

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Навчально–науковий інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

ПОЯСНЮВАЛЬНА ЗАПИСКА

до бакалаврської роботи

на ступінь вищої освіти бакалавр

на тему: «Розробка телеграм-боту «Тамагочі» мовою Python»

Виконав: студент 4 курсу,

групи ПД– 43 спеціальності

121 Інженерія програмного забезпечення

(шифр і назва спеціальності)

Лугина О.В.

(прізвище та ініціали)

Керівник Гаманюк І.М

(прізвище та ініціали)

Рецензент _____

(прізвище та ініціали)

Нормоконтроль _____

(прізвище та ініціали)

Київ – 2022

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ**

Кафедра	<u>Інженерії програмного забезпечення</u>			
Ступінь	вищої	освіти	-	<u>«Бакалавр»</u>
Спеціальність	-	121	<u>«Інженерія програмного</u>	<u>забезпечення»</u>

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

О.В. Негоденко

“ _____ ” _____ 2022 року

**З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ**

Лугина Олександр Володиморович

(прізвище, ім'я, по батькові)

1. Тема роботи: **«Розробка телеграм-боту «Тамагочі» мовою Python»**

Керівник роботи ст. вкл. Гаманюк І.М.

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

затверджені наказом вищого навчального закладу від “18” лютого 2022 року № .

2. Строк подання студентом роботи 03.06.2022.

3. Вихідні дані до роботи:

3.1. Положення аналізу створення телеграм-бота на рушії PyCharm;

3.2. Методи розробки створення телеграм-бота в pyCharm;

3.3. Існуючі інструменти та бібліотеки для створення телеграм-бота в pyCharm ;

3.4. Науково-технічна література;

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

4.1. Огляд телеграм ботів

4.2. Огляд та аналіз інструментів для створення телеграм-ботів

4.3. Створення та реалізація телеграм-бота

4.4. Висновки

5. Перелік графічного матеріалу.

5.1. Титульний слайд

5.2. Мета, об'єкт та предмет дослідження

5.3. Актуальність роботи

5.4. Аналоги

5.5. Порівняння з аналогами

5.6. Технічне завдання

5.7. Програмні засоби реалізації

5.8. Інструменти використані для реалізації

5.9. Архітектура телеграм-бота

5.10. Апробація результатів дослідження

5.11. Висновки

6. Дата видачі завдання: 11.04.2022

КАЛЕНДАРНИЙ ПЛАН

№з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.2022	Виконано
2	Дослідження існуючих інструментів для автоматизації тестування ПЗ	20.04.2022	Виконано
3	Проектування архітектури системи	21.04.2022 – 22.04.2022	Виконано
4	Розробка системи для автоматизації тестування ПЗ	23.04.2022 – 27.04.2022	Виконано
5	Висновки, оформлення роботи	27.04.2022 – 30.04.2022	Виконано
6	Розробка демонстраційних матеріалів	30.04.2022	Виконано
7	Попередній захист роботи	24.05.2022	
8	Здача роботи	03.06.2022	

Студент

(підпис)

Керівник роботи

(підпис)

_____ Лугина О.В.

(прізвище та ініціали)

_____ Гаманюк І.М.

(прізвище та ініціали)

РЕФЕРАТ

Об'єкт дослідження – надання розважальних послуг в галузі телеграм-ботів.

Предмет дослідження – розважальне програмне забезпечення у вигляді ігри в жанри тамагочі.

Мета роботи – покращити розважальне процес в галузі комп'ютерних игр .

Методи дослідження – являє собою виявлення аналогів, впровадження недоліків та порівняння с ними

Результати та їх новизна – результатом представляється мала фан-база всіх аналогів, в який на даний час являє собою мало поширеною в маси. Новизна являє собою якорем в котрій продуманість проектів являє собою недоліком.

Рекомендації щодо використання результатів роботи – Провадження в цю сферу розробки більш поширеної реклами для успішності проекта.

Сфера застосування – застосується в буденності.

Значимість роботи – представляє собою додавання бакалаврської роботі в новий жанр бази телеграм-ботів в вигляді ігри «тамагочі», які розважають користувачів в вільний час.

Висновки – пропозиції, впровадження більш великої ланки розробників, тестерів, дизайнерів і не тільки їх, в розробку проектів в цю сферу.

Зміст

ВСТУП.....	9
1. ОГЛЯД ТЕЛЕГРАМ БОТІВ А ТАК ЖЕ ТЕЛЕГРАМ ИГР ...	11
1.1. Аналіз та опис телеграм-бота.....	11
1.2. Аналіз та опис телеграм-бота «Жаба-бот»	12
1.3. Аналіз та опис телеграм-бота «Ферма:Mining»	15
1.4. Аналіз та опис телеграм-бота «Ligmar - Fantasy MMORPG»	18
2. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ ІСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ТЕЛЕГРАМ БОТА	20
2.1. Огляд Python та аналіз	20
3. СТВОРЕННЯ ТА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТА	30
3.1. Планування розробки проекту	30
3.2. Опис програми та її алгоритми.....	31
3.3 Реалізація.....	36
ВИСНОВКИ	40
ПЕРЕЛІК ПОСИЛАНЬ.....	41
Додаток А	43
Додаток Б	46

ВСТУП

Обґрунтування вибору теми та її актуальність: бота один із способів відпочити від важкого дня або роботи. Існують багато жанрів в тому числі як ферми и тд.

Для продовження розвитку ігрової індустрії та можливість відпочити, та трішки розважитись будтн створено просту гру у жанрі тамагочі на основі язика python на рушії pyCharm

Ступінь вивчення проблеми: На даний момент існує досить мало різноманітних ігор, які мають багато різноманітних механік. Але, небагато за таких продуктів розроблено саме на рушії pyCharm. До останніх належать лише одна розроблена гра на python Escape from «жаба-бот», «Ферма: mining» Але навіть остання телеграм-бот являється випущеною тільки для и не буде мати кросс-платформенность.

У зв'язку з тим, що инди-разработчики використовують частіше , та набагато рідше інші, проблема являється в тому, що вони вважають використання таких рушіїв, як telegram-bots не достатнього технічно розвинутими або певних механік. Також проблемою являється ще те , що розробники не мають можливість получать відсотки за продаж или на безплатної основі своєї гри створеним для телеграма тому-що фан-база . На даний час, компанії витрачають більше кошти на створення своїх рушіїв, ніж на використання інших.

Об'єктом дослідження надання розважальних послуг в галузі телеграм-ботів.

Предметом роботи розважальне програмне забезпечення у вигляді ігри в жанри тамагочі.

Метою роботи є створити декілька механіків для домашнього улюбленця на рушії pyCharm.

Завданням роботи є розробка телеграм-бота шутер за допомогою pyCharm

Методика дослідження: покращити розважальне процес в галузі комп'ютерних игр .

Враховуючи вимоги до специфіки програмного продукту, буде використана архітектурна технологія MVC.

Наукова новизна роботи: Таким чином, наукова новизна полягає в створенні простої гри в жанрі «Тамагочі»

Практична значущість результатів: Даний телеграмм-бот може бути використаний для всіх вікових категорій.

1. ОГЛЯД ТЕЛЕГРАМ БОТІВ А ТАК ЖЕ ТЕЛЕГРАМ ИГР

1.1. Аналіз та опис телеграм-бота

Сьогодні в It-індустрії одним з найпопулярнішим напрямком є Game development.

На даний момент при створенні диплома телеграм-бота "Тамагочі" на основі python я хочу побачити базову логіку і команди жанру Тамагочі. Самого вихованця якого:

- Треба годувати,щоб не помер з холоду;
- Давати йому спати, щоб відновив енергію щоб відправитися на роботу;
- Відправляти на роботу, щоб заробити грошей для покупок їжі, енергії і т.д.

Сьогодні в It-індустрії одним з найпопулярнішим напрямком є Game development.

На даний момент при створенні диплома телеграм-бота "Тамагочі" на основі python я хочу побачити базову логіку і команди жанру Тамагочі. Самого вихованця якого:

- Треба годувати,щоб не помер з холоду;
- Давати йому спати, щоб відновив енергію щоб відправитися на роботу;
- Відправляти на роботу, щоб заробити грошей для покупок їжі, енергії і т.д.

Telegram забезпечує зашифроване голосове та видеодзвінки та додаткові зашифровані «секретні» чати. Хмарні чати, групи зашифровується між програмою та сервером, тому провайдери та інші сторонні особи не можуть получить или взломать вас, але сам телеграм має ключ дешифрування. Користувачі можуть посилать повідомлення а так же голосові , анімовані наклейки, відеозвінки а так же голосі звінки. Ділитися необмеженою кількістю зображень, файлів до 2 гб, розташування користувачів, контактів та музики.

Боти спілкуються з користувачем інтернет-служб за допомогою instant messaging, Internet relay chat або іншими веб-інтерфейсів на приклад ютуб-боти, скайп-боти. Боти дозволяють людям задавати запитання англійською а потім

получають відповідь на своє звернення .

На самом ділі дуже багато було суперечок по поводу використання ботів у функції автоматизованої в телеграмне. Акційний сайт еБей розпочав суд, гамаючись придушити сторонні компанії як телеграм які використовували ботів для обходу їх веб-сайту в пошуках вигідних пропозицій а напямую за допомогою телеграма звернень до еБей. Заснована біржа ставок БетФаир побачила наскільки обсяг трофіки во зросло за допомогою ботів, який надходило від них, що вона запустила спеціально сервіс,націлений на програмістів-ботів.

Чат-боти – це програмне забезпечення розроблене на платформі месенджерів або соціальних мереж, яке має вигляд звичайного акаунту, але з деякими відмінностями та призначене для взаємодії з користувачем за допомогою текстових або аудіо повідомлень без участі людини-оператора.

1.2. Аналіз та опис телеграм-бота «Жаба-бот»

«Жабобот із жабами та жабаками»: як «кишеньковий» тамагочі з елементами ММОРПГ об'єднав у чатах тисячі людей

Понад сто тисяч користувачів щодня годують віртуальних вихованців та відправляють їх на війни.



Рисунок 1.1 – Видгляд ттоgrg «Жаба-бот»

«Моя жаба», «погодувати жабу», «надіслати жабу на роботу», «дуель прийняти» - такими надсиланнями заповнені чати, в яких грають із «Жабботом». Бот дозволяє не тільки доглядати а так же угаживать віртуальну жабу, але й вступати в битви з іншими гравцями або збирати паті для битви з босом.

«Жаббот» з'явився у січні 2020-го та за рік набрав понад 130 тисяч користувачів у телеграмі та у «ВКонтакте». Ті розібрався, чим тамагочі з жабами привабив стільки людей, яке співтовариство склалося навколо гри і хто займається розробкою бота.

Що таке «Жаббот» і як у нього грати?

Для початку гри бота потрібно додати в чат - кожні дві години він

надсилатиме картинку зі жабою або жабою. Сподобався земноводний учасник чату може забрати собі. Після цього за жабою доведеться доглядати: дати їй ім'я, годувати, підтримувати у хорошому настрої, у разі потреби реанімувати, відправляти на роботу та забирати з неї.

Як і в будь-якому тамагочі, головне не забувати годувати вихованця. Жаба їсть один раз на 12 годин, і кожне годування підвищує її рівень. Якщо не годувати вихованця 10 днів поспіль, він помре.

Внутрішньоігровою валютою служать козишки. За них можна купувати бусту, які збільшують настрій, реанімують жабу чи підвищують шанси на перемогу у міні-іграх. Ще валюта потрібна, щоб брати участь у ігрових подіях.

Заробляють комашок переважно на роботі. Жаба ходить на роботу разів у вісім годин. Працює вона лише дві години, а потім вирушає на шестигодинний відпочинок. Для вихованця можна вибрати три «професії». Найбезпечніше - сходити в їдальню, де можна отримати комашок, бонус до ситості або безкоштовну реанімацію.

Більш небезпечні «професії», круп'є та грабіжник, на відміну від відвідувача їдальні, приносять матеріали для крафтингу та більше качан. Пограбування дозволяють отримати максимальний прибуток, але водночас є ризик, що вихованця поб'ють чаплі — у такому разі жабу треба реанімувати. Вона також може передумати йти на роботу та звернути у СПА-салон. Або взагалі нічого не здобути.

Ще один ризиковий спосіб заробити комашок – участь у перегонах. Кожен учасник ставить певну суму, а переможець отримує 85% комашок від загальної суми вкладених коштів. Якщо жаба купує пиво, то шанси виграти перегони підвищуються.

Крім цього, комашок дають за участь у дуелі, яку можна викликати будь-якого учасника чату. Сутичка триває кілька раундів, а результат залежить від рівня та переважно від спорядження жаби. Переможець отримує комашок, а того, хто програв, доведеться реанімувати. Жаби можуть «відлетіти» – у такому разі станеться нічия.

Інгредієнти для крафтингу спорядження шукають у підземеллях: бронзових,

срібних чи золотих. Щоб їх знайти, потрібно вирушити у підземелля, заплатити за вхід, зібрати команду з жаб (максимум із п'яти) та перемогти боса. Можна піти і одному, але в такому разі шанси здолати боса серйозно знижуються. Команда, що програла, нічого не отримує, а переможці йдуть з лутом.

Після програшів у дуелях та побиття чаплями у жаби знижується настрій, який можна підвищити або льодяниками, або походом на «тусу». Вхід на вечірку платний: 150 комашок. Чим більше жаб приїде на тусу, тим більше підвищиться настрій кожної.

«Жаббот» також дозволяє одружуватися та заводити жабят. Шлюб стоїть комашок, і платить за нього той, хто пропонує. Розлучення теж небезкоштовне, зате кожні кілька днів подружжя отримує льодяники і бонус до ситості жаб. При цьому шлюб полягає не між жабами, а між користувачами - якщо партнер уклав шлюб в іншому чаті, то його можна перевірити на вірність і дізнатися про це.

Якщо одружуватися досить довго - 10 днів - то подружжя отримує можливість завести жабеня, якого поки можна тільки годувати. Годування дитини підвищує або знижує настрій жаби залежно від того, чи погоджується жабеня їсти або вередує. Якщо жабеня довго не годувати, він помре. Крім того, він не переносить розлучення батьків.

Усі маніпуляції з жабами відбуваються у межах одного чату. У цьому з іншими чатами можна взаємодіяти з допомогою кланів. Лідер може створити клан, запросити до нього до чотирьох жаб і вступати у війни з випадково обраним ворожим кланом. Битви між проходять з механіки дуелей, а переможці одержують качан. У кланів є свої рейтинги та система нагороджень за участь та перемоги у війнах.

1.3. Аналіз та опис телеграм-бота «Ферма:Mining»

Різні ігри, що дозволяють приємно провести час і заробити в наші дні далеко не рідкість. Колись давно навіть існували комп'ютерні ігри, які робили виплати в біткоінах на момент, коли він тільки з'явився. Проте варто чітко розділяти межу

між заробітком та дозволяям. Оскільки більшість проєктів не принесуть значного доходу.



Рисунок 1.2– Зовнішній вигляд «Ферма: Mining»

Експерти «Служби Безпеки» перевірили гру – робота «Ферма: Mining» (@farmgamebot) у Telegram. Під час перевірки факту шахрайства не було виявлено. Оскільки виплати з ігрового проєкту є реальними. Але справді заробити вдається лише одиницям, і то за умови, що доведеться інвестувати реальні гроші.

У гри - бота "Ферма: Mining" є свій телеграм - канал з новинами проєкту "Новини гри "Ферма: Mining"", на ньому налічується понад 23 тисячі передплатників. Також є спеціальний чат "Чат гри "Ферма: Mining"". Де

користувачі можуть спілкуватися між собою та обговорювати способи заробітку. У чаті бере участь понад 30 тисяч користувачів. Аудиторія на проекті жива, кількість переглядів у каналі новин хороша.

Реальні імена засновників проекту невідомі, з усіх питань необхідно звертатися до телеграм до адміністратора проекту – Daniil Pryshchepa (@cozeroool). Або на пошту проекту.

Запропонований даним проектом спосіб заробітку дуже невігядливий, користувачам бота доступні різноманітні гри в спеціальному боті. Концепт гри являє собою якусь віртуальну ферму, на якій користувач може вирощувати тварин, збирати ресурси, продавати їх та багато іншого.

Про засоби заробітку в ігровому боті «Ферма: Mining»

По суті всі способи заробітку в боті можна розділити на 3 категорії:

Збір та торгівля ресурсами;

Виконання завдань за типом передплати на різні канали;

Участь у реферальній програмі та залучення друзів.

"Ферма: Mining"

Способи заробітку внутрішньоігрової валюти.

Також у грі присутні 4 внутрішньоігрові валюти: FarmCoin, EUR, USD та RUB. FarmCoin є тією валютою, яку заробляє користувач, її можна отримати під час продажу ресурсів. EUR є універсальною валютою, її можна обміняти будь-яку валюту, висновок відбувається у ній. USD та RUB необхідні для покупки тварин.

Варто відзначити той факт, що розвиток ферми є дуже тривалим процесом, тому всім охочим прискорити його надається можливість внести реальні гроші у проект. Саме для цього і створено цей ігровий бот «Ферма: Mining».

Виведення зароблених коштів із проекту «Ферма: Mining»

Після того, як користувач інвестував свої кошти, і спробував їх відпрацювати – у нього виникає бажання перетворити ігрову валюту на реальні гроші. Докладніше про поповнення рахунку та виведення коштів можна дізнатися зі спеціального посібника з управління ботом Ферма: Mining» (посилання у розділі контакти). Однак нам хотілося б відзначити кілька підозрілих фактів: кожні 100

рублів, внесені в проект перетворюються на 5 EUR. Але при виведенні ті самі 100 рублів є еквівалентом 10 EUR. Власники проекту пояснюють таку різницю тим, що половина коштів йде рефералам. Завдяки цьому положенню гра дуже схожа на дію фінансових пірамід.

1.4. Аналіз та опис телеграм-бота «Ligmar - Fantasy MMORPG»

«Ідея проекту зародилася ще в далекому 2014 році, коли кілька студентів з гарячими головами скучали на лекціях. Скільки було спроб допілити його, з якими труднощами ми стали ще в студенські роки краще залишити для окремої статті.

Так, у 2019 році, заручившись підтримкою одного «московського інвестора», ми з командою переїхали в Мінськ і занялися розробкою. Перевіряли гіпотези, тестували формати і к осені визначилися з четким планом розробки та розвитку проекту. На початку літа ми розійшлися з нашим «інвестором», залишилися без фінансової підтримки, але вирішили, що ми справимся за рахунок своїх засобів і осінь уж точно вийдем на окупність.



Рисунок 1.3– Фан-арт Бота «Ligmar»

Реальність виявилася більш жорсткою. На окупаемость проекту не вийшов, ми виконали досить багато помилок, не учли дуже багато моментів. У який-то момент пришлось переключитися на інший проект. Ми надеялись, що ми можемо розділити Ligmar хотіти трохи часу, але в підсумку зараз ми не можемо підтримувати проект вовсе».

Таблиця 1.1 – Зведені результати аналогів

Назва бота	Ligmar - Fantasy MMORPG	Ферма: Mining	Жаба-бот
Платформа	Телеграм	Телеграм	Телеграм
Популярність	Ні	Так	Так
Простота старту	Так	Ні	Так
Успішність бота, фінансово	Так	Так	Так
Складність проекту	Так	Так	Ні
Взаємодія с ігрокami	Так	Ні	Так

2. ОГЛЯД ТА АНАЛІЗ ІСНУЮЧИХ ІСТРУМЕНТІВ ДЛЯ СТВОРЕННЯ ТЕЛЕГРАМ БОТА

2.1. Огляд Python та аналіз

«Бот» (скороченно від «робот») – програма яка робить автоматично готову задачу, яка повторюється.. Боти зазвичай імітують поведінку користувача або замінюють його. Користувачі виконують задачі набагато повільніше ніж боти які автоматизовані. Вони виконують корисні функції, наприклад обслуговування клієнтів або індексація пошукових систем. Алеж боти також можуть бути шкідливими програмами, які використовуються для управління і повного контролю над комп'ютером и зашкоджувати роботі користувача.

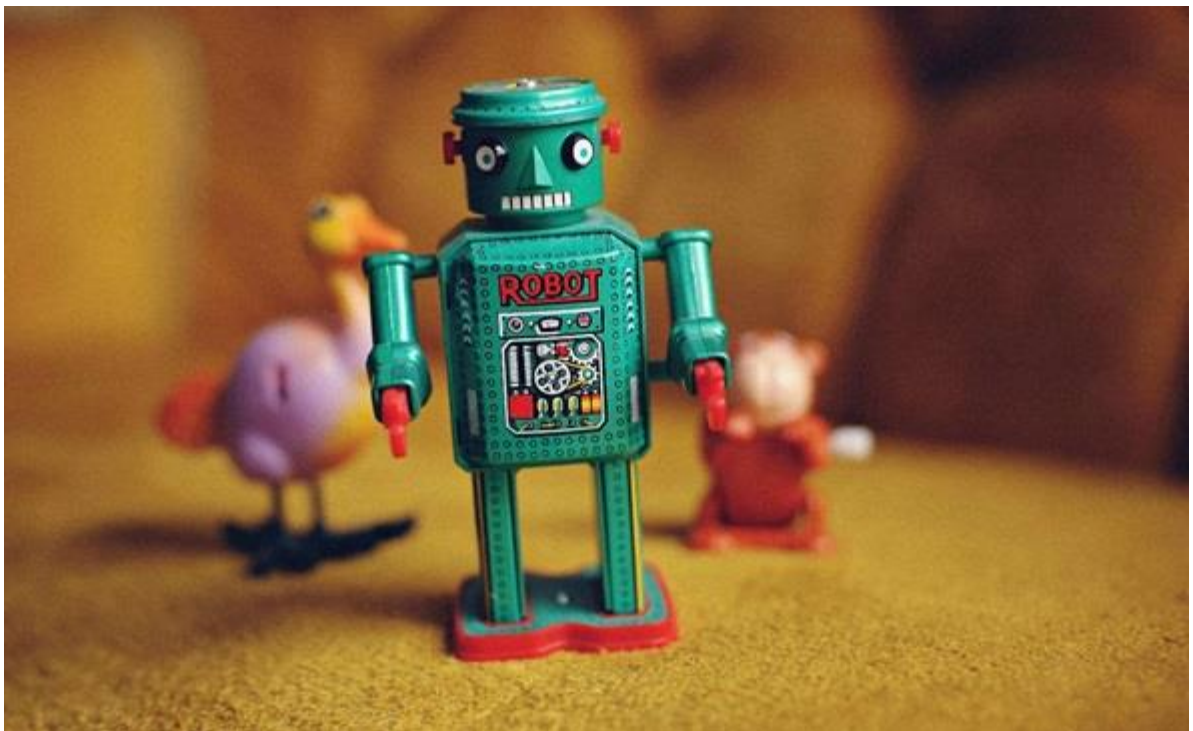


Рисунок 2.1 Уявне представлення бота

Що таке комп'ютерні та інтернет-боти?

Комп'ютерні інтернет боти є цифровими інструментами і, як будь-який

інструмент, можуть використовуватися як на благо, так і на шкоду.

Інтернет-боти також можуть називатися павуками, пошуковими роботами або веб-ботами.

"Хороші" боти виконують корисні завдання, а "погані" або шкідливі боти можуть використовуватися для злону, розсилки спаму, шпигунства, переривання та злону веб-сайтів будь-якого розміру. За оцінками, в даний час до половини всього інтернет-трафіку припадає на комп'ютерних ботів, що виконують певні завдання, такі як автоматизація обслуговування клієнтів, імітація людського спілкування в соціальних мережах, допомога компаніям у пошуку контенту в Інтернеті та пошукової оптимізації.

Компанії та приватні особи використовують ботів для заміни повторюваних завдань, які інакше довелося б виконувати людям. Завдання, що виконуються ботами, зазвичай прості і виконуються набагато швидше, ніж якби їх робила людина. Однак не всі завдання, що виконуються ботами, безпечні. Іноді ботів використовують для злочинних дій, таких як крадіжка даних, шахрайство та DDoS-атаки.

Відмінність телеграм-бота від бота

Telegram Бот — спеціальні акаунти в Telegram, створені для того, щоб автоматично обробляти та надсилати повідомлення. Користувачі можуть взаємодіяти з ботами за допомогою повідомлень, що надсилаються через звичайні або групові чати. Логіка робота контролюється за допомогою HTTPS запитів до нашого API для роботів.

Що таке тамагочі

Начало тамагочі – це дитяча іграшка в виде яйця з екраном чорно-білого відтінка та трьома кнопками как на рис.(2.2). Начало тамагочі почався 23 листопада 1996 року. Його розробкою займалася студія Bandai.

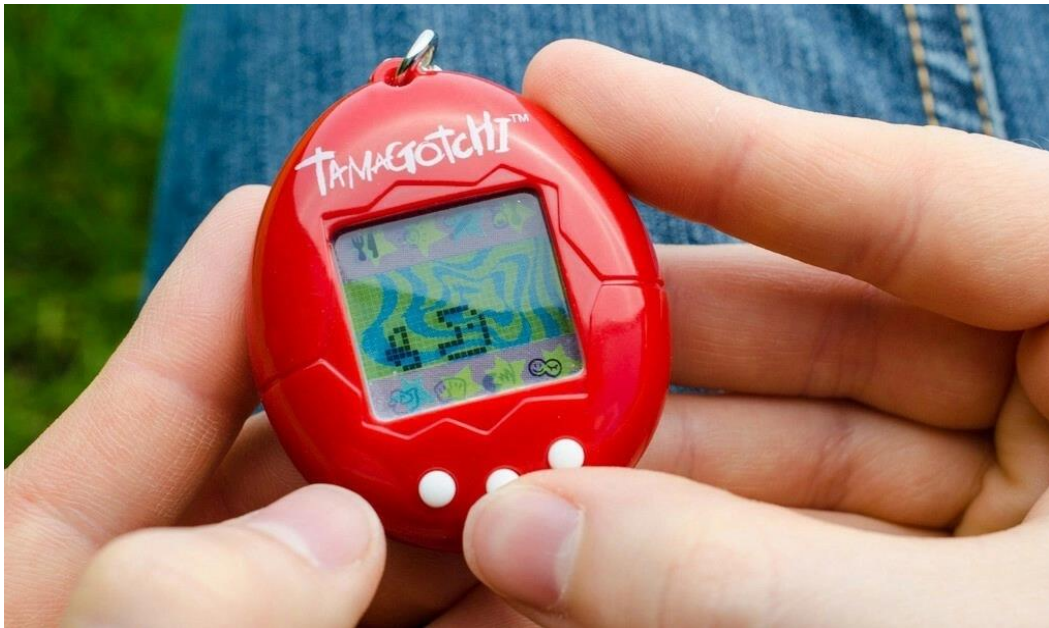


Рисунок 2.2 – Вигляд тамагочі

"Tamagotchi" получила свою назву при поєднанні двох слів - японського слова "tamago" - "яйце" і англійського слова "watch" - "дивитися". За легендою авторів, тамагочі – це інопланетні створіння, які прилетіли на Землю, щоб вивчити умови життя на ній, але змушені залишатися у захисній оболонці – яйце. Щоб вижити, звірята з іншої планети потребують постійної турботи людини.

Python - високорівнева мова програмування загального призначення з динамічною строгою типізацією та автоматичним управлінням пам'яттю орієнтована підвищення продуктивності розробника, читаності коду та її якості, і навіть забезпечення переносимості написаних у ньому програм. Мова є повністю об'єктно-орієнтованим у цьому плані, що це об'єктами. Незвичайною особливістю мови є виділення блоків коду пробільними відступами . Синтаксис ядра мови мінімалістичний, рахунок чого практично рідко виникає необхідність звертатися до документації. Сама ж мова відома як інтерпретована і використовується в тому числі для написання скриптів . Недоліками мови є найчастіше нижча швидкість роботи і більш високе споживання пам'яті написаних ним програм порівняно з аналогічним кодом, написаним компілюваних мовами, як-от C чи C++.



Рисунок 2.3– Логотип python, пример кода на заднім фоні

Python є мультипарадигмальною мовою програмування, що підтримує імперативне, процедурне, структурне, об'єктно-орієнтоване програмування, метапрограмування та функціональне програмування. Завдання узагальненого програмування вирішуються рахунок динамічної типизації.

за рахунок використання JIT-компіляції. Підтримка версії Python 2 закінчилася в 2020 . Зараз активно розвивається версія мови Python 3. Розробка мови ведеться через пропозиції щодо розширення мови PEP (англ. Python Enhancement Proposal), в яких описуються нововведення, робляться коригування відповідно до зворотного зв'язку від спільноти та документуються підсумкові рішення .

Стандартна бібліотека включає великий набір корисних функцій, що переносяться, починаючи з можливостей для роботи з текстом і закінчуючи засобами для написання мережевих додатків. Додаткові можливості, такі як математичне моделювання, робота з обладнанням, написання веб-додатків або розробка ігор можуть реалізовуватися за допомогою великої кількості сторонніх бібліотек, а також інтеграцією бібліотек, написаних на Сі або С++, при цьому і сам інтерпретатор Python може інтегруватися в проекти, написані цими мовами. Існує і спеціалізований репозиторій програмного забезпечення, написаного на Python -

PyPI . Цей репозиторій надає засоби для простої установки пакетів в операційну систему та став стандартом де-факто для Python. Станом на 2019 рік у ньому містилося понад 175 тисяч пакетів.

Python став однією з найпопулярніших мов, він використовується в аналізі даних, машинному навчанні, DevOps та веб-розробці, а також в інших сферах, включаючи розробку ігор. За рахунок читабельності, простого синтаксису та відсутності необхідності в компіляції мова добре підходить для навчання програмування, дозволяючи концентруватися на вивченні алгоритмів, концептів та парадигм. Налагодження ж і експериментування значною мірою полегшуються тим фактом, що мова інтерпретується . Використовується мова багатьма великими компаніями, такими як Google або Facebook . Станом на жовтень 2021 року Python посідає перше місце у рейтингу TIOBE популярності мов програмування з показником 11,27%. «Мовою року» за версією TIOBE Python оголошувався у 2007, 2010, 2018 та 2020 роках. PyCharm — інтегрована середовище розробки для мови програмування Python. Предоставляет средства для анализа кода, графический отладчик, инструмент для запуска юнит-тестов и поддерживает веб-разработку на Django. PyCharm разработана компанией JetBrains на основе IntelliJ IDEA.

PyCharm — Середовище розробки Python, що використовує в комп'ютерному програмуванні. Розробкою займається компанію JetBrains. Забезпечує базу а точніше: аналіз коду, графічний налагоджувач, модуль тестування, інтеграція с системою контролю

The screenshot shows the Python Console in PyCharm. The code being executed includes imports for fbprophet, matplotlib, numpy, pandas, and requests. It also shows the use of mpl.style.use('ggplot') and a print statement for the current UTC time. A DataFrame named 'hits_df' is loaded from a CSV file, and its tail is displayed. A tooltip for the 'reindex()' method is visible over the code.

```

Python Console x
... import fbprophet
... import matplotlib as mpl
... import matplotlib.pyplot as plt
... import numpy as np
... import pandas as pd
... import requests

Importing plotly failed. Interactive plots will not work.
In[3]: mpl.style.use('ggplot')
... figsize = (14,7)
In[4]: now = datetime.datetime.utcnow()
... print(f'This notebook was last rendered at {now} UTC')
...
This notebook was last rendered at 2020-06-29 14:45:41.977553 UTC
In[5]: hits_df = pd.read_csv('ipy_nb_counts.csv', index_col=0, header=0, parse_
... hits_df.reset_index(inplace=True)
... hits_df.drop_duplicates(subset='date', inplace=True)
... hits_df.set_index('date', inplace=True)
... hits_df.sort_index(ascending=True, inplace=True)
In[6]: hits_df.tail(3)
Out[6]:
           hits
date
2020-05-27  7697597
2020-05-28  7712332
2020-05-29  7729962
In[7]: til_today = pd.
In[8]: hits_df = hits_df.rein

figsize = (tuple: 2) (14, 7)
hits_df = (DataFrame ...View as DataFrame)
now = (datetime) 2020-06-29 14:45:41.97
til_today = (DatetimeIndex: 2062) [... View]
Special Variables
reindex()
reindex_like(self, other, method, copy, limit, tolerance)
reset_index(self, level, drop, inplace, col_level, col_f...
_reindex_axes(self, axes, level, limit, tolerance, metho...
_reindex_index(self, new_index, method, copy, level, fil...
_reindex_columns(self, new_columns, method, copy, level,...
_reindex_multi(self, axes, copy, fill_value)
_reindex_with_indexers(self, reindexers, fill_value, cop...
_needs_reindex_multi(self, axes, method, level)
Press ↵ to insert, ⇧ to replace

```

Рисунок 2.4–Зовнішній вигляд PyCharm

PyCharm є кросплатформним, з версіями для Windows, macOS та Linux. Видання для спільноти випущено під ліцензією Apache , а також є навчальна версія, а також професійна версія з додатковими функціями (випущена за власною ліцензією, що фінансується підпискою)

- Допомога та аналіз коду із завершенням коду, виділенням синтаксису та помилок, інтеграцією лінера та швидкими виправленнями
- Навігація по проекту та коду: спеціалізовані перегляди проекту, перегляди структури файлів і швидкий перехід між файлами, класами, методами та використанням
- Рефакторинг Python: включає перейменування, метод вилучення, введення змінної, константу, підтягування, натискання та інші
- Підтримка веб-фреймворків: Django, web2py і Flask [лише професійна версія]
- Інтегрований налагоджувач Python
- Інтегроване модульне тестування з порядковим покриттям коду
- Розробка Google App Engine Python [лише професійна версія]
- Інтеграція контролю версій: уніфікований інтерфейс користувача для

Mercurial, Git, Subversion, Perforce і CVS зі списками змін і злиттям

- i) Підтримка наукових інструментів, таких як Matplotlib, NumPy і SciPy [лише професійне видання]
- j) Він конкурує в основному з низкою інших IDE, орієнтованих на Python, включаючи PyDev від Eclipse і більш широку IDE Komodo.

База даних

База даних SQL Server складається з колекції таблиць, у якій зберігається певний набір структурованих даних. Таблиця містить колекцію рядків, також званих записами або кортежами, і стовпців, також званих атрибутами. Кожен стовпець у таблиці призначений зберігання конкретного типу даних, наприклад дат, імен, грошових сум чи чисел.

Основні відомості про бази даних

На комп'ютері може бути один або кілька екземплярів SQL Server. Кожен екземпляр SQL Server може містити одну або кілька баз даних. У базі даних може бути одна або кілька груп об'єктів володіння, які називаються схемами. У кожній схемі присутні об'єкти бази даних, такі як таблиці, уявлення та процедури, що зберігаються. Деякі об'єкти, наприклад сертифікати та асиметричні ключі, можуть утримуватися в базі даних, але не перебувати всередині схеми. Додаткові відомості про створення таблиць див. у розділі Tables.

SQL Server бази даних зберігаються у файловій системі у файлах. Файли можуть бути об'єднані у групи файлів. Додаткові відомості про файли та файлові групи див. у розділі Database Files and Filegroups.

Коли користувачі отримують доступ до екземпляра SQL Server, вони ідентифікуються як ім'я входу. Коли ви отримуєте доступ до бази даних, користувачі ідентифікуються як користувачі бази даних. Ім'я користувача бази даних може базуватися на імені входу. Якщо автономні бази даних включені, користувач бази даних може бути створений не на основі імені входу. Додаткові відомості про користувачів див. у розділі CREATE USER (Transact-SQL).

Користувач, який має доступ до бази даних, може отримати дозвіл на доступ до об'єктів цієї бази даних. Хоча дозволи і можуть бути надані окремим

користувачам, рекомендується створювати ролі бази даних, додаючи при цьому користувачів бази даних до відповідних ролей, а потім надавати дозволи ролям. Надання дозволів ролям, а не користувачам дозволяє легко і зрозуміло керувати процесом розподілу дозволів, незважаючи на постійну зміну та зростання кількості користувачів. Додаткові відомості про роздільну здатність ролей див. у розділі CREATE ROLE (Transact-SQL) та суб'єкти (ядро СУБД).

Робота з базами даних

Більшість користувачів, які працюють з базами даних, використовують SQL Server Management Studio. Засіб Management Studio має графічний інтерфейс користувача для створення баз даних і об'єктів у базах даних. Management Studio також має редактор запитів для взаємодії з базами даних шляхом написання вказівок Transact-SQL. Management Studio можна встановити з диска інсталяції SQL Server або завантажити з сайту MSDN. Додаткові відомості про засіб SQL Server Management Studio див. у розділі SQL Server Management Studio (SSMS).
SQL-процес

Коли ви виконуєте команду SQL для будь-якої СУБД, система визначає якнайкращий спосіб виконання вашого запиту, а механізм SQL визначає, як інтерпретувати завдання.

У цей процес входять різноманітні компоненти.

Ці компоненти -

- a) Диспетчер запитів
- b) Двигуни оптимізації
- c) Класичний механізм запитів
- d) Механізм запитів SQL тощо.

Класичний механізм запитів обробляє всі запити, відмінні від SQL, але механізм запитів SQL не оброблятиме логічні файли.

Нижче наведено просту діаграму, що показує архітектуру SQL.

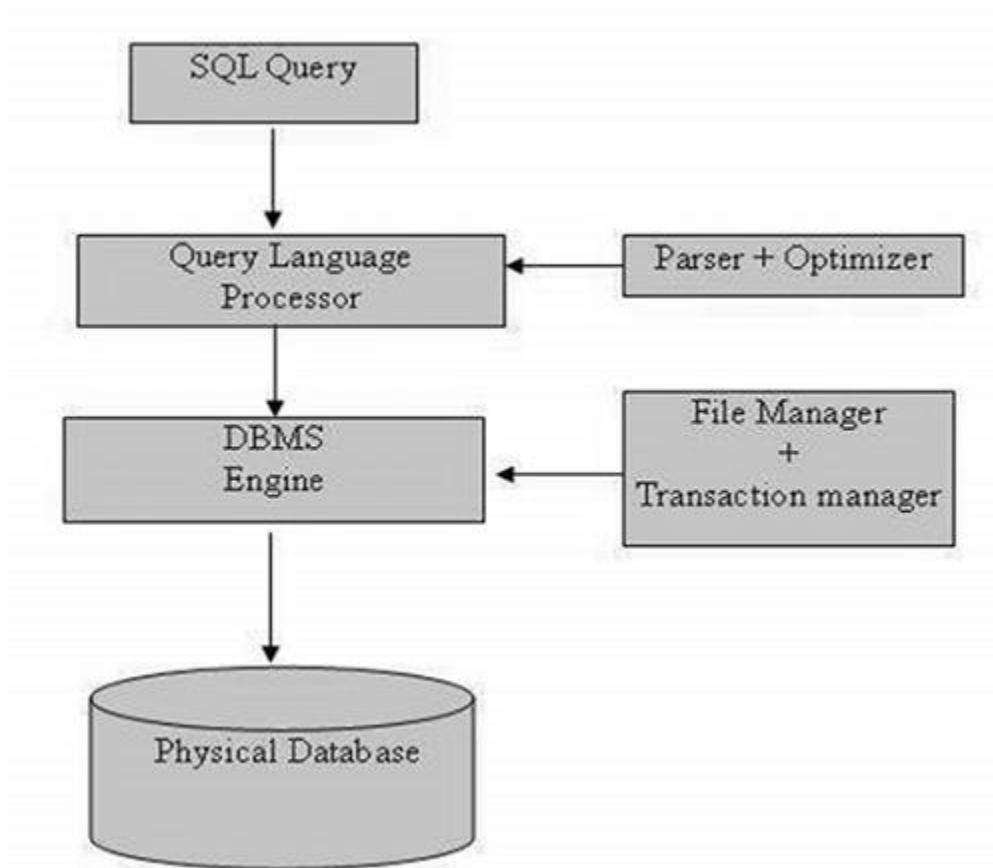


Рисунок 2.5– Діаграма,архітектури SQL

потрібний час, щоб стати екпертом з баз даних або досвідченим адміністратором баз даних. Все це приходить з великим досвідом у різних проектах баз даних та гарними тренінгами.

Але наступний список може бути корисним для початківців, щоб мати хорошу продуктивність бази даних:

BotFather - єдиний бот, який керує ними всіма. Це допоможе вам створювати нових роботів та змінювати налаштування існуючих.

Створення нового бота

Використовуйте /newbot для створення нового бота. BotFather запросить у вас ім'я та ім'я користувача, а потім згенерує токен аутентифікації для нового бота.

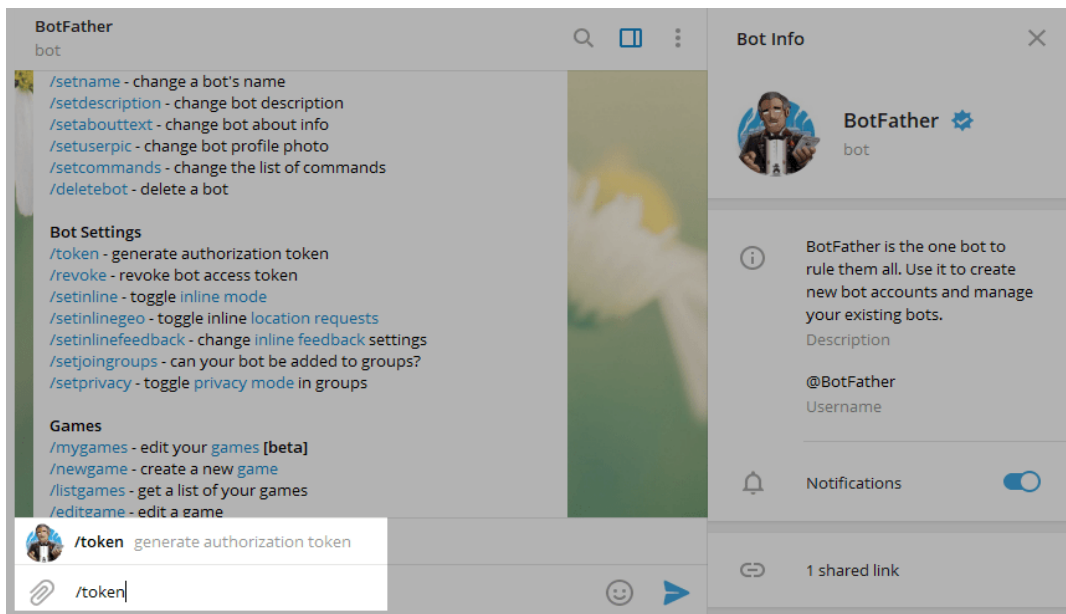


Рисунок 2.4– Пример роботи чат-бота BotFather

Ім'я вашого бота відображається в контактній інформації та інших місцях.

Ім'я користувача – це коротке ім'я, яке слід використовувати у згадках та посиланнях t.me.

Ім'я користувача має довжину до 32 символів та нечутливий до регістру може включати лише латинські літери, цифри та символи. Ім'я користувача бота має закінчуватися на « бот»

TelegramBot API – це інтерфейс на основі HTTP, створений для розробників, які займаються створенням ботів для Telegram.

Щоб дізнатися, як створити та настроїти роботи, перегляньте розділ «Вступ до роботів» та «Поширені запитання про роботи» .

3. СТВОРЕННЯ ТА РЕАЛІЗАЦІЯ ТЕЛЕГРАМ-БОТА

3.1. Планування розробки проекту

Для того, щоб розробити та мати можливість розвивати телеграм-бота буде використаний шаблонн життєвого циклу спірального моделі

Спіральна модель – показує дизайн існуючого спіральною петлі, створено системою. Незавершена робота на будь-якому етапі дозволяє перейти сразу до наступного етапу, не чекавши завершення поточного етапу. Виконуючи ітераційний підхід для розробки, відсутня робота можна буду виконати на наступний ітерації. Завдання – Якомога швидше представити готовий продукт користувачу, тим самим посилюючи вимоги

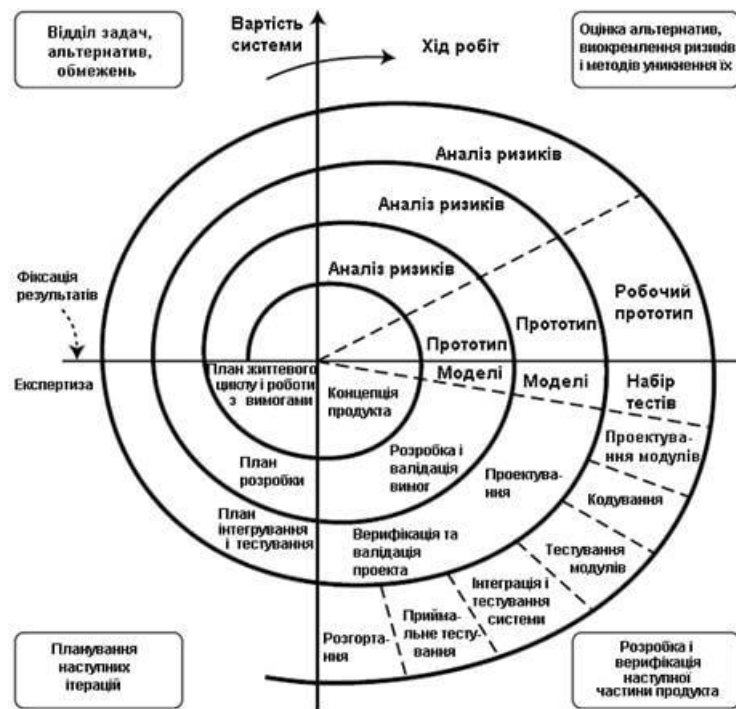


Рис 3.1 Шаблон спіральної моделі

Діаграма буде створена тільки після того як буде визначений елемент документації не буде відповідати фактичному стану розробки, внесення змін яка буде справна на задоволення потреб користувача.

Ідея цієї моделі від каскадної полягає можливість повертатися до формування

вимог і до розробки и робить цикл поки фінальний продукт не буде виконаний

Для програмного продукту така модель не дуже підходить з декількох причин

Перший пункт, вимоги замовника носить суб'єктивний характер, вимоги багаторазово повторюватися протягом розробки і навіть після завершення та випробування, і часом може з'ясуватися оригінал ідеї или продукта зовсім інший

Другий пункт, змінення обставини та умови користування системи,законом програмної інженерії є закон еволюції, де сформулюється до кожній ПС потребує зміни або виводить з експлуатації

Висновки: За допомогою діаграми життєвого циклу в бакалаврській роботі ми будемо доповнювати контентом телеграм бот і аналізувати подальшу роботу його.

На даний час ми маємо тільки мінімальний функціонал, малий контент, мало опрацьована зацькованість проекту.

3.2. Опис програми та її алгоритми

Діаграма прецедентів (або діаграма використання) – Діаграм що показую взаємодії акторами і прецедентам в системі. Прецедент - це граф, що складається з багатих акторів, прецедентів, обмежених систем, асоціація прецедентами та акторами, зв'язки між зв'язками та прецедентами та акторами. Суть діаграми прецедентів полягає в тому, що розроблена система представляється як набір сутностей або акторів, які взаємодіють із системою за допомогою так званих випадків використання. Варіанти використання використовуються для опису послуг, які система надає акторам. Іншими словами, кожен варіант використання визначає набір дій, які система виконує під час розмови з актором. При цьому про те, як буде реалізована взаємодія учасників із системою, не йдеться.

Різновиди UML стандартних видів зв'язки с акторами:

- a) асоціації
- b) включення
- c) розширення
- d) узагальнення

Відношення асоціації — використовується під час побудови всіх графічних моделей систем у формі діаграм.

Включення — відношення залежності між базовим варіантом використання і його окремим випадком. При цьому відношенням залежності є таке відношення між двома елементами моделі, за якого зміна одного елемента (незалежного) спричиняє зміну іншого елемента (залежного).

Розширення відношень визначається базовою варіантом використання з інакшими варіантами, функціональна поведінка якої разом з базовим не завжди, а тільки за виконання додаткових умов

Функції системи представлені UML діаграма прецедентів(Див. рис. 3.3)



Рисунок 3.3 - UML діаграма прецедентів

Ця діаграма зображує, що користувач може:

- Новий користувач може зробити анкети улюбленця
- Відправити на роботу домашнього улюбленця, щоб получить
- Купляти для вихованця їду а так-же аксесуари
- Годувати нашого домашнього вихованця
- Відправити нашого улюбленця спати, для того щоб обновити енергію на наступний день.

Алгоритм — визначений набір закінчених правил для виконання завдань класу або набір інструментів У старій інтерпретації замість слова «послідовний» використовувалося слово «послідовність», але з розвитком паралелізму в комп'ютерах слово «послідовність» взагалі почало замінюватися словом «послідовний». Незалежні інструкції можуть виконуватися паралельно в будь-якому порядку, якщо це дозволяє використовуваний виконавець. «Алгоритм» був написаний раніше російською мовою, яка зараз використовується рідко, за одним винятком Виконавцем, як правило, є комп'ютер, але поняття алгоритму не обов'язково застосовується до комп'ютерних програм – наприклад, чітко описаний рецепт також є алгоритмом, у цьому випадку виконавцем є особа (і, можливо, механізм, напр. ткацький або токарний верстат) числовим управлінням).

Можна виділити алгоритми обчислювальні (далі йдеться переважно про них) і управляючі. Обчислювальні алгоритми, насправді, перетворюють деякі початкові дані у вихідні, реалізуючи обчислення деякої функції. Семантика керуючих алгоритмів істотно може відрізнятися і зводитися до видачі необхідних керуючих впливів або в задані моменти часу, або як реакцію зовнішні події (у цьому випадку, на відміну від обчислювального алгоритму, керуючий може залишатися коректним при нескінченному виконанні).

Форми запису алгоритму:

- а) словесна або вербальна;
- б) псевдокод
- в) схемна:

1. структурограми

2. графічна

Властивості алгоритмів

Алгоритми мають ряд важливих властивостей:[30]

1. Скінченність

алгоритм завершуватись повинен завжди після того як виконання скінченної кількості кроків. Алгоритм ,без,можливо,скіченості називається метод обчислення яка має характеристику алгоритма

2. Дискретність

Процес який визначається тільки алгоритмом, можна розділити на окремі кроки, з яких кожний етап буде називатися кроком алгоритмічного процесу чи алгоритма

3. Вхідні дані

Алгоритм має певну кількість (можливо, нуль) вхідних даних, тобто значень, які були встановлені перед його запуском або значення яких були визначені під час виконання алгоритму. Вихідні дані Алгоритм має один або кілька вихідних даних, тобто значень, які мають явне відношення до вхідних даних.

4. Ефективність

Алгоритм вважається ефективним, якщо всі його оператори досить прості, щоб виконувати їх точно за скінченний час за допомогою олівця та аркуша паперу..

5. Масовість

Властивість алгоритму полягає в тому, що алгоритм повинен надавати рішення будь-якої задачі з подібного класу задач на будь-яких вхідних даних, які потрапляють в область дії алгоритму. Алгоритм універсального бота Telegram — це послідовність того, як користувач використовує гру Telegram. Наприклад, користувачі можуть вийти з Telegram безпосередньо перед тим, як почнуть спілкуватися зі своїми домашніми тваринами. Крім того, користувачі можуть негайно виконувати послідовні запити.

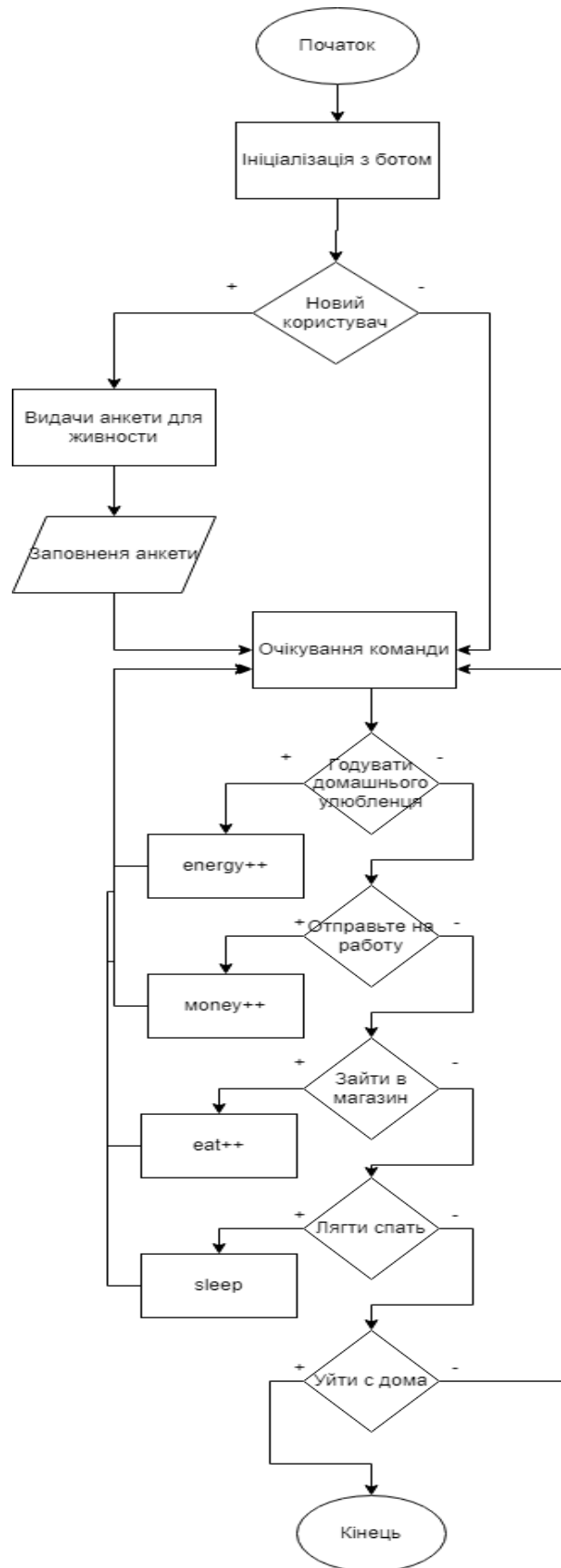


Рисунок 3.4 - Алгоритм роботи програми

3.3 Реалізація

У месенджера Телеграм відкритий API, тому без проблем підтримує роботу чат-ботів. Для створення в програмі є свій офіційний бот, який називається. Його ім'я потрібно ввести до рядка пошуку.

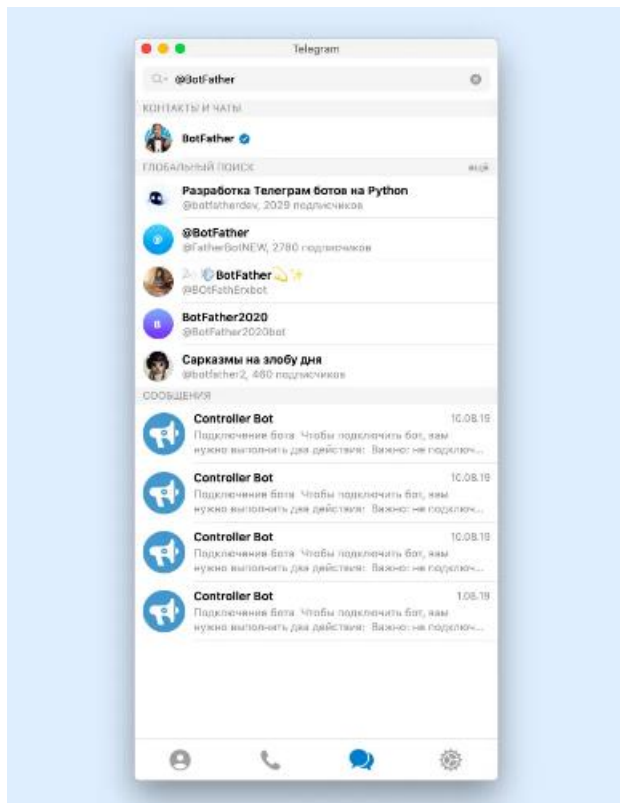


Рисунок 3.5 – Пошук BotFather

Далі натискаємо кнопку "Почати" внизу екрана - так запускається робота з ботом.

Натисніть кнопку "Почат."

Бот запропонує список команд, за допомогою яких можна керувати. Нам потрібна команда / Newbot - на неї можна просто натиснути.

Натисніть /newbot

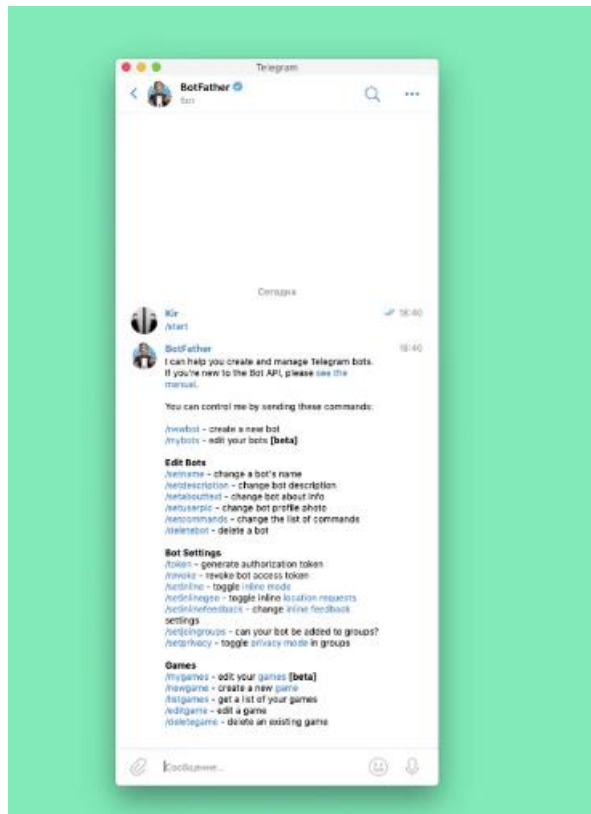


Рисунок 3.6 – Список команд

Нікнейм має бути унікальним (не повторюватися з іншими) та закінчуватися на слово bot.Новий бот створено. Телеграм відразу надсилає на нього посилання: t.me/ім'я_бота та токен - це спеціальний ключ від бота, за допомогою якого його можна підключати до сторонніх сервісів. Токен потрібно зберегти і нікому не показувати - він такий же важливий, як і пароль від пошти. Рекомендації щодо налаштування бота можна переглянути на команді /help.

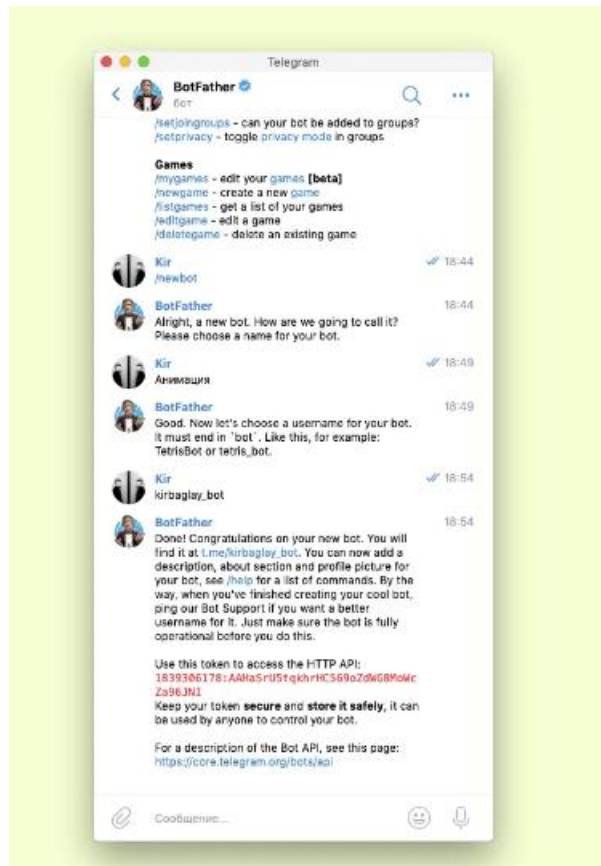


Рисунок 3.6 – Перелік команд /help

Новий бот створено. Всю інформацію щодо нього Телеграм надсилає одразу в одному повідомленні.

Після реєстрації чат-бота в самому Телеграмі, його потрібно налаштувати та підключити до будь-якого сервісу створення ботів. Але на цьому етапі важливо зрозуміти, які завдання повинен вирішувати бот і вибрати сервіс з відповідним функціоналом. Після цього можна буде налаштувати сценарій взаємодії з користувачем, додати кнопки та вибрати дизайн.

Після введення токена бот створюється і налаштовується, і можна приступати до роботи.

Після налаштування вам потрібно буде перейти до самого бота, щоб створити сценарій, додати меню або автоматизувати публікацію. Все залежить від місії бізнесу та для чого ви запускаєте чат-бота. все готово.

Якщо ви чогось не розумієте, можете написати в службу підтримки, але спочатку краще ознайомитися з розділами «Довідка» та «Налаштування» — вони

знаходяться на кнопках внизу інтерфейсу.

Сервіс для створення ботів в Telegram Як ми вже з'ясували, для легкої та зручної настройки бота можна використовувати сторонні сервіси, в тому числі безкоштовні. У бета-версії бакалаврської роботи спробуйте підключитися до бази даних і сервера

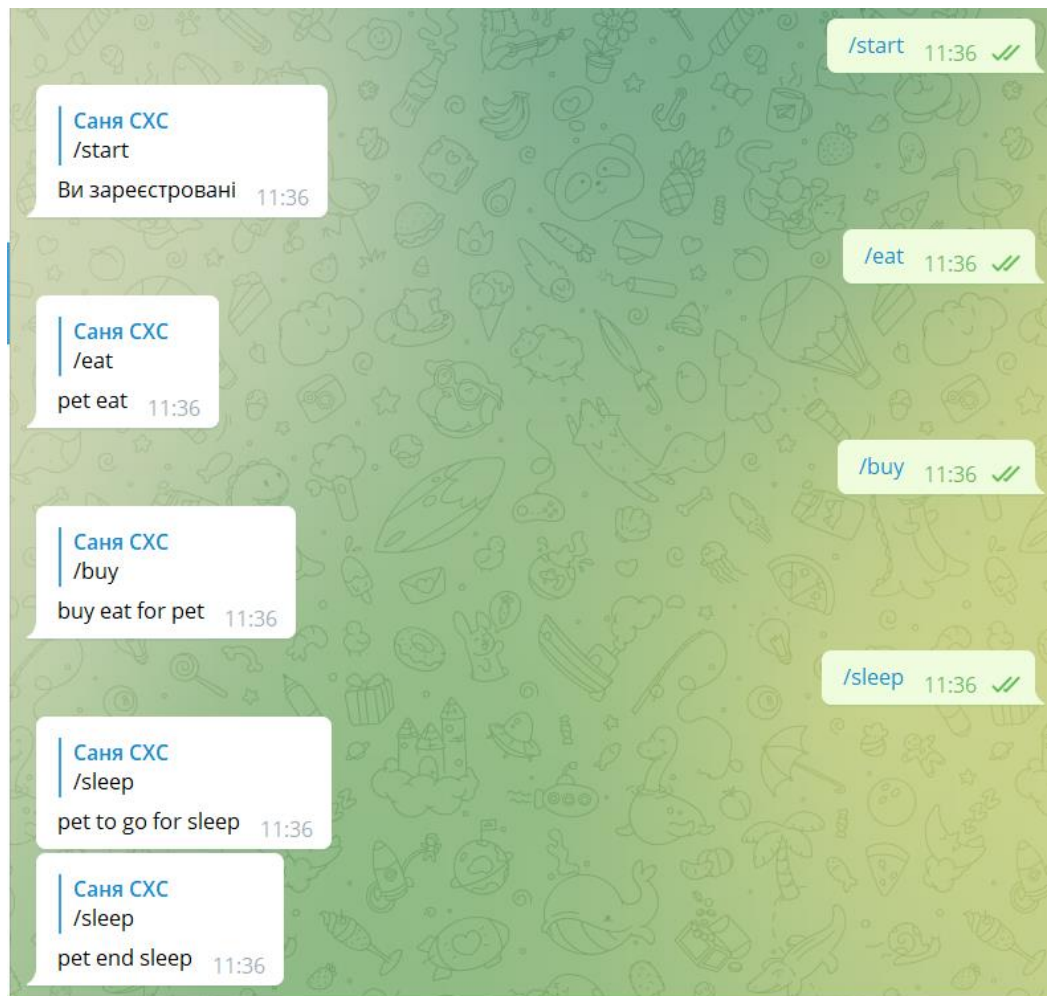


Рисунок 3.7 – проба з'єднатись базою даних и сервером

ВИСНОВКИ

1. У процесі проведення предметного дослідження дипломна робота визначає основні вимоги до інформаційної системи. Також вивчається впровадження інформаційних систем. Найбільшою перевагою чат-бота є його простота, тобто його можна використовувати скрізь, де можна спілкуватися через НТТР. Робот також може працювати на різних пристроях, включаючи смартфони, комп'ютери, планшети та інші пристрої, які можуть підключатися до глобального Інтернету.

2.Результатом моделі прецедентів

користувач має можливість виконувати базові функції гри.

3.Визначено послідовність виконання функцій, де представлена базова виконання програми в дії.

4.Архітектура являє собою сферичною моделлю життєвого цикла в бакалаврської роботі, яка буде в подальшому доповнюватися.

5.В роботі були реалізовані наступні технології:

- мова програмування – Python 3.8;
- модулі – TeleBot, pathlib,, googleapiclient.discovery.
- база даних – mySql;

Telegram,

ПЕРЕЛІК ПОСИЛАНЬ

1. А. С., Безносюк, Ю. С., Ведикер, П. К. & Воронович, Н. Е. Використання чат-ботів в навчальному процесі. The 2th International Conference on Digitalization 2019 [Електронний ресурс] – Режим доступу: https://elar.urfu.ru/bitstream/10995/82473/1/978-80-88327-04-2_017.pdf.
2. Statistical Data and Programming Languages Analysis for Developing a Chatbot in Telegram – 2020 - [Електронний ресурс] / Victoria O. Rytikova, Maxim A. Medvedev – Режим доступу: <https://aip.scitation.org/doi/epdf/10.1063/5.0041762>
3. Токарева Ю.А., Розробка чат-ботів і розмовних інтерфейсів. – Перев. с англ. – Київ: ДМК Пресс, 2018 р. – 340 стор.
4. Messenger and everything about it [Електронний ресурс] – <https://aboutmessengers.ru/messendzher-hto-eto-takoe>.
5. "Telegram Messenger". App Store. Retrieved 26 March 2021.
6. 'Nobody can block it': how the Telegram app fuels global protest". TheGuardian. Retrieved 7 November 2020.
7. Hakim, Danny (2 December 2014). "Once Celebrated in Russia, the Programmer Pavel Durov Chooses Exile". The New York Times. Retrieved 19 November 2015.
8. Telegram Reaches 1 Billion Daily Messages, Telegram, 8 December 2014
9. PostgreSQL: About [Електронний ресурс]: - Режим доступу: <https://www.postgresql.org/about/>
10. 1M rows/s from Postgres to Python [Електронний ресурс]: - Режим доступу: <http://magic.io/blog/asyncpg-1m-rows-from-postgres-to-python/>
11. Python's Requests Library (Guide) – Real Python [Електронний ресурс]: – Режим доступу: <https://realpython.com/python-requests/#getting-started-with-requests>
12. Bots: An introduction for developers [Електронний ресурс]: – Режим доступу: <https://core.telegram.org/bots>

13. What is Telegram and how different it is from other messaging apps [Электронный ресурс]: - Режим доступа: <https://justaskthales.com/en/telegram-different-messaging-apps/>

14. Intro to Chatbots [Электронный ресурс] - Режим доступа: <https://capacity.com/chatbots/intro-to-chatbots/>

15. Beautiful Soup Documentation – Beautiful Soup 4.9.0 documentation [Электронный ресурс]: – Режим доступа: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

16. datetime – Basic date and time types – Python 3.9.5 documentation [Электронный ресурс]: – Режим доступа: <https://docs.python.org/3/library/datetime.html#module-datetime>

Додаток А

Програмний код

```

import telebot
import traceback
import time
import mysql.connector
import updater
from telebot import types
from telebot.types import ReplyKeyboardMarkup

TOKEN = '5311348001:AAHq_g19Gd4CESJKQeie9goUr19EwuzDQ8E'

bot = telebot.TeleBot(TOKEN)
work = False

conn = mysql.connector.connect(host="localhost", database="pet", user="root")

cursor = conn.cursor()

# @bot.message_handler(commads=['start'])
@bot.message_handler(func=lambda message: message.text == '/start')
def start(message):
    sql = "SELECT id FROM user_pet WHERE user_id = %s"
    param = (message.from_user.id, )
    # param = (1, )
    cursor.execute(sql, param)
    res = cursor.fetchall()
    if len(res) < 1:
        bot.reply_to(message, "Вас не знайдено")
    else:
        print(res[0][0])
        bot.reply_to(message, "Ви зареєстровані")
    print(res)

@bot.message_handler(func=lambda message: message.text == '/test')

import telebot
import traceback
import time
import mysql.connector
import updater
from telebot import types
from telebot.types import ReplyKeyboardMarkup

TOKEN = '5311348001:AAHq_g19Gd4CESJKQeie9goUr19EwuzDQ8E'

bot = telebot.TeleBot(TOKEN)
work = False
eat = 100
energy = 100
money = 0

```

```

conn = mysql.connector.connect(host="localhost", database="pet", user="root")

cursor = conn.cursor()

# @bot.message_handler(commads=['start'])
@bot.message_handler(func=lambda message: message.text == '/start')
def start(message):
    sql = "SELECT id FROM user_pet WHERE user_id = %s"
    param = (message.from_user.id, )
    # param = (1, )
    cursor.execute(sql, param)
    res = cursor.fetchall()
    if len(res) < 1:
        bot.reply_to(message, "Вас не знайдено")
    else:
        print(res[0][0])
        bot.reply_to(message, "Ви зареєстровані")
    print(res)

@bot.message_handler(func=lambda message: message.text == '/test')
def starts_eat(message):
    param = (message.from_user.id, message.chat.id, 100, 100, 0, 0, )
    sql = "INSERT INTO user_pet (user_id, chat_id, eat, energy, money, work_time)
VALUES (%s, %s, %s, %s, %s, %s)"

    cursor.execute(sql, param)
    conn.commit()

@bot.message_handler(func=lambda message: message.text == '/eat')
def starts_eat(message):
    global eat
    global energy
    if eat == 100:
        bot.reply_to(message, 'pet no need eat')
    else:
        eat -= 25
        energy += 25
        sti = open('eat.jpg', 'rb')
        bot.send_sticker(message.chat.id, sti)
    # param = (message.from_user.id, message.chat.id, 75, 100, 0, 0, )
    # sql = "UPDATE INTO user_pet (user_id, chat_id, eat, energy, money,
work_time) VALUES (%s, %s, %s, %s, %s, %s)"
    bot.reply_to(message, 'pet eat +25')

@bot.message_handler(func=lambda message: message.text == '/buy')
def starts_eat(message):
    global eat
    global money
    if money == 0:
        bot.reply_to(message, 'pet no have money')
    else:
        eat += 50
        money -= 25
        bot.reply_to(message, 'buy eat for pet')
        sti = open('buy.jpg', 'rb')
        bot.send_sticker(message.chat.id, sti)

```

```

@bot.message_handler(func=lambda message: message.text == '/sleep')
def starts_eat(message):
    global energy
    energy += 25
    # param = (message.from_user.id, message.chat.id, 75, 100, 0, 0,)
    # sql = "UPDATE INTO user_pet (user_id, chat_id, eat, energy, money,
work_time) VALUES (%s, %s, %s, %s, %s, %s)"
    bot.reply_to(message, 'pet to go for sleep')
    sti = open('sleep.jpg', 'rb')
    bot.send_sticker(message.chat.id, sti)
    time.sleep(10)
    bot.reply_to(message, 'pet end sleep')
    sti = open('weekup.jpg.jpg', 'rb')
    bot.send_sticker(message.chat.id, sti)

@bot.message_handler(func=lambda message: message.text == '/work')
def mess(message):
    global money
    global energy
    if energy == 0:
        bot.reply_to(message, 'pet not working for slow energy')
    elif eat == 0:
        bot.reply_to(message, 'pet not working for hungry')
    else:
        energy -= 50
        bot.reply_to(message, 'pet to go for work')
        sti = open('work.png', 'rb')
        bot.send_sticker(message.chat.id, sti)
        time.sleep(20)
        money += 50
        bot.reply_to(message, 'pet end work')

@bot.message_handler()
def message_handler(message):
    user_id = message.from_user.id
    if message.text == "Реєстрація":
        keyboard = types.ReplyKeyboardMarkup(resize_keyboard=True)
    else:
        bot.reply_to(message, "Не розумію")

bot.polling()

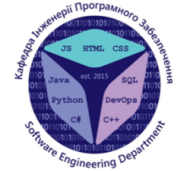
```

Додаток Б

Презентація



ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



Розробка телеграм-боту «Тамагочі» мовою Python

Виконав студент 4 курсу
групи ПД-43
Лугина Олександр Володимирович

Київ – 2022

Аналоги



Ligmar - Fantasy MMORPG



Жаба-бот



Ферма: Mining

Порівняльна таблиця програм телеграм-игр

Назва бота	Ligmar - Fantasy MMORPG	Ферма: Mining	Жаба-бот
Платформа	Telegram	Telegram	Telegram
Популярність	-	+	+
Простота старту	+	-	+
Успішність бота, фінансово	+	+	+
Складність проекту	+	+	-
Взаємодія з гравцями	+	-	+

Постановка задачі

На даний момент при створенні диплома телеграм-бота "Тамагочі" на основі python я хочу побачити базову логіку і команди жанру Тамагочі. Самого виховання якого:

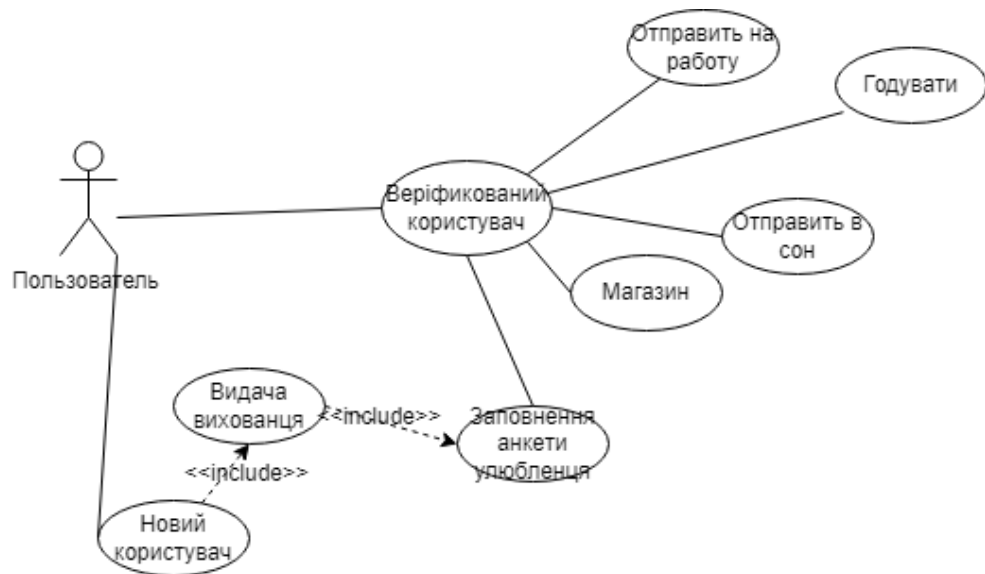
- Треба годувати, щоб не помер з холоду;
- Давати йому спати, щоб відновив енергію щоб відправитися на роботу;
- Відправляти на роботу, щоб заробити грошей для покупок їжі, енергії і т.д.

Аналіз аналога телеграм-бота «Жаба-бота»

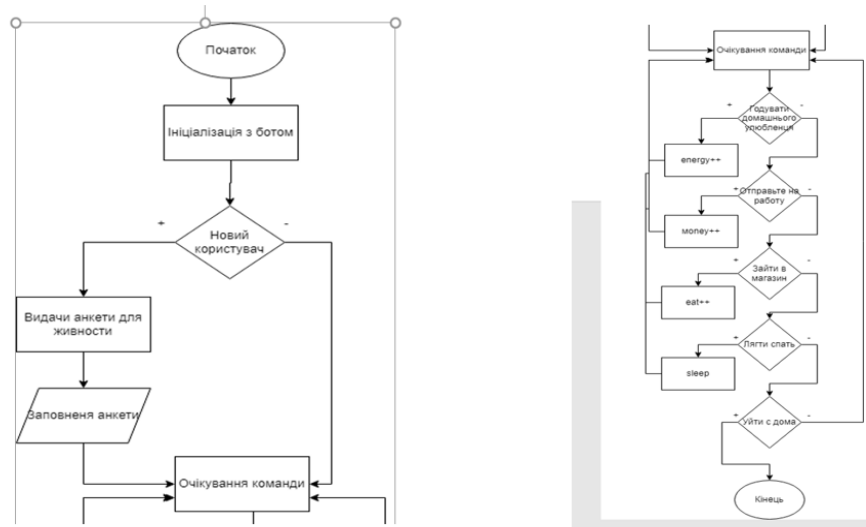
«Жабобот із жабами та жабами»: як «кишеньковий» тамагочі з елементами MMORPG об'єднав у чатах тисячі людей

Понад сто тисяч «жабокористувачів» щодня годують віртуальних вихованців та відправляють їх на війни.

Засоби реалізації



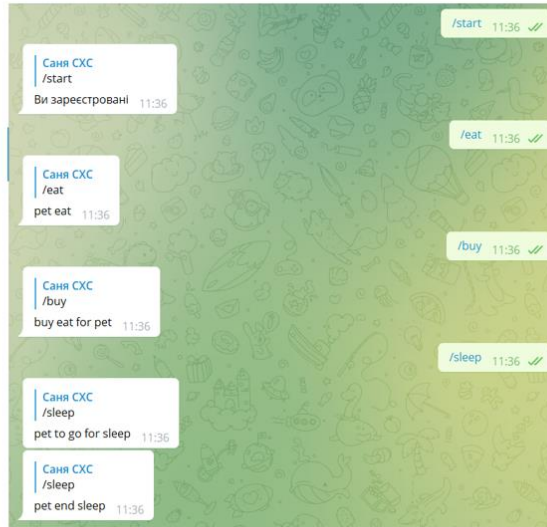
Алгоритм роботи бота



Апробація результатів дослідження

- 1. Лугина О.В. Дослідження чат-ботов в телеграммі :Матеріали науково-технічної конференції: «Застосування програмного забезпечення в ІКТ». 44-47стр. Збірник тез.\- 20.04.2022

Пример работы телеграм-бота



9

ДЯКУЮ ЗА УВАГУ!