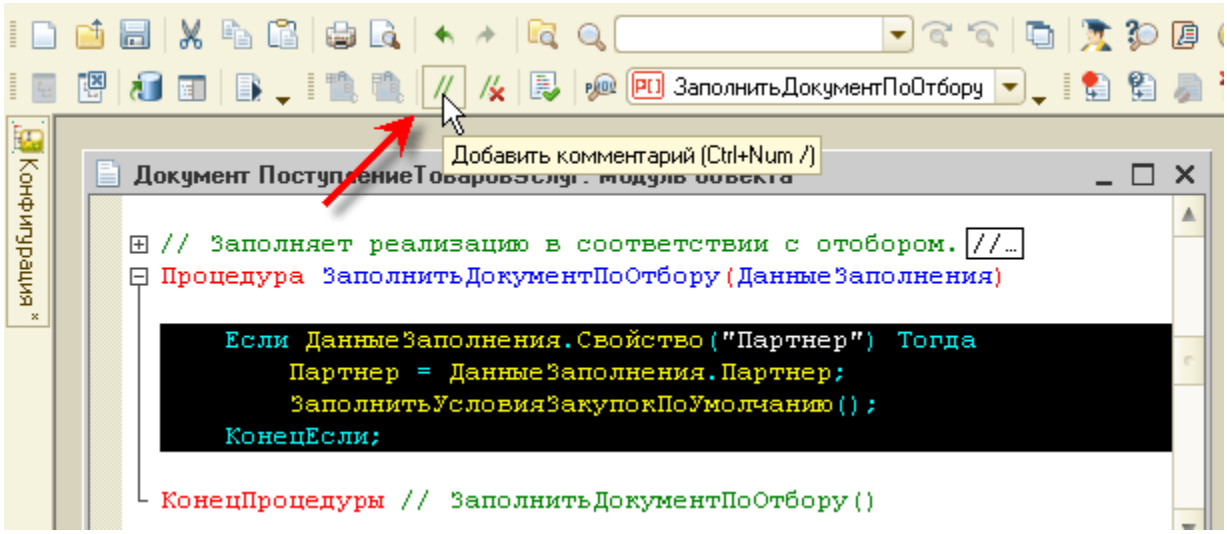

Рисунок 2.11 – Додавання коментарів

Використання додавання та видалення переносу рядка часто застосовується при перенесенні текстів запиту між модулем та, наприклад, консоллю запитів.

Таким чином, налагодивши запит у консолі запитів, розробник може просто скопіювати текст запиту з консолі, вставити його в модуль і одним рухом додати перенесення рядка до всіх рядків тексту [12].

У випадку, коли модуль містить велику кількість процедур та функцій, зручно використовувати режим пошуку процедур, який підтримується редактором. Процедури та функції відображаються в окремому вікні в порядку їх розташування в модулі, однак розробник може відсортувати їх за абеткою. Піктограми ліворуч від назви позначають наявні процедури та функції, а імена у кутових дужках відповідають визначеним процедурам, які зараз відсутні, але можуть бути розміщені в даному модулі (рис. 2.12).

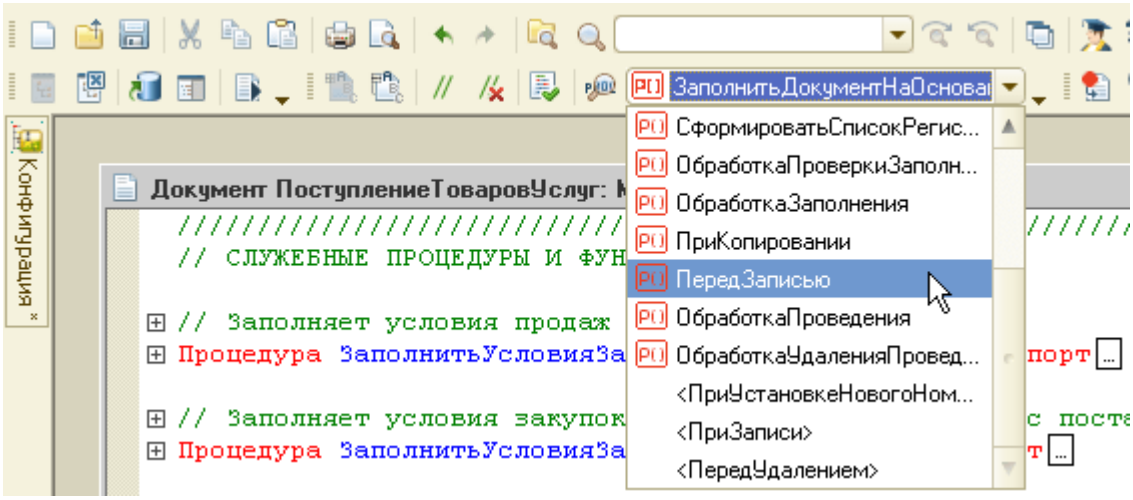

Рисунок 2.12 – Перехід за процедурами та функціями

Якщо встановити курсор на тій процедурі, яка ще відсутня в модулі, та натиснути Перейти, конструктор автоматично вставить у текст модуля заголовок наперед визначеної процедури.

Редактор дозволяє автоматично переходити до визначення процедури або функції, що використовується в тексті модуля. Для цього достатньо встановити курсор на ім'я потрібної функції в тілі модуля та виконати команду контекстного меню або натиснути гарячу клавішу. У вікні редактора буде відкрито текст необхідної процедури або функції (рис. 2.13):

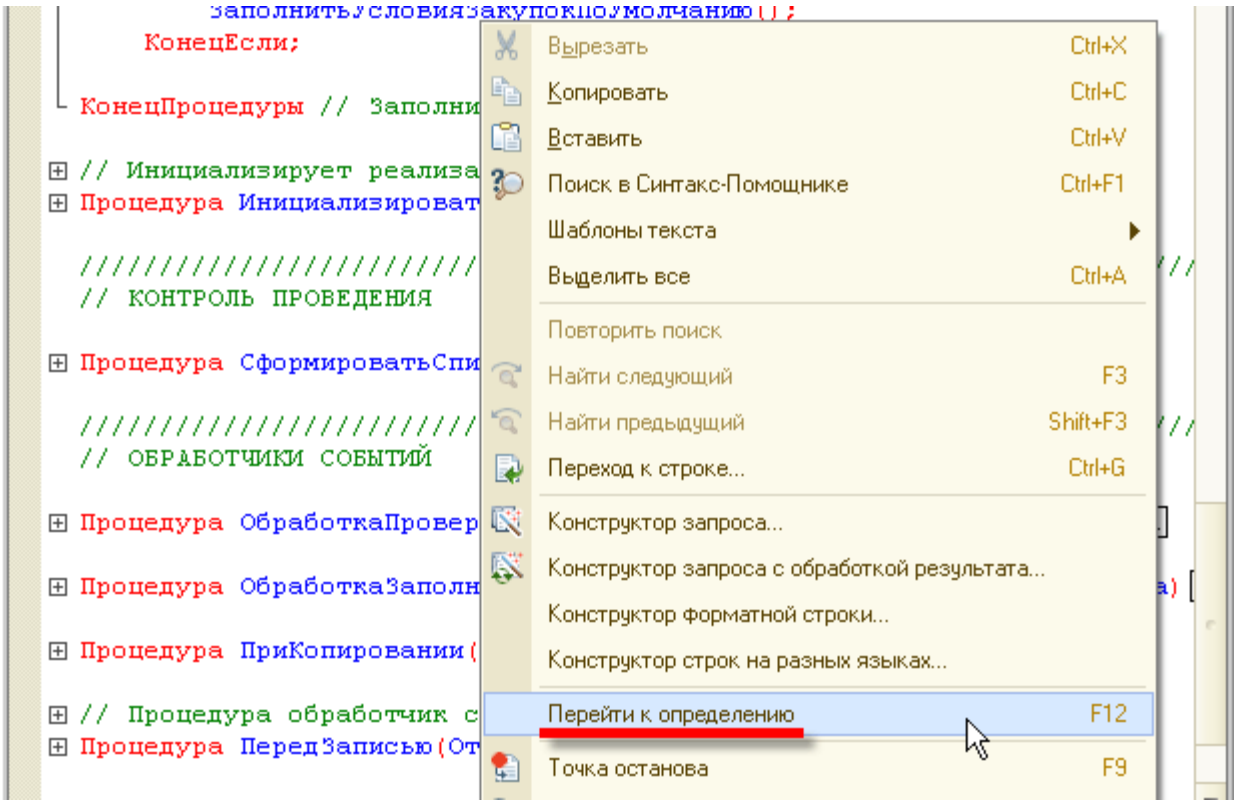

Рисунок 2.13 – Перехід до визначення функції та процедури

Редактор надає засіб контекстного введення виразів з використанням системних об'єктів, їх властивостей, методів тощо. правильно набирати тексти модулів (рис. 2.14):

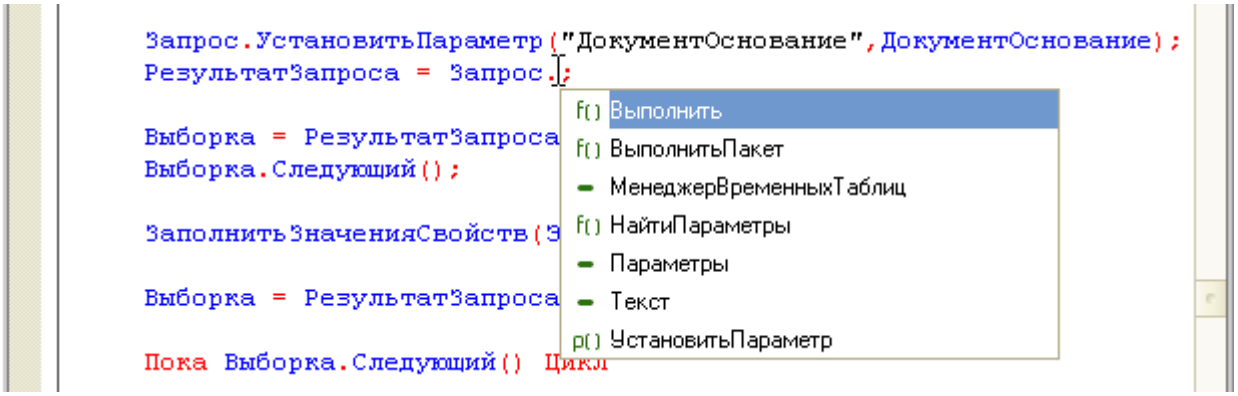

Рисунок 2.14 – Контекстна підказка

Контекстна підказка також працює для параметрів деяких методів, якщо ці параметри задаються рядковими літералами.

При написанні тексту модуля розробник може просто перетягувати мишею імена об'єктів або їх реквізитів з дерева метаданих у потрібне місце модуля.

Модуль, що редагується, може бути перевірений на правильність використання синтаксичних конструкцій вбудованої мови, коректність звернень до методів і властивостей об'єктів «через точку», а також на коректність деяких параметрів, що мають тип «Строка» (рис. 2.15) [14]:

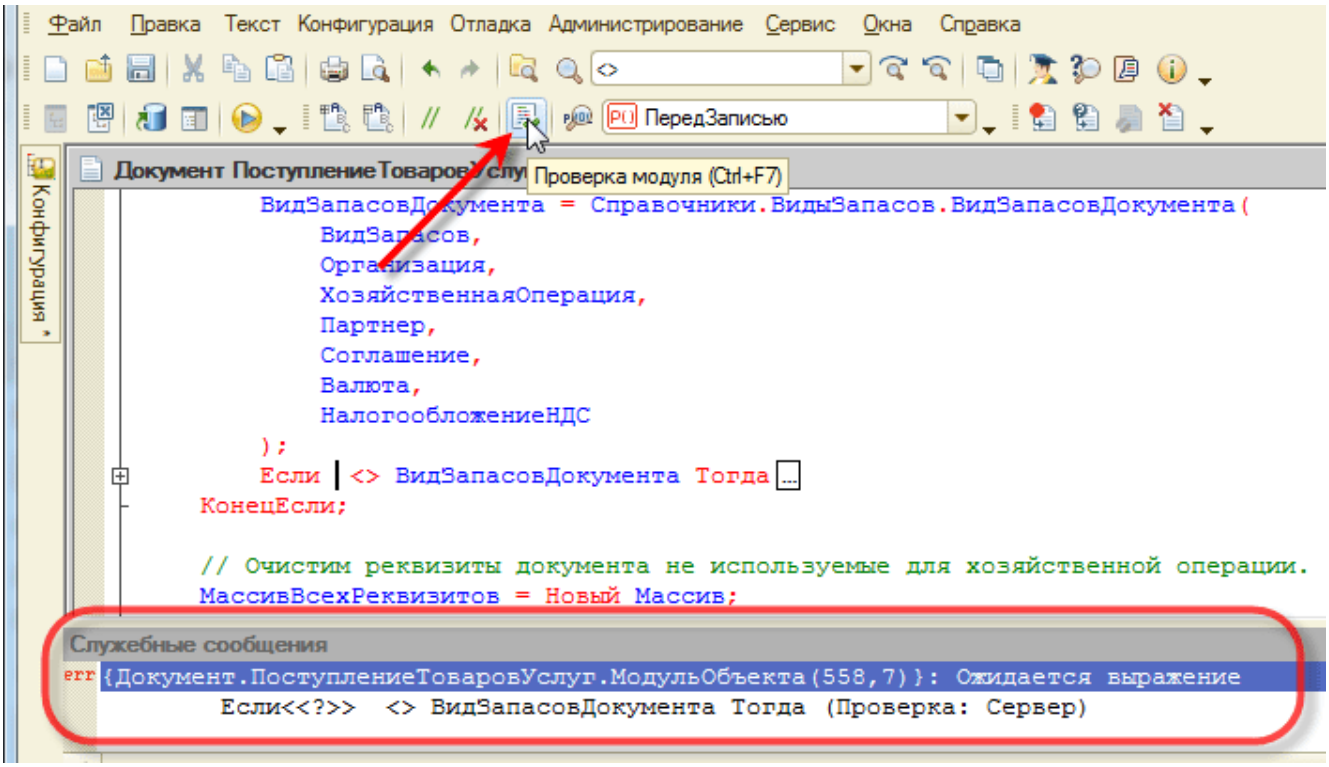

Рисунок 2.15 – Перевірка модуля на коректність

За наявності помилок у модулі їх список видається у вікні стану. Клацнувши мишею на повідомленні про помилку, можна перейти до рядка модуля, що спричинило помилку. За бажанням розробник може увімкнути автоматичне виконання синтаксичного контролю модуля за його закритті чи збереженні всієї конфігурації.

Крім цього, конфігуратор підтримує виконання повної перевірки всіх модулів, що містяться в прикладному рішенні.

У процесі роботи з модулем розробник має можливість отримувати контекстну підказку з вбудованої мови, використовуючи синтакс-помічник. Для цього достатньо встановити курсор на елемент мови, що цікавить, і натисканням комбінації клавіш

(або по контекстному меню) перейти до опису цього елемента мови в синтакс-помічнику.

Більшість модулів прикладного рішення можна встановити пароль доступу, який захищає авторські права розробника конфігурації. При спробі відкрити захищений модуль виводиться діалог уведення пароля (рис. 2.16):

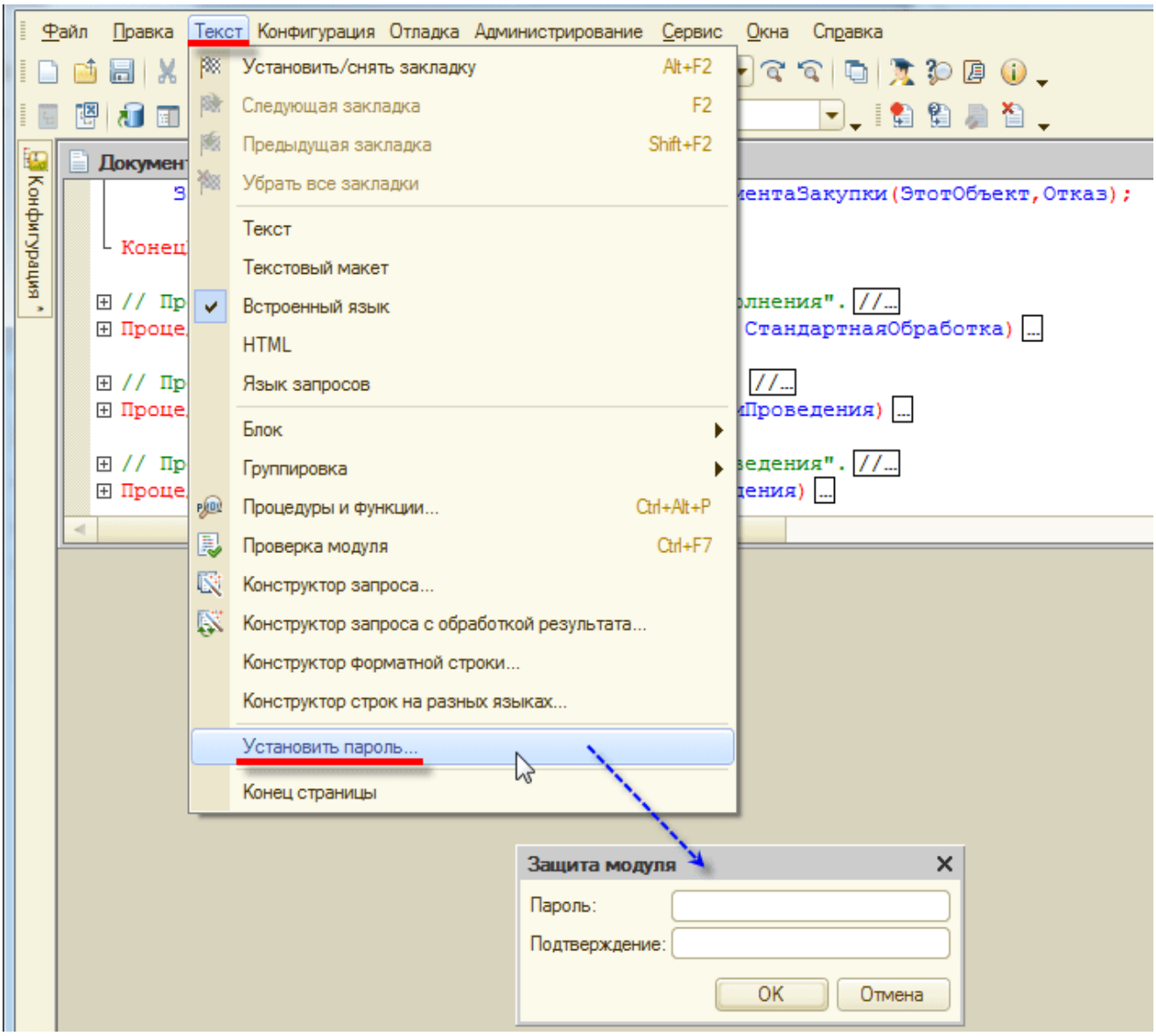

Рисунок 2.16 – Встановлення пароля модуля

Під час редагування текстових документів і модулів конфігуратор надає розробнику можливість використовувати механізм шаблонів для автоматичного встановлення часто використовуваних фрагментів тексту.

2.3 Визначення головних функцій додатку

Основною нашою задачею буде систематизація навчальних матеріалів та подання їх у приємному інтерфейсі «Таксі». Не дивлячись на те, що багато компаній використовують старий інтерфейс, який часто називають звичайними формами, ми все ж таки будемо іти у ногу із трендами. Так як вище згадані організації не переходять на більш юзабіліті інтерфейс через складнощі з переносом даних. А так як ми не маємо потреби це робити, то й не будемо.

Систематизація – це дія, або посладовність дій, унаслідок виконання яких певна множина розрізнених елементів зводиться у систему елементів між якими є зв'язки. В свою чергу, елемент – це не випадкові компоненти предмета, а тільки такі, об'єднання яких дає змогу одержати цей предмет. Необхідною умовою систематизації є представлення фундаментальної мети, що дає змогу даним досягнути логічного поєднання.

Провівши процес систематизації, ми наділяємо дані певними властивостями:

- підвищення продуктивності роботи з ними;
- поліпшення взаємодії з інформацією;
- полегшення пошуку потрібної інформації;
- сприяння детальнішому ознайомленню та дослідженню;
- ідентифікування й усунення протирічь, пропусків, непогодженностей;
- покращення пізнавального процесу формування людської свідомості.

Систематизація бере за основу результати процесів аналізу, класифікації та синтезу певних систем. Дослідження ознак і властивостей елементів (аналіз), підпорядкування одного елемента іншому по заздалегідь визначеній характеристиці (класифікація) та згруповування їх у єдину цілісність (синтез) – успішне наслідування цих етапів дозволяє якісно провести систематизацію наших даних [20].

Взагалі, якщо обговорювати повномасштабну картину нашого додатку, то в ньому повинні знаходитися файли усіх навчальних матеріалів дисциплін. В процесі систематизації вони приймаються за множину об'єктів. Першочергово, потрібно

проаналізувати всі файли та визначити їх основні характеристики. Наступним кроком є виявлення спільних ознак, щоб в подальшому обрати критерій, за яким будуть класифікуватися ці дані. Останнім етапом буде синтез – об’єднання матеріалів за визначеними категоріями.

Якщо аналізувати навчальні матеріали, можна виявити ряд характеристик. Це те, що серед них є такі, що відповідають певній дисципліні та поділяються на лекційні матеріали, практичні та лабораторні роботи. При побудові класифікації ми будемо користуватися операцією обмеження. Операція обмеження – це процес розділення основної множини об’єктів на менші підмножини. Тобто рух при створенні класифікації проходить у напрямі від найзагальнішого поняття (множина всіх навчальних матеріалів), до проміжного (підмножини матеріалів конкретних дисциплін), та нарешті одиничного (підмножини лекцій, практик та лабораторних). Далі ділити поняття безглуздо, так як одиничний елемент не можна розділити на ще менший, окрім додавання до нього порожніх понять. Отже, прийнявши саме такий вид класифікації, синтезуємо дані. Після цього, вже систематизовані дані можна приміняти при створенні додатку [22].

Окрім систематизації приділимо нашу увагу до розподілення прав для користувачів. Так, зробимо ролі студента, який матиме змогу тільки читати навчальні матеріали, викладача – який зможе завантажити лекції, практичні та лабораторні роботи та адміністратора, за яким залишимо змогу створювати користувачів.

Так як ми розробляємо додаток у першу чергу для напряму Комп’ютерної інженерії, збір початкових даних будемо використовувати серед цих дисциплін.

Перерахованого вище буде достатньо для нашої першочергової цілі, розширенням функціоналу можна заянятися згодом.

2.4 Вибір комплектуючих для коректної роботи програмного забезпечення

Фірма «1С» проводить сертифікацію комп'ютерів щодо сумісності із системою програм «1С:Підприємство» версій 8.3.

Комп'ютери, що пройшли сертифікацію, отримують сертифікат, що засвідчує їхню сумісність із системою програм «1С:Підприємство».

До сертифікації приймаються комп'ютери з характеристиками не гіршими за ті, що представленні нижче [15].

Системні вимоги для 32-розрядного сервера 1С:Підприємства:

- процесор Intel Pentium IV/Xeon 2,4 ГГц та вище;
- оперативна пам'ять 1024 Мб та вище;
- жорсткий диск 40Гб та вище;
- пристрій для читання компакт-дисків;
- USB-порт;
- SVGA-відеокарта.

Системні вимоги для 64-розрядного сервера 1С:Підприємства:

- процесор з архітектурою x86-64 (Intel із підтримкою EM64T, AMD з підтримкою AMD64);
- оперативна пам'ять 2048 Мб та вище;
- жорсткий диск 40Гб та вище;
- пристрій для читання компакт-дисків;
- USB-порт;
- SVGA-відеокарта.

Системні вимоги до сервера баз даних:

- технічні характеристики комп'ютера та операційна система повинні відповідати вимогам Microsoft SQL Server, PostgreSQL, IBM DB2, Oracle Database.

Товстий клієнт:

- процесор Intel Pentium Celeron 2400 МГц та вище;
- оперативна пам'ять 1024 Мб та вище;
- жорсткий диск 40Гб та вище;
- пристрій для читання компакт-дисків;
- USB-порт;
- SVGA-відеокарта.

Тонкий клієнт:

- процесор Intel Pentium Celeron 1800 МГц та вище;
- оперативна пам'ять 256 Мб та вище;
- жорсткий диск 40Гб та вище;
- пристрій для читання компакт-дисків;
- USB-порт;
- SVGA-відеокарта.

Веб-клієнт:

- процесор Intel Pentium Celeron 1800 МГц та вище;
- оперативна пам'ять 256 Мб та вище;
- жорсткий диск 40Гб та вище;
- пристрій для читання компакт-дисків;
- SVGA-відеокарта.

Останнім часом необхідність у пристрою для читання компакт-дисків та USB-порту відпала через те, що програмні продукти разом із ліцензіями постачаються у цифровому вигляді, хоча ситуації коли софт купується на фізичних носіях все ще трапляються.

Як ми бачимо, технічні вимоги до програмних продуктів 1С досить низкі, тим паче, що у наших планах немає частого використання додатку для запису, у більшості випадків буде читання даних, тому необхідності змінювати апаратну складову немає.

2.5 Підведення проміжних висновків

Озираючись назад на ознайомлення з необхідною у подільшій роботі інформацією, можна підвести проміжні висновки, а саме звернути увагу на основні моменти про які потрібно пам’ятати:

- ми створюємо додаток для систематизації матеріалів, які будуть використовуватися у процесі навчання у аудиторіях, на десктопних комп’ютерах;
- провівши аналіз навчальних приміщень Державного університету телекомунікацій, на більшості комп’ютерів було виявлено операційну систему Windows, тому саме на цю платформу ми будемо орієнтуватись;
- першочергово, ми маємо на меті систематизувати навчальні матеріали дисциплін напряму Комп’ютерної інженерії, тому в якості першопроходців будуть обрані предмети саме серед цих представників;
- отримавши файли з навчальними матеріалами, необхідно їх систематизувати, класифікувавши на групи лекцій, практичних та лабораторних робіт, при цьому встановивши їх згідно обраному учбовим відділом порядком вивчення студентами;
- сам код буде виконаний мовою програмування 1С;
- в свою чергу за графічний інтерфейс користувача буде відповідати найбільш поширений на цей момент інтерфейс «Таксі».

Маючи на озброєнні ці відомості, можна приступати до практичної частини дипломної роботи.

3 Реалізація удосконаленого апаратно-програмного комплексу

Нарешті, ознайомившись із теоретичними матеріалами, ми можемо приступати безпосередньо до практичної частини нашої магістерської роботи.

3.1 Створення інформаційної бази

Щоб створити нову чисту інформаційну базу, у вікні запуску 1С:Підприємство 8.3 зі списком інформаційних баз натисніть кнопку Додати.

У вікні буде запропоновано два варіанти додавання інформаційної бази:

- створення нової інформаційної бази;
- додавання до списку наявної інформаційної бази.

Виберіть "Створення нової інформаційної бази" та натискаємо кнопку "Далі".

На наступному етапі також буде запропоновано два варіанти створення інформаційної бази:

- створення інформаційної бази із шаблону;
- створення інформаційної бази без конфігурації для розробки нової конфігурації або завантаження раніше завантаженої інформаційної бази.

Так як нам потрібна чиста база, тому ми обираємо пункт створення інформаційної бази без конфігурації для розробки нової конфігурації або завантаження раніше завантаженої інформаційної бази.

У наступному вікні вказуємо найменування вашої нової інформаційної бази (наприклад: Наша конфігурація) та вибираємо місце розташування інформаційної бази на даному комп'ютері або на комп'ютері в локальній мережі.

Вказуємо шлях до каталогу, в якому розташовуватиметься нова інформаційна база. При необхідності створюємо новий чистий каталог у вибраному місці на комп'ютері.

На наступному етапі можна вказати параметри запуску інформаційної бази (рис. 3.1).

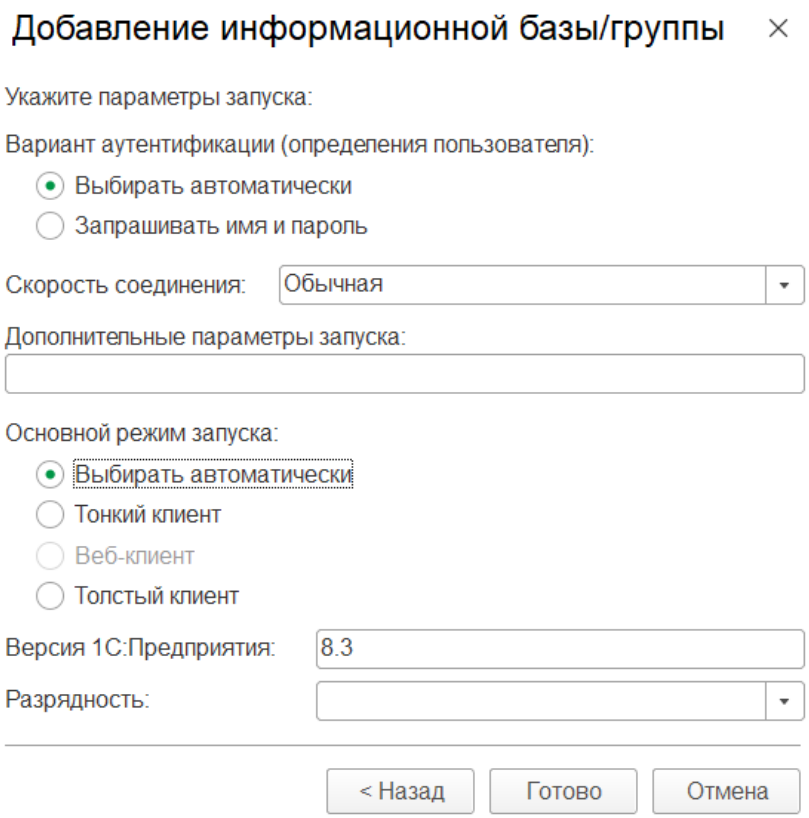

Рисунок 3.1 – Параметри запуску бази

Параметр Варіант автентифікації може приймати наступні значення.

Вибирати автоматично – у цьому випадку спочатку буде спроба виконати аутентифікацію засобами ОС, а у разі невдачі – запропоновано ввести логін/пароль для доступу до інформаційної бази.

Запитувати ім'я та пароль – у цьому випадку автентифікація завжди буде виконуватись у діалозі введення імені користувача та пароля.

Параметр Швидкість з'єднання дозволяє встановити швидкість з'єднання з інформаційною базою або сервером "1С:Підприємства". Параметр може приймати такі значення Звичайна – нормальна швидкість. Під час роботи системи нічого очікувати ніяких особливостей та Низька – низька швидкість з'єднання. У цьому режимі робота програми "1С:Підприємство" супроводжуватиметься деякими особливостями, описаними в розділі "Режим низької швидкості з'єднання". Цей режим рекомендується використовувати лише під час роботи через Інтернет через повільні канали зв'язку, наприклад, при підключенні через GPRS-модем.

Вибирати під час запуску – у цьому режимі вибирати швидкість з'єднання можна буде при кожному запуску інформаційної бази за допомогою прапора Низька швидкість з'єднання у нижній частині вікна запуску 1С:Підприємства. Якщо у властивостях інформаційної бази зазначено конкретне значення (Звичайна або Низька), прапор швидкості з'єднання буде недоступний для зміни.

Поле Додаткові параметри запуску дозволяє вказати різні параметри командного рядка, які будуть передані файлу, що виконується. Докладніше про параметри командного рядка можна прочитати у вбудованій довідці (розділ Запуск "1С:Підприємства 8" та параметри запуску) [16].

Параметр Основний режим запуску визначає, який клієнт використовуватиметься під час роботи з даною інформаційною базою:

- вибирати автоматично – у цьому режимі вигляд клієнтської програми визначатиметься автоматично;
- тонкий клієнт – для запуску використовуватиметься "Тонкий клієнт 1С";
- веб-клієнт – для запуску використовуватиметься веб-клієнт (наприклад, Chrome);
- товстий клієнт – для запуску використовуватиметься "Товстий клієнт 1С".

Поле Версія 1С:Підприємства: дозволяє вказати конкретний номер версії, який необхідно використовувати для доступу до бази даних.

Параметр Розрядність дозволяє вказати розрядність програми для роботи з базою:

- 32 (x86) – запуск 32-розрядної програми;
- пріоритет 32 (x86) – пошук найактуальнішої версії, і якщо для знайденої версії доступні програми в обох варіантах розрядності, вибирається версія з розрядністю 32;
- 64 (x86_84) – запуск 64-розрядної програми;

– пріоритет 64 (x86_64) – пошук найактуальнішої версії, і якщо для знайденої версії доступні програми в обох варіантах розрядності, вибирається версія з розрядністю 64.

Якщо не знаєте, який параметр на що впливає, то можете залишити всі значення за замовчуванням.

Натисніть кнопку Готово. База створена!

3.2 Використання Бібліотеки стандартних підсистем для початкового заповнення додатку функціоналом

Інструментарій розробника «1С:Бібліотека стандартних підсистем» (БСП) надає набір універсальних функціональних підсистем, готові розділи для документації користувача та технологію для розробки прикладних рішень на платформі «1С:Підприємство». Із застосуванням БСП стає можливою швидка розробка нових конфігурацій із вже готовою базовою функціональністю, а також включення готових функціональних блоків до існуючих конфігурацій. Використання БСП при розробці прикладних рішень на платформі «1С:Підприємство» дозволяє також досягти більшої стандартизації конфігурацій та зменшити час на вивчення та впровадження прикладних рішень за рахунок їх уніфікації з набору стандартних підсистем, що використовуються [17].

Підсистеми, що входять до БСП, охоплюють такі області, як адміністрування інформаційної бази, адміністрування користувачів програми, настроювання доступу до даних інформаційної бази, ведення різної нормативно-довідкової інформації (адресний класифікатор, курси валют, календарні графіки та ін.). БСП надає базові користувальницькі та програмні інтерфейси для роботи із завданнями та бізнес-процесами, що прикріплюються файлами та електронними підписами, контактною інформацією, додатковими реквізитами та відомостями, поштовими повідомленнями та ін.

Підсистеми можуть бути використані в конфігурації, що розробляється як всі разом, так і окремо. З точки зору технології застосування всі підсистеми БСП можна умовно розділити на дві категорії:

Підсистеми, що реалізують самостійну функціональність. Впроваджуються простим перенесенням функціональності («впровадив і забув») і вимагають істотних додаткових налаштувань.

Підсистеми, що інтегруються, надають функціональність, призначену для використання в тих чи інших об'єктах конфігурації-споживача (так звана «тісна інтеграція»). При впровадженні потрібно визначити склад об'єктів конфігурації-споживача, котрим необхідно виконати використання функціональності, після чого виконати кілька додаткових налаштувань, внести зміни у код і форми обраних об'єктів.

Разом із підсистемами БСП пропонує й окремі методики розробки прикладних рішень. У розпорядженні розробника є готові розділи для включення до складу документації користувача до прикладного рішення на базі БСП.

Для завдання первинного застосування та подальших оновлень версій БСП у прикладних рішеннях є спеціальний інструмент — помічник застосування. У його функції входить підтримка прийняття рішень щодо впровадження БСП у конфігурацію, і навіть автоматичний контроль фактичного результату застосування БСП. Оскільки прийняття рішень щодо впровадження БСП та контроль їх виконання не вимагають «занурення» в технічні деталі реалізації, то проектування, реалізація та контроль можуть виконуватись різними спеціалістами та в різні моменти часу.

Отож, для початку завантажимо БСП на комп'ютер, на якому буде відбуватися розробка. Для цього перейдемо на офіційний сайт https://releases.1c.eu/project/SSL31 та завантажимо актуальний реліз.

Після успішного встановлення визначимося зі складом підсистем, які будемо впроваджувати. Варто сказати, що є три підсистеми, які являються обов'язковими для

встановлення. Це базова функціональність, оновлення версії інформаційної бази та користувачі. Окрім них, впровадимо ще підсистем, які нам знадобляться:

- завершення роботи користувачів;
- аніліз журналу реєстрації;
- налаштування програми;
- звіт про рухи документів;
- пошук та видалення дублів;
- регламентні завдання;
- видалення помічених об'єктів;
- управління підсумками та агрегатами;
- контактна інформація;
- налаштування порядку елементів;
- друк;
- властивості;
- структура підпорядкованності;
- префіксація об'єктів;
- робота з файлами.

У каталозі зі встановленим БСП знаходимо обробку «Перше впровадження БСП». Відкриваємо її в чистій базі у користувацькому режимі (рис. 3.2).

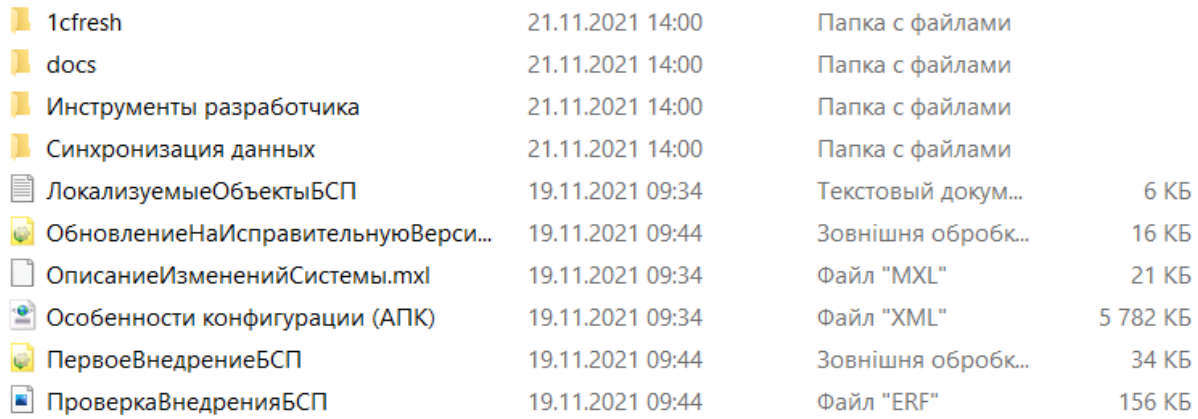

Рисунок 3.2 – Вміст каталогу БСП

Обираємо заздалегіть визначені підсистеми зі списку. Варто звернути увагу на те, що існує залежність між підсистемами. Якщо обрати підсистему, що залежна від іншої, то та інша підсистема автоматично буде обрана також до переносу (рис. 3.3).

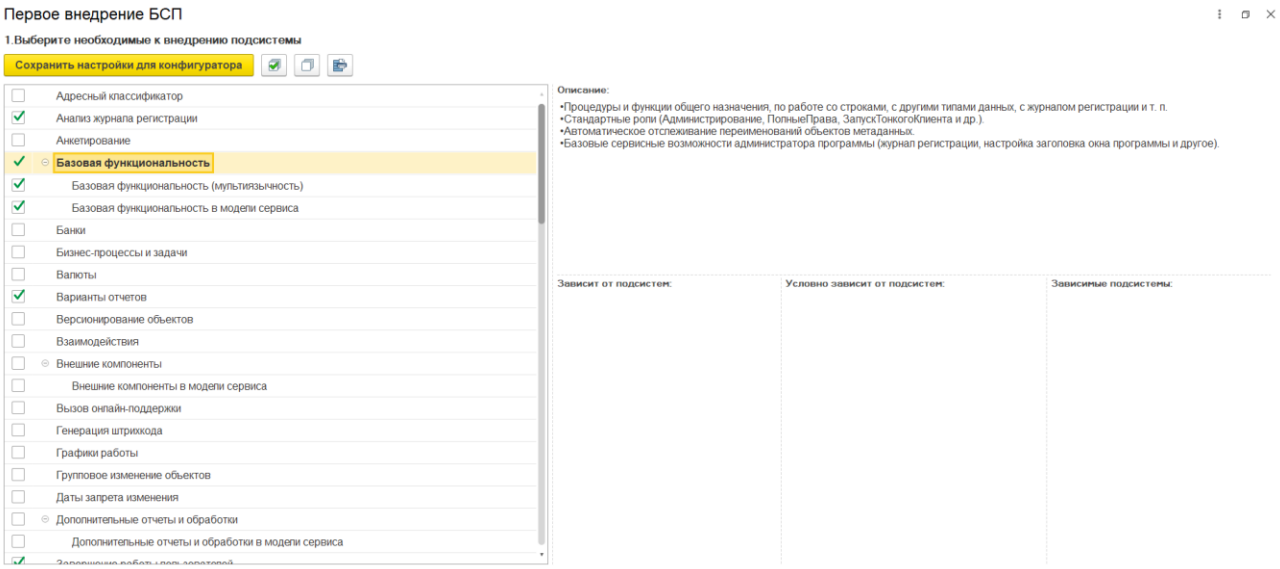

Рисунок 3.3 – Обробка першого впровадження БСП

Після того, як ми обрали всі необхідні нам підсистеми зберігаємо наш вибір. При цьому буде створено файл розширення .XML. Зміст його має бути приблизно наступним: об’єкт БСП, який може бути перенесений у нашу конфігурацію та властивість переносити чи ні.

```
<?xml version="1.0" encoding="UTF-8"?>
<Settings xmlns="http://v8.1c.ru/8.3/config/merge/settings" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" version="1.2" platformVersion="8.3.11">
    <MainConfiguration>
        <Name>Конфигурация</Name>
    </MainConfiguration>
    <SecondConfiguration>
        <Name>БиблиотекаСтандартныхПодсистем</Name>
        <Version>3.1.5.256</Version>
```

<Vendor>Фирма "1С"</Vendor>
</SecondConfiguration>
<Parameters>
<ConfigurationsRelation>ConfigurationsNotRelated</ConfigurationsRelation>
<AllowMainConfigurationObjectDeletion>false</AllowMainConfigurationObjectDeletion>
<CopyObjectsMode>false</CopyObjectsMode>
</Parameters>
<Objects>
<Configuration>
<Properties>
<Property name="Name">
<MergeRule>DoNotMerge</MergeRule>
</Property>
<Property name="Synonym">
<MergeRule>Merge</MergeRule>
</Property>
</Object>
</Objects>
</Settings>

Після цього нам потрібно перейти у конфігуратор та у підменю Конфігурація обрати пункт порівнення та об’єднання з конфігурацією із файла. Порівнювати нашу поки що пусту конфігурацію ми будемо із тією, що у нас є у каталозі БСП. Серед усіх представлених там нам необхідна 1Cv8 (рис. 3.4).

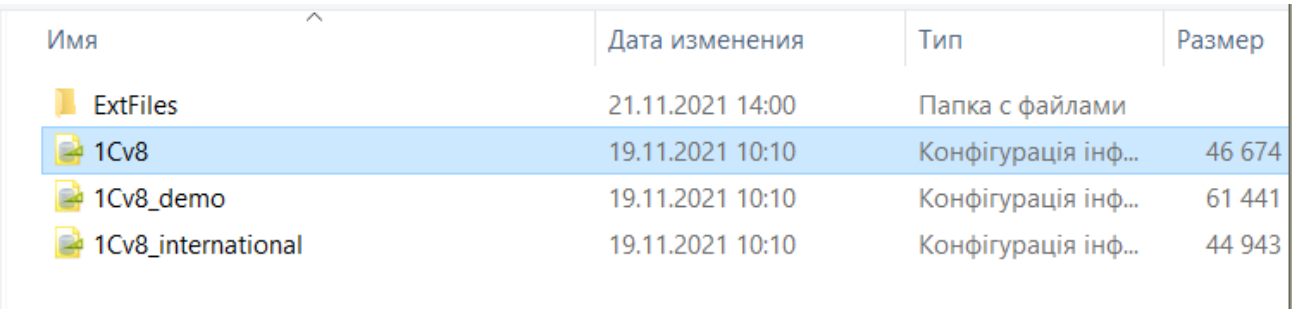

Рисунок 3.4 – Вміст каталогу БСП

Обравши необхідну конфігурацію та запустивши порівняння нам буде запропоновано виконати повне завантаження конфігурації із файлу так як поточна конфігурація пуста. Ми відмовляємося, так як нам потрібні лише певні об'єкти, які ми обрали раніше у обробці першого впровадження (рис. 3.5).

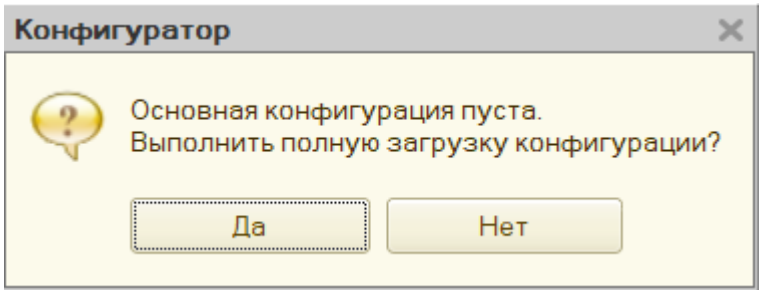

Рисунок 3.5 – Вибір повного завантаження конфігурації

Перед нами з'явиться вікно порівняння та об'єднання.

Механізм порівняння та об'єднання конфігурацій дозволяє порівнювати між собою два прикладні рішення та об'єднувати їх повністю або вибірково за результатами порівняння.

Цей механізм забезпечує як порівняння загальних властивостей об'єктів прикладного рішення (довідників, документів тощо. буд.), а й порівняння їх окремих реквізитів, табличних частин. Також виконується порівняння форм: порівнюються тексти модулів, тексти описів та макети.

Усі результати порівняння можна переглянути у детальному вигляді.

При запуску режиму порівняння система аналізує порівнювані конфігурації та встановлює відповідність між об'єктами конфігурацій, виходячи з їх імен.

Однак не виключена ситуація, коли однакові об'єкти прикладного рішення матимуть різні імена або навпаки, різні об'єкти будуть називатися однаково. У цьому випадку розробник має можливість відмовитись від відповідностей, встановлених за замовчуванням, та встановити їх вручну.

Результат порівняння конфігурацій відображається у спеціальному вікні. Розробник має можливість налаштувати склад інформації, що відображається у цьому вікні. Можливий перегляд всіх об'єктів прикладного рішення, що тільки відрізняються, тільки змінених, присутніх тільки в будь-якій конфігурації або лише незмінених об'єктів.

Для кожного об'єкта, що відрізняється, можна переглянути детальну інформацію про відмінності. Крім того, інформацію про відмінності можна отримати у вигляді звіту (рис. 3.6).

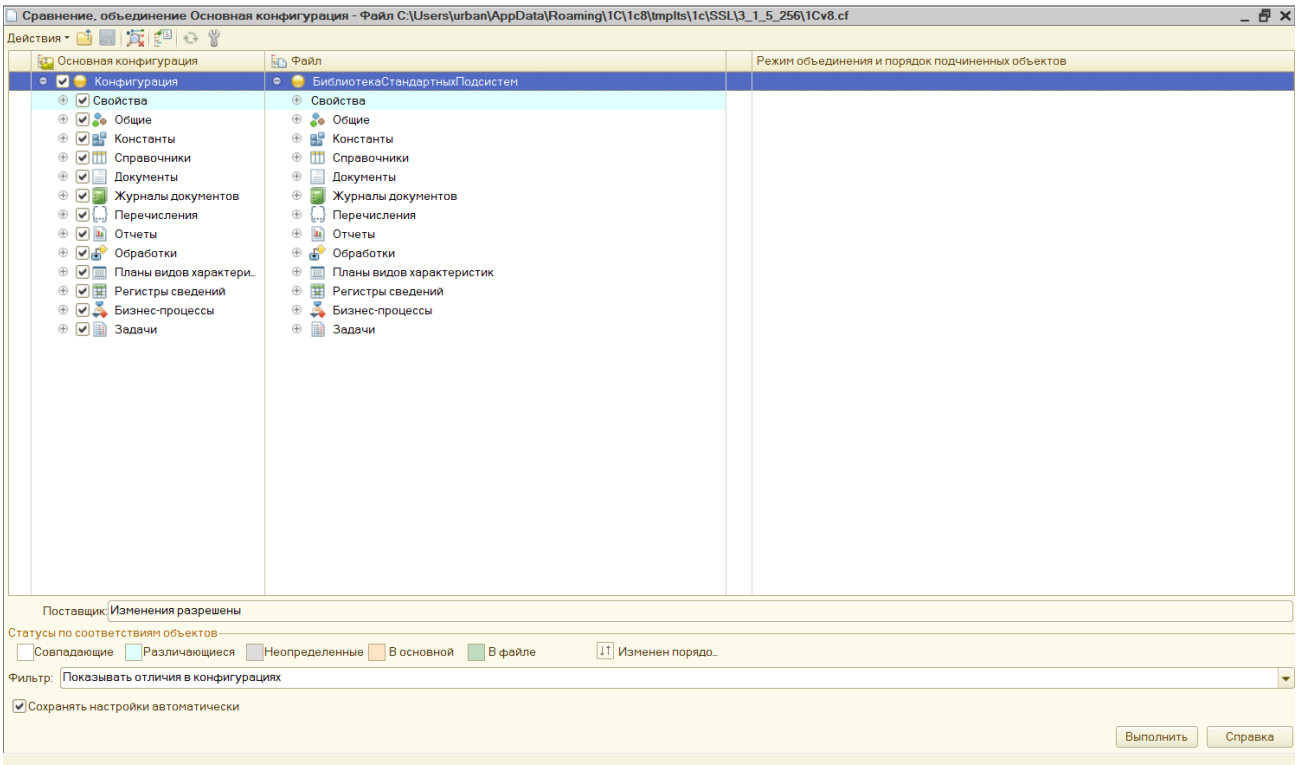

Рисунок 3.6 – Вікно порівняння та об’єднання конфігурацій

На виконання об'єднання змін слід зазначити ті об'єкти прикладного рішення, які брати участь у об'єднанні та встановити режим об'єднання конфігурацій.

Установка режиму об'єднання змін можлива як для всієї конфігурації в цілому, так і для кожного елемента прикладного рішення окремо.

Система підтримує порівняння та об'єднання різних видів конфігурацій. Як порівнювані змінні можуть виступати:

- основна конфігурація;
- конфігурація бази даних;
- конфігурація, збережена у зовнішньому файлі;
- конфігурація постачальника.

Таким чином, наприклад, можливе порівняння двох конфігурацій, збережених у зовнішніх файлах, або порівняння основної конфігурації із конфігурацією постачальника.

Налаштування об'єднання конфігурацій (або налаштування оновлення конфігурації на підтримці) можна зберігати у файл xml. Також доступна і зворотна операція – завантаження цих налаштувань із файлу.

Пакетний режим запуску конфігуратора також підтримує використання налаштувань під час об'єднання та оновлення конфігурацій. Таким чином, при об'єднанні конфігурацій, що містять велику кількість змін, коли об'єднання виконується регулярно, існує можливість повністю автоматизувати операції складання конфігурацій.

Саме тому ми завантажуємо раніше написані налаштування. При цьому автоматично розставляться галочки навпроти тих об'єктів, які будуть перенесені в конфігурацію. Виконуємо процедуру, після чого оновлюємо базу даних для прийняття змін.

Перед запуском програми у користувацькому режимі, потрібно виконати ряд процедур.

Перш за все, викликаємо правим кліком по кореню конфігурації її властивості, де вказуємо назву, версію та розробника.

Потім у дереві об'єктів знаходимо загальний модуль ОбновлениеИнформационнойБазыБСП та копіюємо його, змінюючі при цьому у назві БСП на скорочення нашої назви (рис. 3.7).

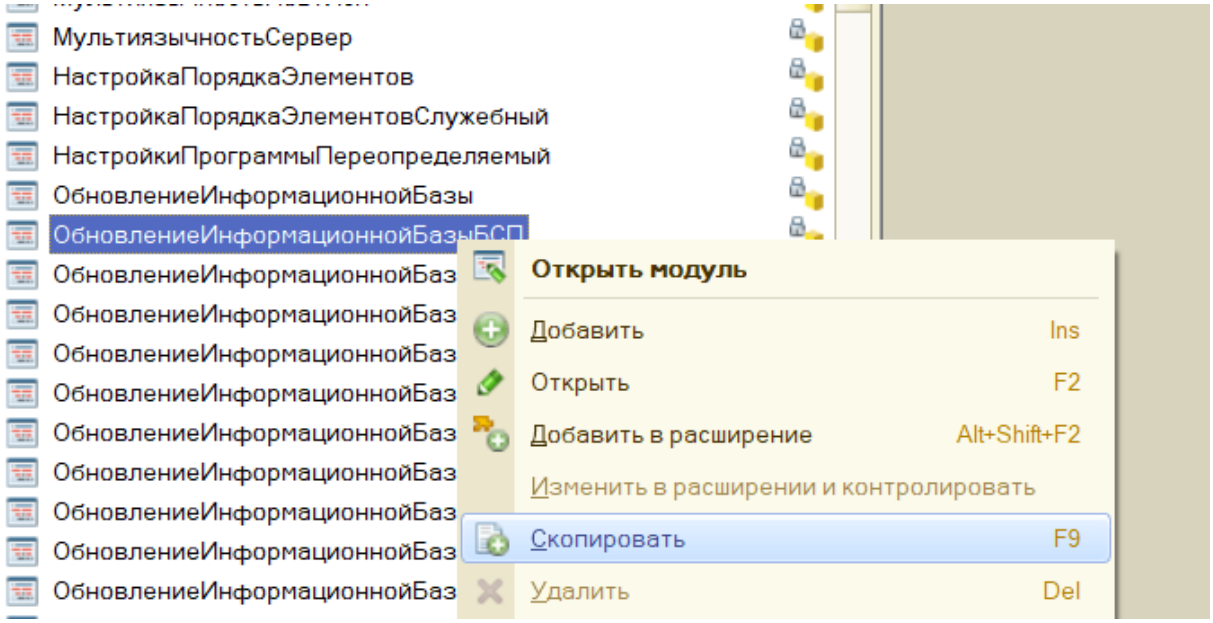

Рисунок 3.7 – Копіювання загального модулю

Всередині модуля змінюємо процедуру наступним кодом (рис. 3.8).

```
Процедура ПриДобавленииПодсистемы(Описание) Экспорт
    Описание.Имя = "МояКонфигурация";
    Описание.Версия = "1.0.1.1";

    // Требуется библиотека стандартных подсистем.
    Описание.ТребуемыеПодсистемы.Добавить("СтандартныеПодсистемы");
КонецПроцедуры
Процедура ПриДобавленииОбработчиковОбновления(Обработчики) Экспорт

КонецПроцедуры
Процедура ПередОбновлениемИнформационнойБазы() Экспорт

КонецПроцедуры
```

```
	Процедура ПослеОбновленияИнформационнойБазы(Знач ПредыдущаяВерсия, Знач ТекущаяВерсия,
		Знач ВыполненныеОбработчики, ВыводитьОписаниеОбновлений, МонопольныйРежим) Экспорт

	КонецПроцедуры
	Процедура ПриПодготовкеМакетаОписанияОбновлений(Знач Макет) Экспорт

	КонецПроцедуры
	Процедура ПриДобавленииОбработчиковПерехода СДругойПрограммы(Обработчики) Экспорт

	КонецПроцедуры
	Процедура ПриОпределенииРежимаОбновленияДанных(РежимОбновленияДанных, СтандартнаяОбработка) Экспорт

	КонецПроцедуры
	Процедура ПриЗавершенииПерехода СДругойПрограммы(Знач ПредыдущееИмяКонфигурации, Знач ПредыдущаяВерсияКонфигурации, Параметры) Экспорт

	КонецПроцедуры
```

Рисунок 3.8 – Код заміни процедури модуля

Потім потрібно додати наш створений модуль у той, що виконує перевірку. Проте просто вписати код ми не можемо так як цей модуль знаходиться на підтримці

та забороняє виконувати над ним будь-які зміни. Тому нам потрібно це виправити. Щоб це зробити треба включити можливість внесення змін до конфігурації.

Для цього виконайте команду Конфігурація – Підтримка – Налаштування підтримки. Відкривається форма "Налаштування підтримки".

У верхньому рядку діалогу відображається поточний режим. Там також знаходиться кнопка включення можливості змін для переведення у відповідний режим. Зворотної можливості переведення в режим повної підтримки не існує. Варто відзначити, що для редагування конфігурації постачальника достатньо включити можливість змін, знімати конфігурацію з підтримки не потрібно.

У цій формі натиснути кнопку Увімкнути можливість зміни. Відповісти “Так” на питання системи про неможливість виконання автоматичного оновлення (рис. 3.9).

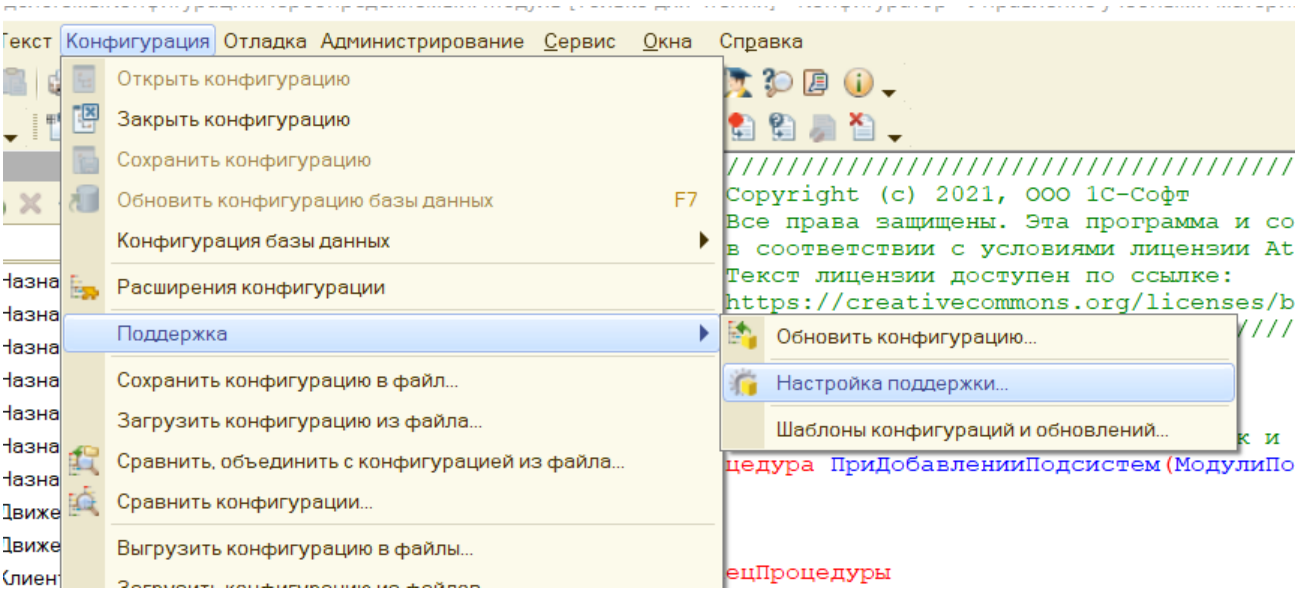

Рисунок 3.9 – Налаштування підтримки

При цьому створюється конфігурація постачальника.

У формі “Налаштування правил підтримки” встановити перемикачі в значення Об'єкт постачальника не редагується. Після цих дій можна точково відкривати доступ до об'єктів конфігурації, в які необхідно внести зміни. В нашому випадку загальний модуль ПодсистемыКонфигурацииПереопределяемый. І вже в цьому випадку у відкрившомуся вікні ми обираємо пункт Об’єкт постачальника редагується зі збереженням підтримки (рис. 3.10).

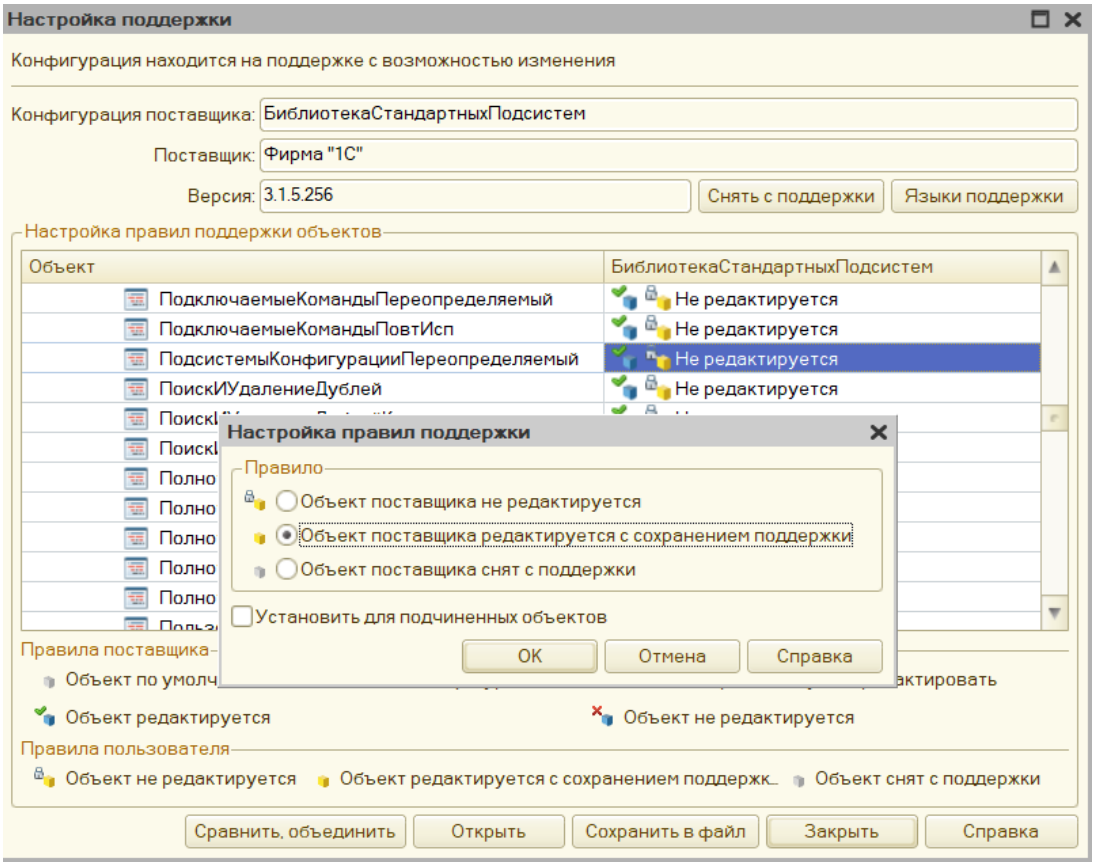

Рисунок 3.10 – Ввімкнення можливості редагування об'єкта

Після цього запускаємо дебаг, де розпочнеться початкове заповнення даних, яке займе декілька хвилин. У цей час не можна відключати пристрій через можливість пошкодити систему (рис. 3.11).

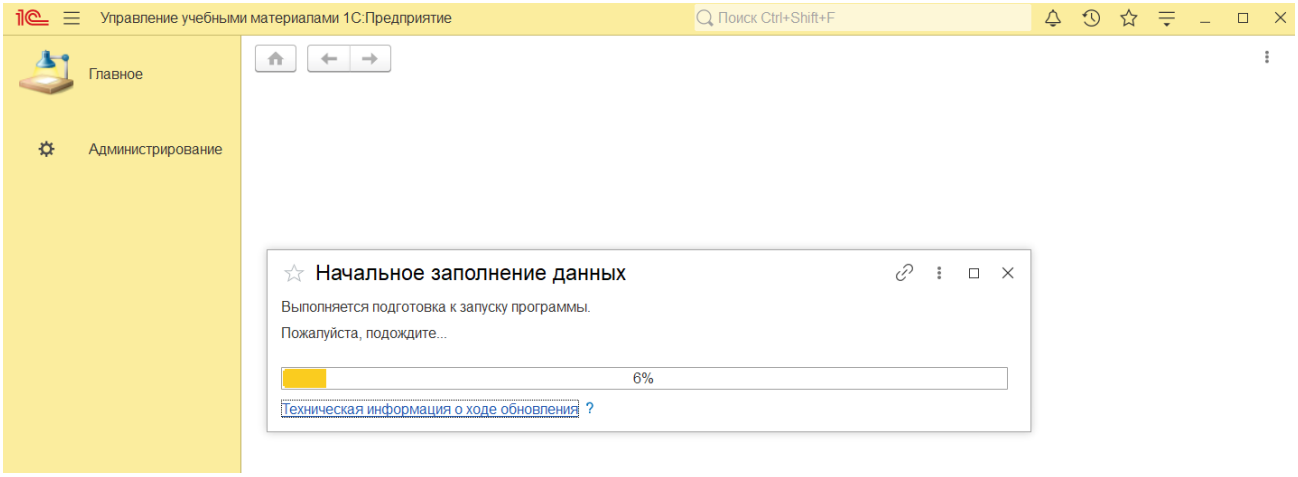

Рисунок 3.11 – Початкове заповнення даних

Коли процес початкового заповнення даних закінчиться, займемося створенням ролей.

3.3 Подальша розробка додатку

Ролі – це спільні конфігураційні об'єкти. Вони призначені для реалізації обмеження прав доступу до прикладних рішень. Роль у конфігурації може відповідати посадам або видам діяльності різних груп користувачів, для яких призначена дана конфігурація. Роль визначає, які дії, над якими об'єктами метаданих може виконувати користувач, який у цій ролі. У процесі ведення списку користувачів прикладного рішення кожному користувачеві ставиться у відповідність одна або кілька ролей.

При спробі користувача виконати дію, на яку він не має дозволу, дія виконана не буде, а система видасть вікно попередження (рис. 3.12):

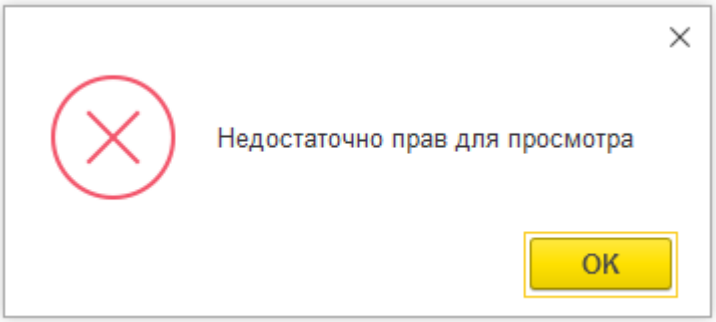

Рисунок 3.12 – Попередження про недостатні права

Для редагування складу ролей платформа містить два редактори.

Редактор ролі – це один із інструментів розробки. Він дозволяє змінювати склад прав однієї обраної ролі прикладного рішення.

Розробник може встановлювати дозволи як вручну, так і використовуючи механізм підсистем.

Редактор «Всі ролі» – це один із інструментів розробки. Він дозволяє змінювати та аналізувати склад прав відразу для декількох або для всіх ролей, що існують у прикладному рішенні.

Цим редактором зручно користуватися у випадках, коли потрібно одночасно встановити чи зняти одне право всім ролей. Або коли потрібно візуально порівняти та змінити відразу кілька ролей.

Якщо для будь-якого права потрібно встановити або зняти його дозвіл у всіх ролях, достатньо в першій колонці встановити або зняти прапорець дозволу.

Так як у нас буде лише три ролі: адміністратор, викладач та студент ми скористаємось редактором ролі (рис. 3.13).

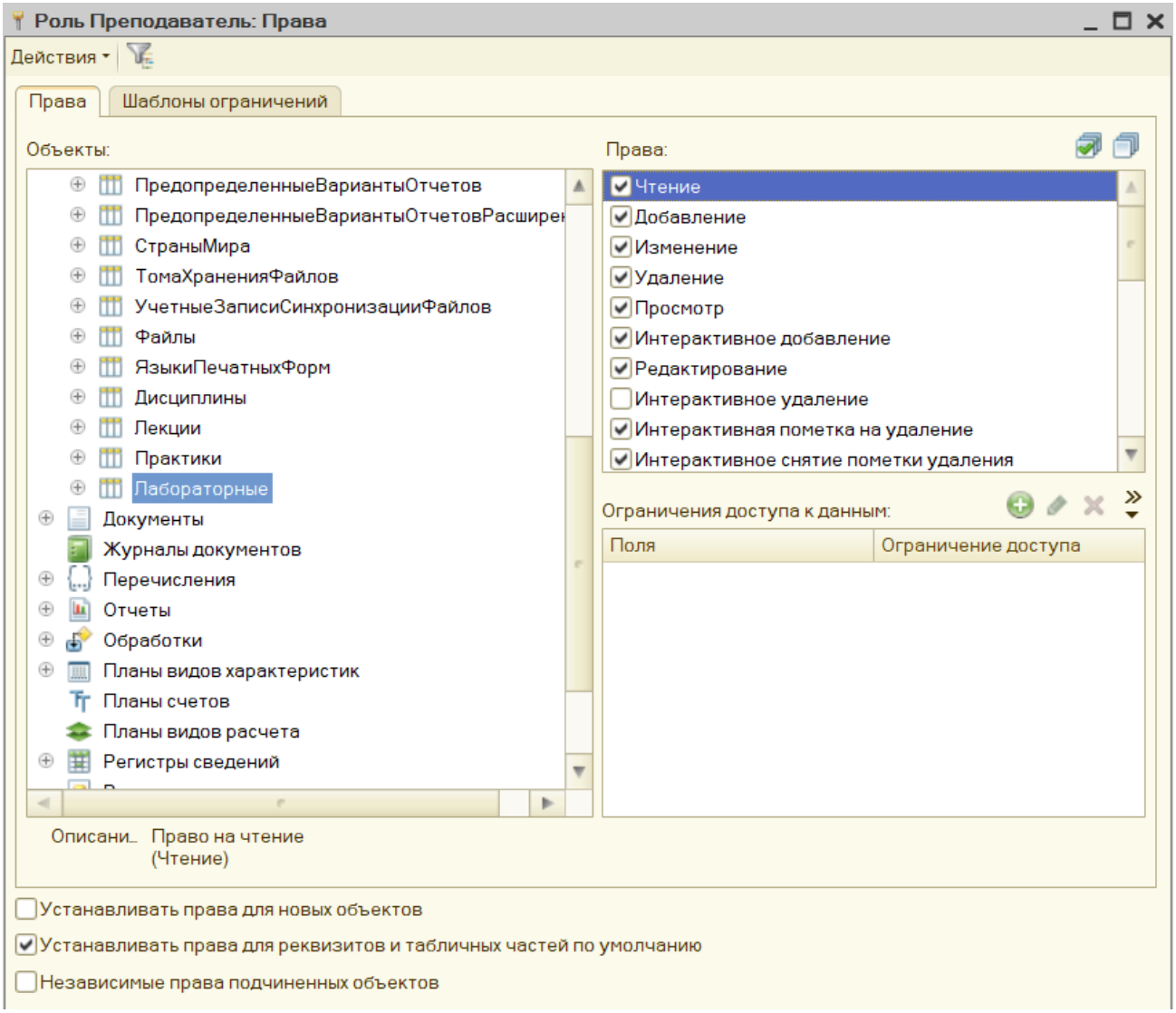

Рисунок 3.13 – Редактор ролі

Створимо користувачів. Зайдіть до Адміністрації – Установки користувачів та прав. Відкрийте підрозділ Користувачі та перейдіть за однойменним посиланням. Натисніть кнопку Копіювати або Створити.

У картці створення користувача занесіть повне ім'я та ім'я для входу. Якщо користувач є неактуальним, встановіть галочку Недійсний, і цей користувач сховається зі списку вибору.

У налаштуванні користувача можливі 3 варіанти аутентифікації:

– аутентифікація 1С: Підприємства;

– аутентифікація за протоколом OpenID;

– аутентифікація операційної системи.

У першому варіанті при вході в програму запитується ім'я користувача, під яким потрібно увійти.

Аутентифікація за протоколом OpenID використовується під час роботи з веб-клієнтом 1С: Підприємство. При цьому аутентифікація 1С: Підприємства також увімкнено.

При використанні ідентифікації Аутентифікація операційної системи вибирається користувач, під яким здійснюється вхід у Windows, і тоді вхід до 1С здійснюється автоматично під цим обліковим записом. Приберіть галочку Аутентифікація 1С: Підприємства. Аутентифікація операційної системи – найпростіший варіант входу.

Збережіть картку. Користувачі 1С створено (рис. 3.14).

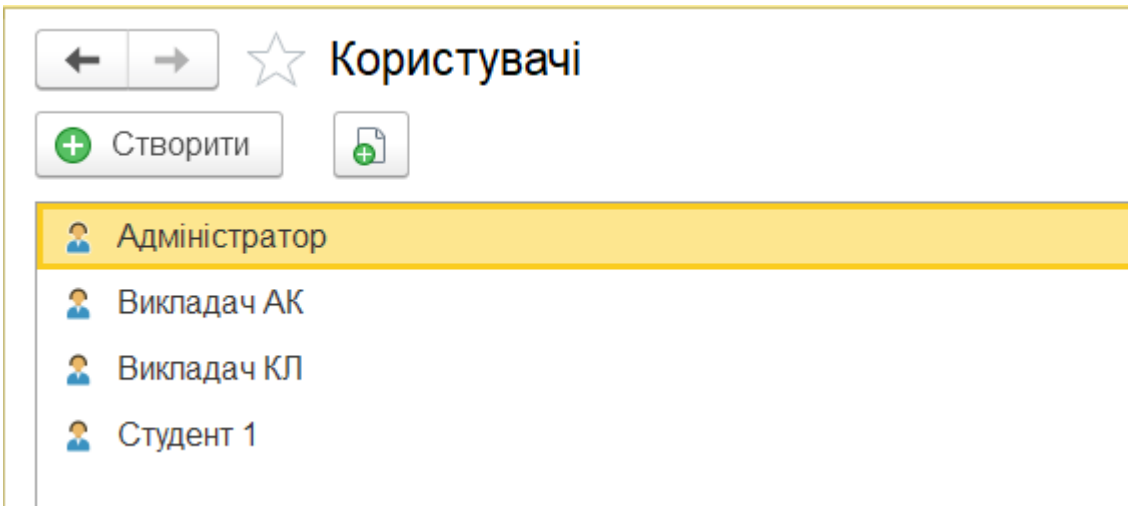

Рисунок 3.14 – Користувачі програми

Після створимо довідник Дисциплін. Одним з основних елементів будь-яких конфігурацій 1С є довідники. Вони зберігають у собі інформацію, що у більшості інших об'єктів додатка 1С. Саме тому при розробці чи модернізації будь-якої системи на платформі 1С насамперед додають нові довідники та заповнюють їх.

Довідники 1С - спеціалізований об'єкт дерева метаданих, який є зберігання статичної інформації довідкового характеру. Інформація в довідниках зазвичай не змінюється. Довідники використовуються практично у всіх об'єктах обліку як розріз обліку або довідкова інформація.

Поговоримо про властивості довідника у конструкторі розробки.

На вкладці «Основні» вказується ім'я, синонім, представлення об'єктів, опис призначення.

На вкладці «Ієрархія» можна встановити ієрархію об’єкта. Ієрархія в 1С 8.3 буває двох типів - "груп і елементів" та "елементів". Відрізняється тим, що у першому випадку батьком (папкою) може лише папка (група), тоді як у другому випадку батьком може бути елемент.

"Розміщувати групи зверху" - прапор відповідає за відображення груп у формі списку.

Також у налаштуваннях можна обмежити кількість груп ієрархії довідника відповідним налаштуванням.

Довідник може бути підпорядкований іншому довіднику. З погляду конфігурування 1С 8.3 це означає, що підлеглий елемент стає обов'язковим реквізит «Власник».

Вкладка "Дані". Найважливіша вкладка з погляду програміста. На ній зазначаються реквізити довідника.

Довідник має набір стандартних реквізитів, які не редагуються програмістом 1С 8.3, список їх можна побачити, натиснувши кнопку «Стандартні реквізити». Зупинюся на кожному детальніше.

Це Група - реквізит з типом булева, що показує, група це або елемент. Доступний лише у ієрархічному довіднику. Зверніть увагу, що значення цього реквізиту неможливо змінити в режимі 1С: Підприємство.

Код — реквізит, тип чи рядок (зазвичай рядок). Номер, який надається системою автоматично. Як правило, розраховується як (попередній код +1). Рекомендую використовувати саме рядковий тип, оскільки сортування числових значень відбувається не так, як потрібно. Можна використовувати як подання довідника у списку та в полях введення. Як правило, використовується для пошуку елемента під час введення рядка. Якщо Вам потрібно забрати поле Код, вкажіть у довжині рядка нуль.

Найменування – реквізит, обов'язковий до заповнення, рядкового типу. Максимальна довжина рядка – 150 символів. Можна використовувати як подання довідника у списку та в полях введення. Як правило, використовується для пошуку елемента під час введення рядка. Якщо Вам потрібно забрати поле Найменування, вкажіть у довжині рядка нуль.

Батько - реквізит, що має тип Довідник Посилання. Доступний лише у ієрархічному довіднику. Вказує на вищого батька в ієрархії. Якщо елемент або група знаходяться в корені довідника, вказується значення Довідник.<Ім'яПоточного Довідника>.

Власник – посилання на елемент-власник поточного елемента (групи) довідника. Доступний лише у підпорядкованому довіднику 1С.

Позначка Видалення - реквізит з типом булева. Відповідає за відображення позначки видалення в системі. Позначений видалення елемент вважається непридатним до використання, проте ньому можуть залишатися старі руху у документах.

Посилання – поле рядкового типу. У цьому реквізиті зберігається унікальний ідентифікатор об'єкта GUID. Те, що в системі ми бачимо у візуальному відображенні під назвою «посилання», — це лише уявлення об'єкта. Неможливо змінити.

Зумовлений тип булево, відображає, чи є елемент зумовленим, про це пізніше. Неможливо змінити.

На вкладці «Дані» також вказується подання довідника в системі, до версії 8.2.16 подання могло бути лише Кодом або Найменуванням. У нових версіях платформи (починаючи з 8.3) представлення можна описати самостійно в модулі менеджера за допомогою обробника «Обробка Отримання Подання».

Вкладка "Нумерація". Тут вказуються налаштування довідника щодо нумерації. Рекомендується використовувати саме автонумерацію. Контроль унікальності — прапор, який допомагає, якщо потрібно зробити код унікальним. Якщо зі встановленим прапором Ви спробуєте записати елемент довідника з неунікальним кодом, у 1С Ви отримаєте повідомлення «Код довідника став неунікальним».

Серія кодів визначає, як нумерувати довідник, можна ввести нумерацію довідника в розрізі власника. Наприклад, контрагент «Роги і копита» матиме свою нумерацію договорів — «1, 2, 3» тощо.

Вкладка "Форми". Тут описуються форми довідника. Якщо конфігурація запускається як у звичайному, так і керованому режимі, тоді вкладок із формами за замовчуванням буде дві: «основні» та «додаткові» — для звичайного та керованого програми різні.

На цій сторінці є важлива властивість довідника - "Введення по рядку". Це дуже зручна функція 1С 8, що дозволяє при заповненні даних у полі введення не заходити до довідника, а набрати його найменування, код або т.п. і вибрати з списку потрібний елемент.

Вкладка "Інше". На вкладці можна отримати швидкий доступ до основних модулів довідника – модуля об'єкта та модуля менеджера.

На сторінці також можна визначити список визначених елементів довідника. Це елементи, які неможливо видалити в режимі Підприємства. До певних елементів можна звернутися в конфігураторі безпосередньо, на ім'я, наприклад: Довідники.Номенклатура.Послуга.

На цій вкладці також визначається режим блокування – автоматичний або керований. Використання повнотекстового пошуку, а також довідкова інформація про довідник, доступна в режимі 1С: Підприємства.

Готовий результат у користувацькому режимі буде виглядати так (рис. 3.15).

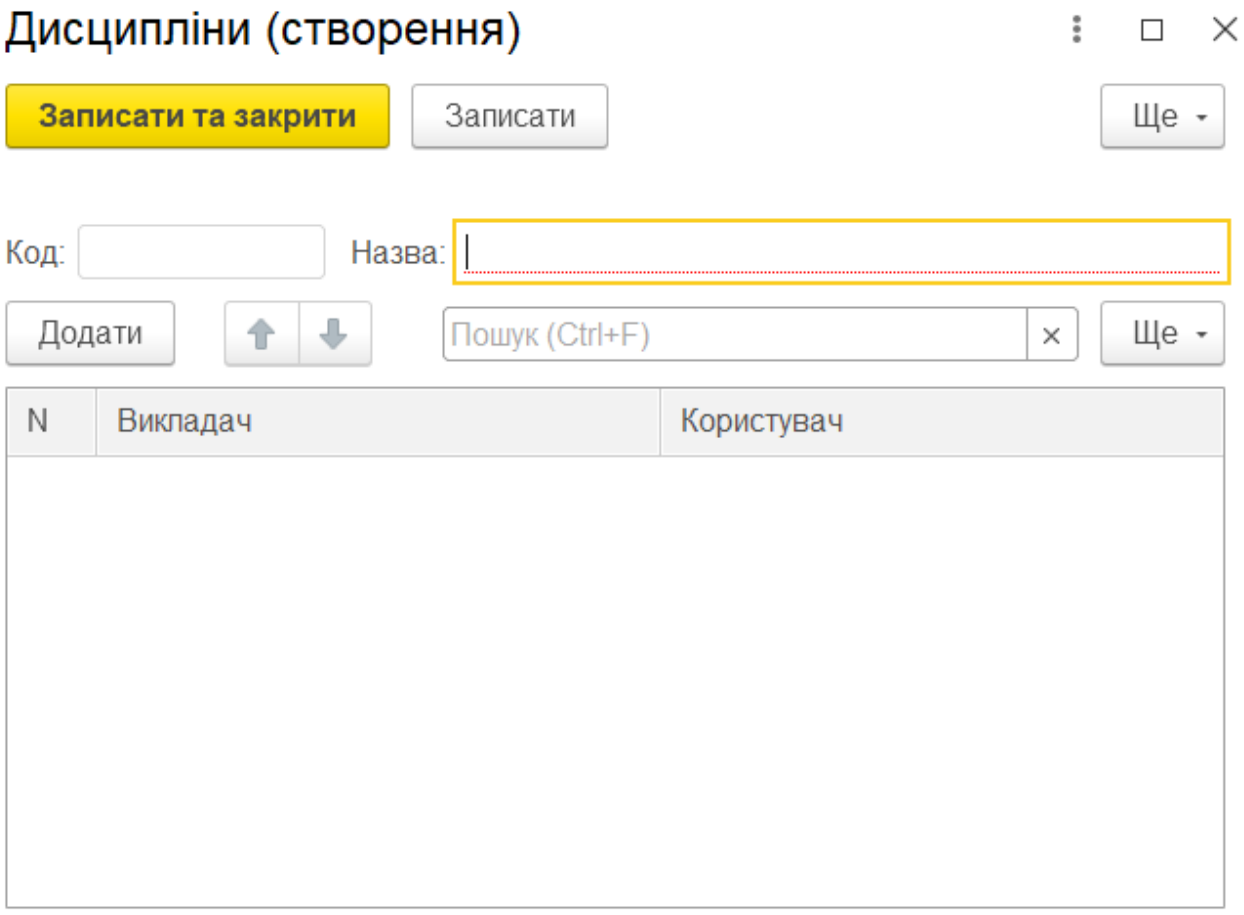

Рисунок 3.15 – Вигляд довідника дисциплін

Створимо також довідники для лекцій, практичних та лабораторних робіт з можливістю прикріплення файлів (рис. 3.16).

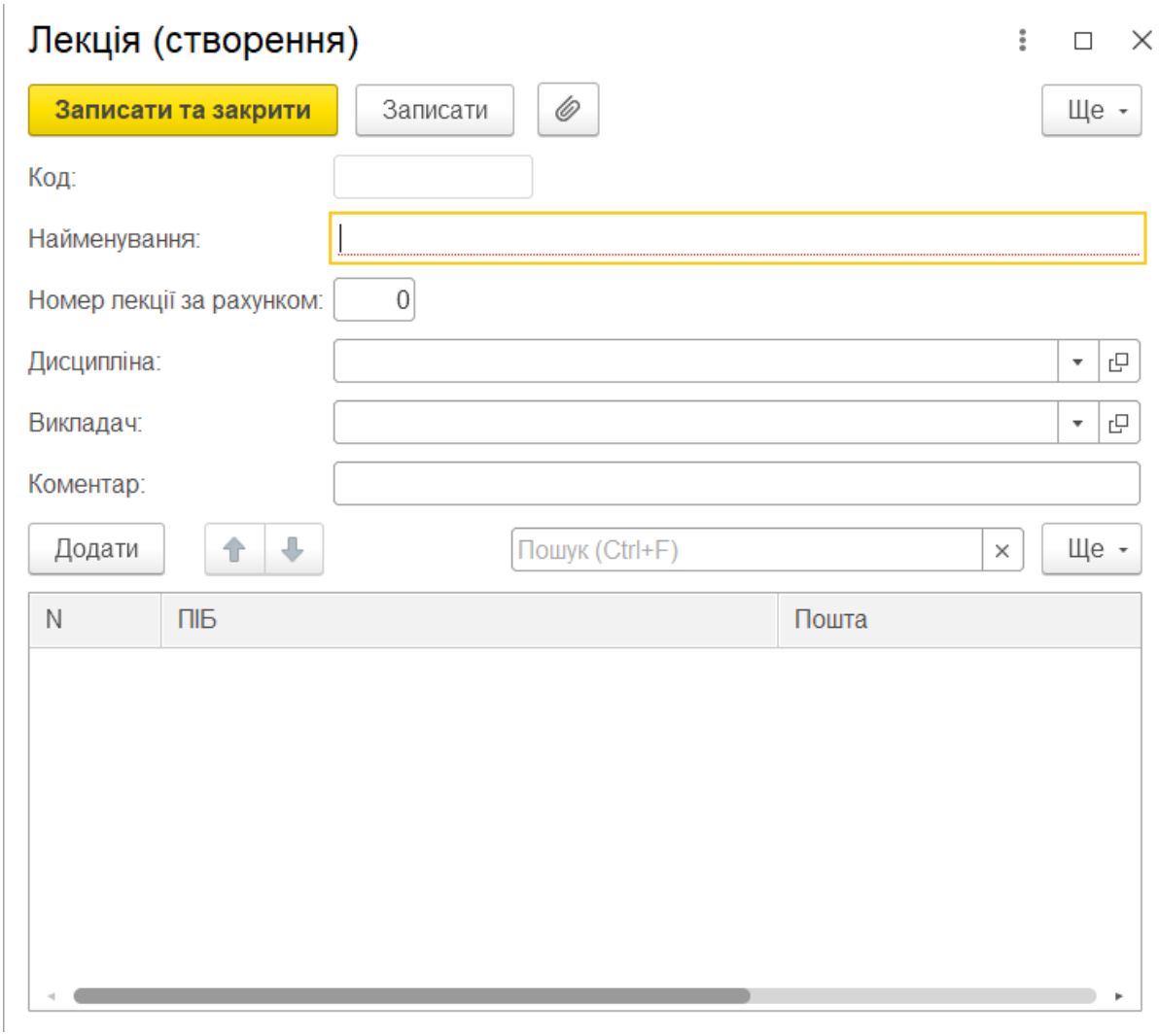

Рисунок 3.16 – Вигляд довідника лекцій

На цьому мета нашої магістерської роботи досягнута та можна приступити до підведення підсумків.

Висновки

Проаналізувавши на початку роботи специфіку сучасного навчального процесу, ми виявили тенденцію зросту переходу навчальних посібників у електроний формат. Для того, щоб забеспечити кожний аудиторний комп'ютер науковими матеріалами, ми мали на меті створити додаток, який вже у собі мав все необхідне, навіть без з'єднання з мережею Інтернет. Тож можна підбити наступні підсумки:

- вибір мови програмування являється найбільш важливим у процесі створення додатку. Адже обравши не підходящий варіант, можна знищити будь-яку можливість на успіх справи. Перш за все, варто звертати увагу на рівень взаємодії мови програмування з комп'ютером. Якщо основним критерієм при створенні програми являється максимальна взаємодія з апаратною частиною чи об'єм займаючої додатком пам'яті, тоді безсумнівно варто віддати перевагу мовам програмування низького рівня Але це робить неможливим переносимість додатку з комп'ютера на комп'ютер та сильно ускладнює процес написання та розуміння коду. В іншому випадку слід обрати мову програмування високого рівня, так як вона краща для сприйняття людиною, має простіший для запам'ятовування синтаксис та легша в транспортуванні. Визначившись із рівнем, варто обрати саму мову, що, в принципі, покладається на особисті вподобання програміста, тому що мови конкретного рівня, зазвичай, мають близький один до одного синтаксис та суттєво не відрізняються. В нашому випадку, ми обрали мову високого рівня 1С;
- в якості інтерфейсу ми використовували найбільш поширений на сьогодні інтерфейс «Таксі» через його простоту та хорошу комунікабельність та юзабільність із користувачем;

- використання Бібліотеки стандартних підсистем спрощує створення додатку через готовий основний функціонал, що міститься у ньому та може бути імплементований у програму;
- розділення користувачів на ролі дає змогу запобігти не бажаних втручань у роботу додатку та підвищує безпеку;
- систематизація даних продемонструвала, як зручно та швидко можна отримати доступ до необхідної інформації. Адже, один раз проаналізувавши дані, обравши конкретний критерій їх класифікації, та синтезувавши все згідно обумовлених катерогій, можна зекономити в майбутньому багато часу, замість того, щоб кожного разу шукати в купі потрібний файл.

Створена нами програма успішно працює на десктопних комп’ютерах та може бути встановлена на відповідні обчислювальні машини. І хоча, на даний момент, вона містить у собі навчальні матеріали лише однієї дисципліни, це не відміняє можливості подальших оновлень з додаванням інших предметів, не тільки напряму Комп’ютерної інженерії, а й з охопленням всіх дисциплін, що викладаються у вищому навчальному закладі.