

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ДОПУСТИТИ ДО ЗАХИСТУ

Завідувач відділенням
комп'ютерних технологій
Наталія СТЕФУРАК /_____/

підпис

«__» _____ 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту
освітньо-кваліфікаційного рівня «молодший спеціаліст»
зі спеціальності 122 «Комп'ютерні науки»

на тему: «Створення Telegram боту для перегляду та оприлюднення результатів
змагань з ловлі хижої риби »

Студент групи КН-41	Огньов І.І.	_____
		(підпис)

Керівник проекту	Чубей О.О.	_____
		(підпис)

Консультанти:

з техніко-економічного обґрунтування	Меленчук Л.І.	_____
		(підпис)

нормоконтролер	Кульчинська Н.З.	_____
		(підпис)

Тернопіль – 2023

Галицький фаховий коледж імені В'ячеслава Чорновола
відділення комп'ютерних технологій
циклова комісія інформатики та комп'ютерних дисциплін

ЗАТВЕРДЖУЮ

Завідувач відділенням
комп'ютерних технологій
Наталія СТЕФУРАК /_____/

підпис

«__» _____ 2022 р.

ЗАВДАННЯ

на дипломне проєктування
на здобуття освітньо-кваліфікаційного рівня «молодший спеціаліст»
студенту Огньов Ігор Ігорвич
(прізвище, ім'я та по-батькові студента)

1. Тема проєкту Створення Telegram боту для перегляду та оприлюднення
результатів змагань з ловлі хижої риби

затверджена наказом по коледжу від “16” грудня 2022 р., № 1193

2. Термін здачі студентом завершеного проєкту “__” _____ 2023 р.

3. Вихідні дані до проєкту Діяльність рибацьких спілок, існуючі рішення
для висвітлення результатів змагань

4. Перелік питань, які повинні бути розроблені в проєкті:

а) основна частина дослідження розробки Telegram ботів, розробка та реалізація за допомогою мови програмування «Java» та створення бази даних.

б) техніко-економічне обґрунтування Розрахунок всіх затрат на виробництво тобто затрати на розробку та реалізацію і ресурси також розрахунок собівартості продукту.

5. Перелік графічного матеріалу JPG-зображення роботи боту в Telegram також уточнюючі зображення що до розробки боту, таблиці

6. Консультанти проєкту: _____

Розділ	Консультанти	Підпис, дата	
		Завдання видано	Завдання прийнято
з техніко-економічного обґрунтування	Меленчук Л.І. (вчена ступень, звання П.І.Б. _____ консультанта)		

КАЛЕНДАРНИЙ ПЛАН дипломного проєктування

№ п/п	Найменування етапу	Терміни	
		початку	завершення
1.	Вибір теми, ознайомлення з вимогами до дипломного проєктування.	21.11.2022 р.	04.12.2022 р.
2.	Огляд типових рішень та написання відповідного розділу ПЗ	05.12.2022 р.	25.12.2022 р.
3.	Дослідження технологій реалізації та написання відповідного розділу ПЗ	23.01.2023 р.	13.02.2023 р.
4.	Розробка функціональних вимог до проєкту та робота над структурою програмного продукту. Написання відповідного розділу ПЗ	14.02.2023 р.	24.02.2023 р.
5.	Встановлення на налаштування середовища реалізації	1.03.2023 р.	23.03.2023 р.
6.	Проектування програмного засобу (функціоналу, інтерфейсу, бази даних продукту)	27.03.2023 р.	10.04.2023 р.
7.	Реалізація та налаштування програмного засобу	11.04.2023 р.	05.05.2023 р.
8.	Доопрацювання модулів	08.05.2023 р.	29.05.2023 р.
9.	Тестування на налагодження продукту	30.05.2023 р.	05.06.2023 р.
10.	Опрацювання економічного розділу	06.06.2023 р.	12.06.2023 р.
11.	Робота над оформленням пояснювальної записки	12.06.2023 р.	17.06.2023 р.
12.	Попередній захист дипломного проєкту, доопрацювання	15.06.2023 р.	
13.	Підготовка до захисту дипломного проєкту	16.06.2023 р.	25.06.2023 р.
14.	Захист дипломного проєкту	26.06.2023 р.	27.06.2023 р.

7. Дата видачі завдання “___” _____ 2022 р.

Керівник Чубей О.О.

Завдання прийняв до виконання Огнєв І.І.

Реферат

Дипломний проєкт. Тема: «Створення Telegram боту для перегляду та оприлюднення результатів змагань з ловлі хижої риби » містить сторінок – 91, рисунків - 21, таблиць - 5, джерел - 36, додатків - 3.

Метою проєкту є Створення Telegram боту для перегляду та оприлюднення результатів змагань з ловлі хижої риби .

Об'єктом дослідження для подальшої розробки є існуючі Telegram боти та їхня взаємодія з користувачем .

Головним завданням є створення функціонуючого та зручного Telegram боту .

Результатом дипломного проєктування став Telegram бот для перегляду та оприлюднення результатів змагань з ловлі хижої риби.

Під час реалізації проєкту використовувався Telegram – це зручний месенджер який дає змогу своїм користувачам розміщати свої проєкти. Середовищем розробки було обрано Java – об'єктно-орієнтована мова програмування, IntelliJ IDEA - це потужне і популярне середовище розробки, яке в першу чергу орієнтоване на роботу з Java.

Abstract

Diploma project. Topic: "Creating a Telegram bot for viewing and publishing the results of predatory fish fishing competitions" contains 94 pages, 21 figures, 5 tables, 36 sources, 3 appendices.

The goal of the project is to create a Telegram bot for viewing and publishing the results of predatory fish fishing competitions.

The object of research for further development is the existing Telegram bots and their interaction with the user.

The main task is to create a functioning and convenient Telegram bot.

The result of the diploma project was a Telegram bot for viewing and publishing the results of predatory fish fishing competitions.

During the implementation of the project, Telegram was used - it is a convenient messenger that allows its users to post their projects.

The development environment was chosen to be Java, an object-oriented programming language, IntelliJ IDEA is a powerful and popular development environment that is primarily focused on working with Java.

ЗМІСТ

Вступ.....	7
1 Аналіз предметної області постановка задачі.....	10
1.1 Система обґрунтування доцільності.....	10
1.2 Огляд існуючих рішень.	16
2 Проектування системи.....	22
2.1 Формалізація вимог до системи.	22
2.2 Проектування систем.....	22
2.3 Проектування структур бази даних.....	30
3 Реалізація тестування системи.....	34
1.3 Огляд засобів організації.....	34
3.2 Реалізація функціоналу чат-боту.....	63
4 Техніко-економічне обґрунтування	65
4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт	65
4.2 Оцінка витрат на оплату праці та відрахувань на соціальні заходи	66
4.3 Розрахунок витрат на електроенергію	69
4.5 Обчислення накладних витрат.....	70
4.6 Складання кошторису витрат та визначення собівартості робіт	70

					ДП. КН 23.519.03.000 ПЗ							
Зм.	Арк.	№ докум.	Підпис	Дата								
Розроб.		Огньов І.І.			Створення Telegram боту для перегляду та оприлюднення результатів змагань з ловлі хижої риби			Літ.	Арк.	Архівів		
Перев.		Чубей О.О.								5	90	
Реценз.		Гавришків Н.Г.						ГФК.ВКТ.ЦКІКД. КН-41				
Н.контр.		Кульчинська										
Зав. від.		Стефурак Н.А.										

Висновки	72
Перелік джерел посилання	74
Додатки.....	75

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		6

ВСТУП

Двадцять перше століття є віком глобалізації, в якій переважаючою тенденцією є повсюдний інтенсивний розвиток інформаційних технологій. Компонентом таких технологій є соціальні мережі, які виступають ресурсами, що забезпечують взаємовідносини серед індивідуальних користувачів, і навіть організацій мережі інтернет.

Сьогодні неможливо уявити життя людини без глянсових зображень у Instagram та безлічі повідомлень Facebook, WhatsApp, Telegram, Viber, Discord. Соціальні мережі впевнено вторглися у життя людей і здобули величезну популярність.

Такі інформаційні ресурси виконують комунікативну функцію для багатьох жителів планети, дозволяють знайти партнерів та однодумців практично в будь-якій галузі, відкривають доступ до величезного набору медіа матеріалів, дають можливість творчо реалізувати себе.

У загальному випадку будь-яка соціальна мережа представляє сайт, за допомогою якого зареєстровані учасники мають можливість спілкуватися між собою та ділитися різними видами інформації (текстовий, відео, аудіо, за допомогою інтернет-додатків).

Разом зі з ростом використання мобільних технологій, виникла потреба у створенні додатків для мобільних пристроїв на базі операційних система Android та iOS. Тому майже для усіх соціальних мереж було створено мобільні додатки. А також з'явилася купа різних месенджерів, що спочатку підтримували обмін текстовими повідомленнями через інтернет, а потім доповнилися можливостями пересилки графічних об'єктів, прослуховуванням та обміном аудіо, та відео повідомленнями, а також фалами різних форматів.

Крім того було створено можливість спілкування з чат-ботами. Бот – це програма, яка створена, щоб виконувати однотипні та повторювані завдання за

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		7

певним алгоритмом. Вона економить час людей беручи на себе рутинні функції, а працює через інтерфейси зі швидкістю, яка набагато вища за людську.

Наразі існує велика кількість чат-ботів для різних месенджерів, що виконують різноманітні задачі – від представлення анекдотів користувачу до замовлення різного роду квитків, повідомлення курсу валют та бізнес-аналітики.

Фактично боти замінили людей у деяких сферах діяльності та спростили отримання тієї чи іншої інформації з використанням зручного та інтуїтивно зрозумілого інтерфейсу. Також бот надає доступ до потрібної інформації користувачеві цілодобово.

Кожна компанія, що створила той чи інший месенджер, додає для розробників спеціальний інтерфейс для розробки – API (Application Programming Interface або інтерфейс програмування додатків) – це сукупність інструментів та функцій у вигляді інтерфейсу для створення нових програм, завдяки якому одна програма буде взаємодіяти з іншою. Це дозволяє розробникам розширювати функціональність свого продукту та пов'язувати його з іншими.

Більшість великих компаній розробляють API для клієнтів або внутрішнього використання. Звичайні користувачі також застосовують різні API.

Метою і завданням дослідження є створення чат-боту з використанням Telegram Bot API для відображення результатів змагань з ловлі хижої риби спінінгом з берега.

Об'єктом дослідження є :

- Об'єктно-орієнтована мова програмування Java.
- IntelliJ IDEA IDE – інтегроване середовище для розробки програм на мові Java.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		8

- Поняття API. Telegram Bot API — представляє собою HTTP-інтерфейс для роботи з ботами в Telegram.
- Spring Framework – найпопулярніший Java-фреймворк для створення проектів різної складності та архітектури.
- Gradle – інструмент для автоматичного збирання проекту та завантаження і використання додаткових бібліотек.

Предмет дослідження – технологія створення Телеграм-боту мові Java з використанням Telegram Bot API.

Наукова новизна одержаних результатів полягає в наступному:

Створення автоматизованої системи представлення результатів змагань з ловлі хижої риби, що допоможе зняти деяке навантаження у роботі з організаторів змагань. Та надасть цілодобовий доступ учасникам та людям, що цікавляться змаганнями, до результатів у форматі 24/7.

Також отримання зручного та функціонального помічника у вигляді телеграм-боту.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		9

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ПОСТАННОВКА ЗАДАЧІ

1.1 Система обґрунтування доцільності

Користувачі спілкуються в соцмережі з іншими користувачами або запитують на сайті виробника співробітника, але часто не розуміють, що говорять з ботом. Потім йдуть записуватись на стрижку в чаті Telegram і знову спілкуються з ботом.

Бот – це програма, яка створена, щоб виконувати однотипні та повторювані завдання за певним алгоритмом. Вона економить час людей беручи на себе рутинні функції, а працює через інтерфейси зі швидкістю, яка набагато вища за людську.

Бот – скорочений від «робот». Так називають автономну програму, яка працює в інтернеті та виконує за людей деякі завдання. Так виходить звільнити людські ресурси там, де їх недостатньо або їх використання невігідно.

Наприклад, бот може бути консультантом. Його інтегрують у чати та на сайти для створення там активності та відповідей на прості запитання. Якщо він не впорається з поставленим завданням, то переведе діалог на реальну людину з підтримки. Так вдається знизити навантаження реальних фахівців, зменшити потік нецільових звернень до них.

Боти в Telegram, наприклад, завантажують фільми (наприклад, @kinomanBot), конвертують файли (@topdf_bot), допомагають засвоювати іноземні мови (andyrobot), знаходити пісні в інтернеті за уривком (@auidbot), навіть заробляти в інтернеті (@Skinve – показує друзям рекламу в Skype, де заробляє користувач). Вони перекладають тексти, розважають, допомагають та розповідають, як витратити час із користю.

Бот може виконувати будь-які завдання, закладені у нього розробниками. Наприклад, може надсилати електронні листи, відповідати на запитання користувачів, надсилати їх на сторінки сайту або до фахівця. За допомогою

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		10

робота можна дізнатися курс валюти, завантажити музику, знайти книги, поширити якусь інформацію, навіть вплинути на репутацію бренду.

Основне завдання будь-якого бота – звільнити людину від роботи, яку їй важко виконувати – одноманітною, постійно повторюваною та нудною. Багато компаній впроваджують ботів, щоб заощадити, зменшити час очікування та підвищити клієнтоорієнтованість. Головне – налаштувати бота так, щоб він не відлякнув клієнта мовчанням, неправильними відповідями чи дурною поведінкою.

Наприклад, якщо бот ставитиме нескінченні уточнюючі питання, то відвідувач не перейде до покупки товару, він просто закриє сторінку та купить у конкурентів.

Ось основні сфери, в яких активно застосовуються боти:

- техпідтримка;
- розрахунок вартості товару;
- консультування з різних питань, наприклад, який кредит клієнту вибрати;
- прогрів користувача та підготовка його до покупки товару;
- розвага відвідувачів;
- лідогенерація - наприклад, бот проводить опитування клієнта, відповідає на його запитання, збирає дані у форму та експортує їх у crm-систему;
- надання необхідної інформації, зокрема контактної;
- відправка КП (комерційної пропозиції);
- допомогу клієнту у виборі конкретного товару із двох моделей;
- оформлення замовлення.

Він може виконувати інші корисні функції. Головне – навчити його це робити. Це не всі галузі, для яких використовуються такі програми.

У соціальних мережах ботів часто використовують, щоб розкрутити групу або обліковий запис. Це потрібно, щоб створити видимість популярності місця для великої аудиторії, підвищити охоплення та кількість показів у стрічках новин. Одні роботи будуть писати коментарі, інші – відповідати на них, ставити лайки, треті – додаватися до груп.

Боту можна довірити різні дії:

- реєстрацію нових користувачів;
- ведення логів чатів;
- контроль за правами учасників;
- блокування користувачів за порушення правил, ненормативну лексику та загрози.

Користувачі все рідше відкривають пошту та розглядають сайти. Вони рідше хочуть дзвонити чи ходити до магазинів. Їм хочеться купити товар у декілька кліків мишкою за кілька секунд. Їм ніколи чекати, коли прийде або зателефонує менеджер. Тому компанії, які користуються чат-ботами, отримують перевагу перед конкурентами.

Чим допоможе бот у бізнесі:

- Заощадить гроші. Деякі обов'язки працівників будуть передані віртуальному помічнику. Можна більш продуктивно використовувати час працівників або не наймати ще одного працівника.
- Збільшить продаж. Якщо клієнти будуть отримувати відповіді на всі свої запитання, то їхня лояльність збільшиться, а сумніви зникнуть. Більше клієнтів придбає товар.
- Буде на зв'язку 24/7. Якщо потрібно терміново відповісти на просте запитання, то бот завжди на зв'язку, відповідь миттєво. Він створить ефект присутності, який подобається багатьом клієнтам.
- Зменшить час очікування. Ніхто не любить чекати у черзі. Чим швидше клієнт отримає відповідь, тим вища ймовірність продажу.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		12

Усередині підрозділів організацій роботи допомагають у внутрішніх запитах, допомагають операторам.

- Збільшить мобільний трафік. Це найпростіший спосіб бути встановленим у телефоні клієнта, тому що роботи не займають багато місця.
- Повідомить про зміни в системі клієнтів та співробітників. А також нагадує необхідність закрити завдання або оплатити покупку.
- Проведе активність. Якщо прописати йому сценарій та описати точку старту, то він впорається із проведенням конкурсу, визначенням переможця і навіть сам йому напише.

Багато в чому саме бюджет майбутнього власника визначається, наскільки «розумним» буде бот. Всі його можливості можна розділити на 2 види:

- Обмежені. Відповідають певну кількість чітко заданих питань. Вони не вчаться, але й коштують дешевше. Його можливості залежать тільки від того, що було прописано в алгоритмі.
- Самонавчання. Створюється складна система, в якій діє нейронна мережа, що самонавчається. Цей вид ботів значно дорожчий і вимагає супроводу людини, особливо спочатку. Але такий робот зможе вловити, про що говорить користувач, та допомогти йому.

Боти всюди. Одні допомагають людям і приносять користь, а інші шкодять, якщо їх створили з такими намірами. Люди часто порівнюють ботів з мобільними програмами і їм складно зрозуміти, що краще - бот або програму.

Чат-бот має багато переваг перед мобільним додатком. Власники компаній поспішають – потрібно якнайшвидше запуснитися, залишатися попереду конкурентів, працювати швидше і краще. Їм здається, що простіше працювати через пошту чи сайт. Створення бота здається непотрібним заняттям, а насправді часто це новий канал продажів, який наводить нових клієнтів.

Але насправді клієнти часто не хочуть користуватися сайтами. А програм вже так багато, що вони швидко набридають і їх видаляють, іноді навіть жодного разу не відкривши. Крім того, пам'яті в телефонах у багатьох клієнтів мало, тому чергову програму вони встановлюють неохоче. За статистикою GlobalWebIndex близько 80% користувачів активно користуються всього трьома програмами, серед яких є хоча б одна соцмережа або месенджер.

Створити робота компанії в рази дешевше, ніж створити цілу програму. Після розробки доведеться займатися тривалим налагодженням та налаштуванням програми. Причому це вкладення грошей це не гарантує, що після налаштування додаток буде цікавим для клієнтів.

У роботі простіше створити сценарій для комунікації. Наприклад, для запису до перукаря людині не захочеться завантажувати програму. Але він легко напише роботу в списку контактів.

Боти портативні. Вони інтегруються в навігатор, програму, на сайт, в програму особистого помічника або на іншу платформу, куди цього хоче користувач. Яскравими прикладами можна назвати роботів на сайті будь-якого банку, роботів у групових чатах різних соцмереж.

Тому зацікавити аудиторію через якісь нові програми набагато складніше, ніж через уже встановлені. Це підтверджують дані із Quantcast. Особливо це стосується месенджерів та соцмереж, які є майже у кожного. Але це не єдині переваги використання роботів.

Боти все частіше використовують для роботи з клієнтами. Під час роботи з клієнтами бот виконує такі операції:

- уточнення;
- замовлення;
- оформлення;
- відповіді на запитання;
- консультації та допомогу в купівлі та прийнятті оплати.

Для роботи зі співробітниками не менш популярними є корпоративні чат-боти, яких використовують для надання послуг співробітникам фірми. Це є певні електронні дії, які підприємство пропонує працівникам.

Через нього можна:

- виконувати замовлення довідок;
- з'ясовувати дату початку відпустки;
- уточнювати час роботи керівництва чи певного співробітника.

Великі компанії пропонують клієнтам на вибір кілька різних чат-ботів із соціальних мереж. Можна вибрати зручний спосіб зв'язку (Рис. 1.1).

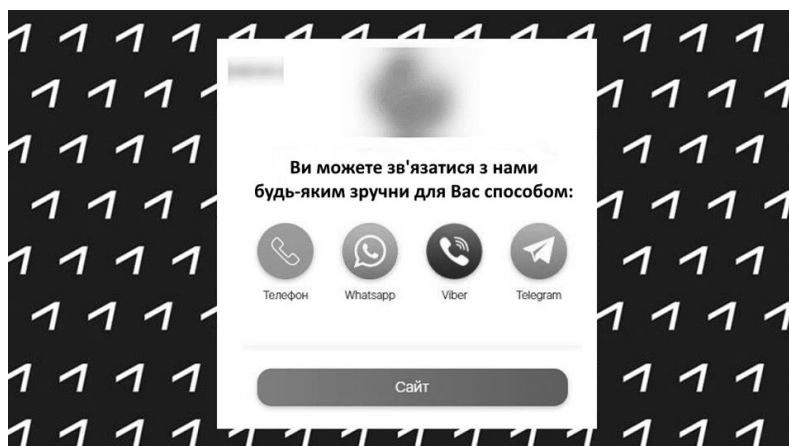


Рисунок 1.1 – Вибір зручного способу зв'язку.

Наприклад, на Aliexpress є свій бот, його звать Eva. Поки що він відповідає не на всі запитання, але він навчається.

Telegram це великий месенджер, у якому можна обмінюватися повідомленнями, медіафайлами, вести групи та здійснювати дзвінки. Тут легко знайти робота.

Там вони інтегруються для взаємодії з клієнтами, відповідають на їхні запитання, пропонують товари та способи доставки.

Існує величезна кількість роботів в Telegram, тому що тут їх реалізувати просто – це чати, в яких користувачі звикли листуватися. Ще один контакт вони охоче додають, а потім легко взаємодіють з ним.

Вони активно застосовуються для:

- розваги користувачів;
- онлайн-консультацій;
- пошуку інформації;
- збирання інформації від користувачів.

Щоб створити бота, необхідно спочатку зареєструвати його через внутрішній бот в Telegram. Називається бот @BotFather. Потім можна скористатись спеціальною програмою або мовою програмування Java, Python або Java Script.

Популярні боти: @topdf_bot – конвертує в PDF, @filesbot – зберігає документи та ділиться з користувачами, @tap2bot – шукає книги, авіаквитки, фільми, таксі, готелі та багато іншого.

Щороку створюється безліч нових ботів, щоб спростити ведення бізнесу і перекинути частину рутинних завдань на штучний інтелект. Але це не єдине їхнє призначення. Їх активно використовують у соціальних мережах, для реєстрації та заробітку грошей. Можна налаштувати такого бота і звільнити час для створення контенту.

1.2 Огляд існуючих рішень.

DeerPaintBot – це телеграм-бот, який може генерувати зображення в стилі різних художників та режимах, використовуючи нейромережі та штучний інтелект (рис.1.2).



Рисунок 1.2 – DeepPaintBot

Один з головних переваг бота – це його інтерфейс. Він має просте та зрозуміле меню, яке дозволяє користувачам швидко вибрати потрібний стиль, а також зображення, що будуть перетворені. Крім того, користувачі можуть вибрати якість та розмір зображення, що також є дуже зручно.

Ще одна перевага бота полягає в тому, що він безкоштовний та не має реклами. Це робить його доступним для користувачів з різних країн, а також робить використання бота приємним та зручним.

Однак, є деякі мінуси, які можуть зменшити задоволення від використання бота:

- Передбачення невірних кольорів в окремих випадках.
- Недостатній рівень деталізації у створеному зображенні.
- Обмеження у виборі формату вхідного зображення.

Для вирішення цих проблем можна запропонувати наступні кроки:

Оновлення моделі навчання з використанням більшої кількості даних з відповідними кольоровими просторами для покращення передбачення кольору.

Додаткова обробка зображень для збільшення рівня деталізації на виході.

Розширення форматів вхідних зображень для зручності користувачів.

Також можна вдосконалити інтерфейс користувача та функціональність, щоб зробити бот більш інтуїтивно зрозумілим та зручним у використанні. Це можна зробити за допомогою додавання нових функцій, таких як можливість змінювати розмір зображення, вибір кольорових просторів, можливість виправляти деякі помилки тощо.

Крім того, можна розглянути можливість додавання інших моделей навчання, що можуть допомогти вирішувати проблеми з передбаченням кольорів та рівнем деталізації. Загалом, варто продовжувати вдосконалювати бота та виправляти його недоліки, щоб зробити його більш корисним для користувачів.

Загалом, DeepPaintBot є корисним телеграм-ботом для генерації зображень. Він має простий та зрозумілий інтерфейс, безкоштовний та без реклами, що робить його доступним та зручним для користувачів. Однак, можуть бути проблеми з точністю результатів та часом відповіді бота.

Filtered News – це телеграм бот, який забезпечує користувачів персоналізованими новинами на основі їхніх інтересів та вподобань. Користувач може обрати тему, яка його цікавить, таку як спорт, політика або наука, і отримувати щоденні новини з цих областей.

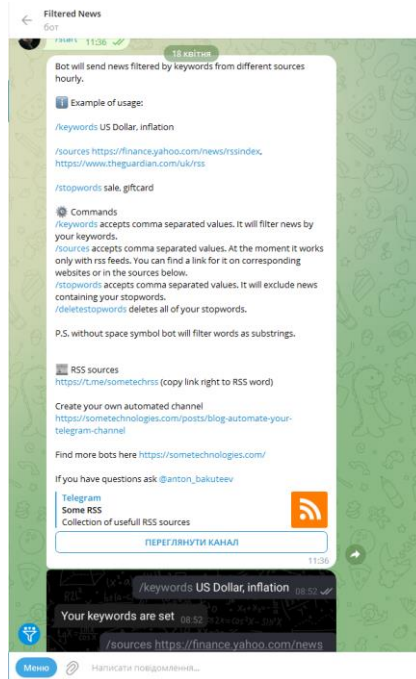


Рисунок 1.3 – Filtered News

Бот працює на основі алгоритмів машинного навчання та штучного інтелекту, щоб розуміти вподобання користувача та надавати новини, які йому можуть сподобатися.

Переваги: Filtered News бот автоматично фільтрує новини з різних джерел та категорій, що дозволяє користувачам економити час на пошуку цікавих новин. Крім того, бот дозволяє налаштувати власні фільтри, що сприяє більш точному відбору новин відповідно до інтересів користувача. Надійність та стабільність. Filtered News бот працює досить стабільно, що забезпечує надійність його роботи. Бот також має хорошу підтримку та регулярно оновлюється, що дозволяє користувачам отримувати свіжі та актуальні новини.

Мінус: Обмеженість джерел новин. Хоча Filtered News бот підтримує багато джерел новин, він все ще обмежений тими, які підтримуються ботом. Користувачам може бути складно знайти новини зі своїх улюблених джерел, які не підтримуються ботом. Filtered News бот може бути обмеженим у виборі новин, оскільки він автоматично фільтрує новини за заданими категоріями. Це

може бути незручним для користувачів, які шукають новини за конкретними ключовими словами чи темами,

Загалом, телеграм-бот "Filtered News" є корисним інструментом для отримання новин, але він має свої обмеження та помилки. Потрібно продовжувати працювати над розвитком бота та виправляти помилки, щоб забезпечити якісний та стабільний досвід для користувачів.

RemindMegaBot є телеграм-ботом, який дозволяє користувачам створювати нагадування та отримувати їх у вказаний час. Ось деякі позитивні аспекти цього бота:

Простота використання: Бот має простий та зрозумілий інтерфейс, що дозволяє користувачам створювати нагадування всього за кілька кроків.

Гнучкість: RemindMegaBot дозволяє користувачам налаштовувати нагадування на будь-який час, що робить його дуже гнучким і придатним для будь-якого виду застосування.

Надійність: Бот надійно працює і не має відомих проблем або помилок.

Однак, є деякі недоліки, які можуть зменшити зручність використання RemindMegaBot: Бот не підтримує багатокористувацький режим.

Вирішення: Щоб зробити бота більш зручним для користувачів, можна додати можливість створювати різні профілі або облікові записи для різних користувачів. Кожен профіль міг би містити свій список нагадувань та інші налаштування.

Користувачі можуть отримувати занадто багато повідомлень від бота.

Вирішення: Можна додати налаштування для кожного нагадування, щоб користувачі могли встановлювати пріоритети та частоту отримання повідомлень. Також можна додати можливість створювати пакети нагадувань, щоб користувачі могли об'єднувати схожі нагадування в одному пакеті.

Бот не підтримує групові повідомлення.

Вирішення: Для розширення функціональності бота можна додати можливість створювати групи та робити групові нагадування. Користувачі зможуть приєднуватися до груп та створювати спільні нагадування для всіх учасників групи.

Підсумовуючи, RemindMegaBot є корисним та потужним телеграм-ботом, який допомагає користувачам організовувати свій час та пам'ятати про важливі справи. Він має багато позитивних аспектів, таких як зручний інтерфейс, можливість додавати різні типи нагадувань, і можливість налаштувати повторювані нагадування. Однак, він має деякі недоліки, наприклад, не завжди працює стабільно та може потребувати певних вдосконалень щодо налаштування нагадувань.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		21

2 ПРОЕКТУВАННЯ СИСТЕМИ

2.1 Формалізація вимог до системи.

Для створення телеграм бота, який буде отримувати результати змагань з ловлі хижої риби та відповідати на запити користувачів щодо цієї інформації, необхідно врахувати ряд вимог, які забезпечать ефективну та надійну роботу бота.

Перш за все, бот повинен мати зручний та інтуїтивно зрозумілий інтерфейс для користувачів, який дозволяє зручно здійснювати запити та отримувати інформацію про результати змагань. Інформація повинна бути представлена у зрозумілому та логічному форматі.

Друга вимога полягає у надійності та стабільності бота. Бот повинен функціонувати без збоїв та перерв, а також забезпечувати швидкий доступ користувачів до необхідної інформації. Для цього можна використовувати відповідні технології та сервіси, які дозволяють забезпечити надійну роботу бота та швидкий відгук на запити користувачів.

Третя вимога полягає у безпеці та конфіденційності даних. Бот повинен забезпечувати безпечний обмін даними між користувачами та забезпечувати конфіденційність особистих даних користувачів. Для цього можна використовувати шифрування та інші технології, які дозволяють забезпечити безпеку даних.

2.2 Проектування систем.

Боти – спеціальні акаунти в Telegram, створені для автоматичного оброблення та надсилання повідомлень. 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.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		22

Ось кілька прикладів використання роботів:

- Інтеграція з іншими сервісами. Наприклад, бот може надсилати коментарі або керувати «розумним будинком». Або, наприклад, надсилати вам повідомлення при скоєнні якоїсь дії або події (Приклади: GitHub Bot, Image Bot).
- Утиліти та інструменти. Бот може відображати погоду, перекладати тексти або попереджати про майбутні події на ваш запит (Наприклад: бот опитувань).
- Однокористувацькі і багатокористувацькі ігри. Бот може погратися з вами в шашки або шахи, проводити вікторини і таке інше. (Приклад: Trivia bot).
- Соціальні послуги. Бот може знаходити вам співрозмовника, ґрунтуючись на ваших спільних інтересах та захопленнях. (Приклад: HotOrBot).

Як уже було сказано раніше, роботи - спеціальні акаунти, які не вимагають номери телефону при створенні. По суті, ці облікові записи відіграють роль інтерфейсу до вашого сервісу, який працює на віддаленому сервері.

Найцікавіше в роботах це те, що для їх створення вам не потрібно вивчати низькорівневі методи роботи з MTProto та шифруванням – спілкування з роботом організовано за допомогою звичайного інтерфейсу HTTPS зі спрощеними методами Telegram API.

Насправді можна створити бота в телеграм за допомогою бота. Для цього потрібно написати користувачеві @BotFather , а потім слідувати його інструкціям. Після створення буде отримано спеціальний ключ авторизації (токен). Виконати необхідні налаштування можна в розділі документації Bot API.

Після простої реєстрації боту, отримано токен для роботи з ботом з Java-коду (рис. 2.1).

					ДП.КН 23.519.03.000 ПЗ	Анк.
						23
Зм.	Арк.	№ докум.	Підпис.	Дата		

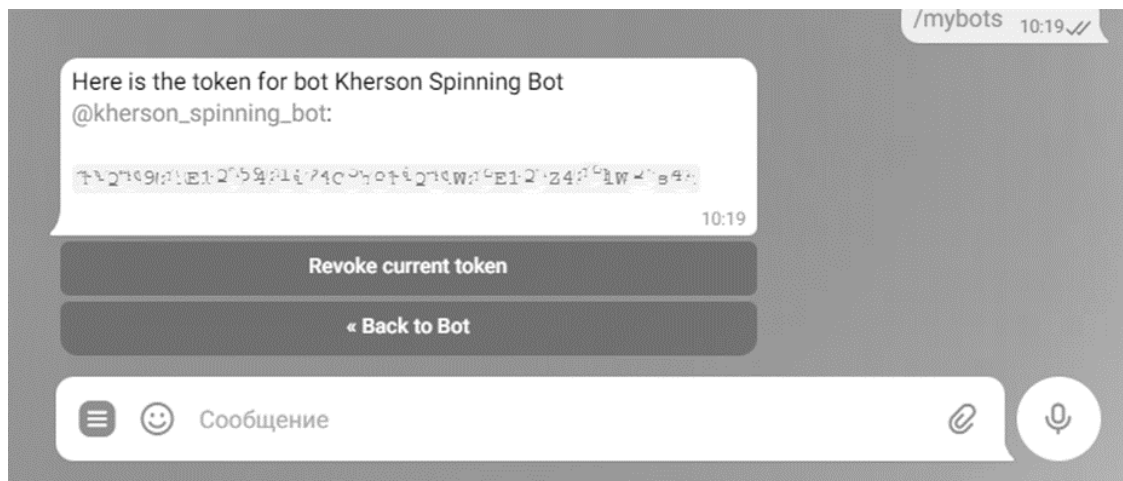


Рисунок 2.1 – Ім'я боту та реєстраційний токен

Токен – це спеціальний ключ від робота, 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. Токен – інформація про бот, яку потрібно тримати засекреченою, щоб його не змогли використовувати зловмисники.

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.

Для початку роботи над проектом потрібно створити каркас проекту за допомогою Spring Initializr.

Щоб щоразу не створювати з нуля Spring проект на Java і не шукати останніх версій залежностей, можна скористатися сервісом Spring Initializr, який надає інтерфейс для генерації заготовлі проекту з додаванням стандартних залежностей (рис 2.2). Їх можна конфігурувати в залежності від потреб. Як збирач проекту виберемо Gradle, і Spring Initializr автоматично згенерує скрипт збирання. Потім завантажуюмо цю заготовку і запускаємо її в IntelliJ Idea від JetBrains.

Project
☐ Maven Project
 ☒ Gradle Project

Language
☒ Java
 ☐ Kotlin
 ☐ Groovy

Spring Boot
☐ 3.0.0 (SNAPSHOT)
 ☐ 3.0.0 (M2)
 ☐ 2.7.0 (SNAPSHOT)
 ☐ 2.7.0 (RC1)
 ☐ 2.6.8 (SNAPSHOT)
 ☒ 2.6.7
 ☐ 2.5.14 (SNAPSHOT)
 ☐ 2.5.13

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 18 ☐ 17 ☒ 11 ☐ 8

Рисунок 2.2 – Інтерфейс Spring Initializr та налаштування проекту.

Після відкриття проекту в IDE потрібно заповнити файл `application.properties` даними про бота. З цього файлу будуть підтягуватись потрібні дані для роботи.

Spring Boot надає спосіб екстерналізації конфігурацій з використанням файлів властивостей, які перебувають за адресою до класів проекту. Це дуже корисно для налаштування кількох середовищ, тому властивості можна динамічно налаштовувати для кожного середовища.

Файл властивостей за промовчанням, який генерує Spring Boot, називається `application.properties` і знаходиться в каталозі `src/main/resources/`. При ініціалізації проекту файл властивостей за замовчуванням залишається порожнім.

Файл `application.properties` у нашому випадку містить два приклади властивостей:

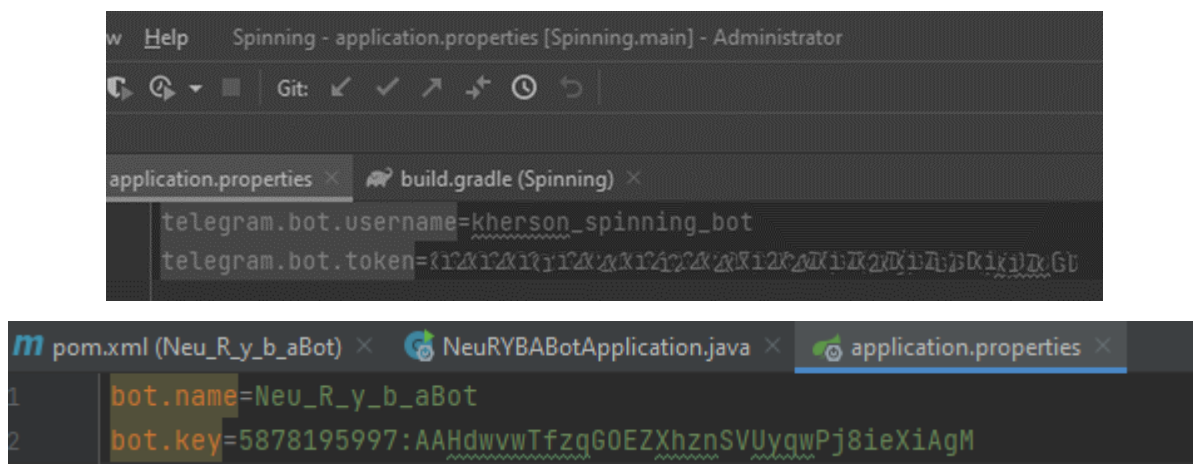


Рисунок 2.3 – Файл application.properties з налаштуваннями бота

Цей файл містить значення для імені бота та згенерований токен. Властивості слідують однаковому синтаксису key=value (ключ=значення).

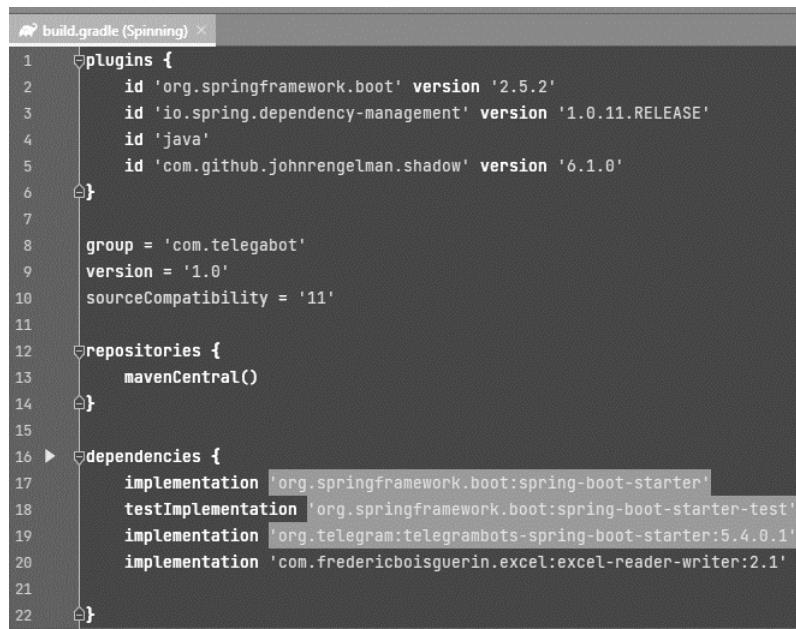
Оскільки бот буде вибирати дані з таблиці у форматі MS Excel, потрібно підключити залежність Apache POI для роботи з електронними таблицями, а також підключити API для роботи з телеграм. Ці маніпуляції проводитимуться у файлі build.gradle, що визначає конфігурацію збирання:

implementation 'org.telegram:telegrambots-spring-boot-starter:5.4.0.1'

implementation 'com.fredericboisguerin.excel:excel-reader-writer:2.1'

Gradle – це система автоматизації складання з відкритим вихідним кодом, яка дозволяє вам вникати у процес складання та її логіку.

Після налаштувань файл build.gradle матиме наступний вигляд:



```

1  plugins {
2      id 'org.springframework.boot' version '2.5.2'
3      id 'io.spring.dependency-management' version '1.0.11.RELEASE'
4      id 'java'
5      id 'com.github.johnrengelman.shadow' version '6.1.0'
6  }
7
8  group = 'com.telegabot'
9  version = '1.0'
10 sourceCompatibility = '11'
11
12 repositories {
13     mavenCentral()
14 }
15
16 dependencies {
17     implementation 'org.springframework.boot:spring-boot-starter'
18     testImplementation 'org.springframework.boot:spring-boot-starter-test'
19     implementation 'org.telegram:telegrambots-spring-boot-starter:5.4.0.1'
20     implementation 'com.fredericboisguerin.excel:excel-reader-writer:2.1'
21 }
22

```

Рисунок 2.4 – Налаштування та залежності файлу build.gradle.

Такими маніпуляціями проект підготовлено до розробки. Наступним кроком буде створення класів програми. Класи – це основа основ програмування Java. Клас — це, насправді, шаблон для об'єкта. Він визначає, як об'єкт буде виглядати і які функції мати. Кожен об'єкт є об'єктом якогось класу.

Існує два способи, як бот може отримувати повідомлення з серверів Telegram. Вони називаються *longpolling* (довгим опитуванням) і *webhooks* (вебхуками).

При роботі через вебхуки про всі зміни в чатах, каналах та групах, в яких бере участь наш бот, йому відразу повідомляє Telegram, відправляючи в http-запитах структури даних, звані апдейтами (Updates). Повна структура апдейтів (об'єкти, поля, типи даних та все інше) є на сторінці з описом API для створення роботів.



Рисунок 2.5. – Схема роботи webhook ботів.

Webhook – зручна і швидка схема, але його використання вимагає деяких додаткових маніпуляцій:

По-перше, потрібно вказати Телеграму, де розташована програма-оператор. Щоб Телеграм міг знайти бота, використовуючи систему DNS, нам потрібно мати місце на хостингу та зареєстрований інтернет-домен. Простіше кажучи, щоб сказати, де шукати – потрібно отримати в мережі якусь постійну прописку. Далі розташовуємо проект на сайт і повідомляємо месенджеру про те, де він розташований. Це називається «встановити webhook».

По-друге, Telegram згоден надсилати запити виключно за захищеним протоколом https. Тобто потрібна підтримка хостингом цього протоколу та SSL-сертифікат. Але багато хостинги безкоштовно видають сертифікат SSL початкового рівня і підтримують https.

У випадку з методом longpolling бот повинен сам періодично вимагати апдейти у Telegram, використовуючи метод getUpdates. Опитування має відбуватися не рідше 1 разу на добу, оскільки довше Telegram не згоден зберігати апдейти. Крім того, скористатися методом getUpdates можна тільки в тому випадку, якщо для бота не встановлено webhook.

У випадку з longpolling треба вибрати компроміс між оперативністю отримання даних та ресурсами, які витрачаються на опитування серверів Telegram.

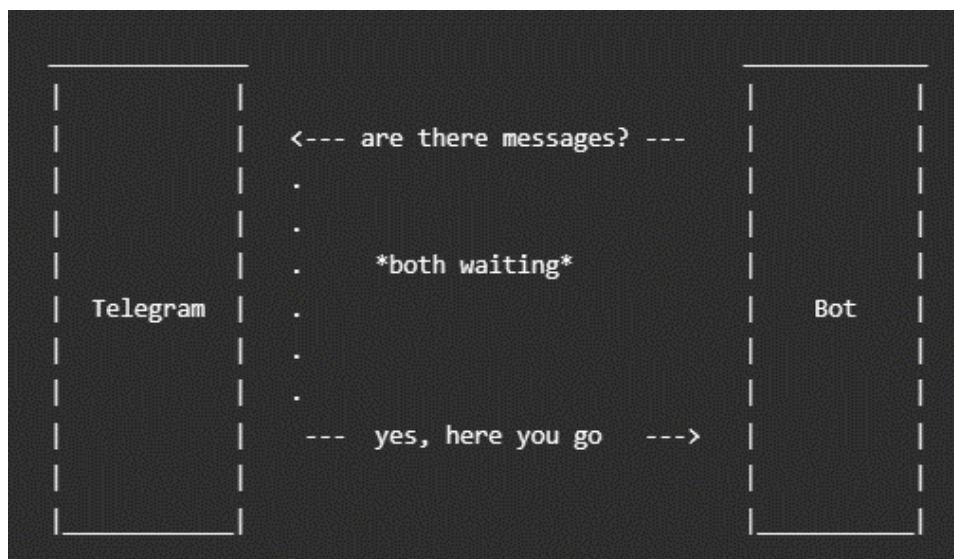


Рисунок 2.6. – Схема роботи longpolling ботів.

Якщо опитувати дуже рідко – буде занадто великий проміжок між моментом, коли хтось посилає роботу дані, і моментом, коли бот ці дані отримує. В результаті бот буде дуже загальмовано реагувати на повідомлення, що йому посилаються.

Занадто часто опитування може викликати занадто велике навантаження на ваш сервер, оскільки в цьому випадку доводиться ходити по мережі і обробляти дуже багато марної інформації (формувати та посилати на сервери Telegram запити, відповідями на які будуть повідомлення, що нічого не надіслано).

Єдиний, зате найчастіше вирішальний плюс методу longpolling полягає в тому, що не потрібно мати в мережі ніякої постійної прописки, оскільки не просимо нічого нікуди надсилати і не потрібно залишати жодної зворотної адреси. Тобто в цьому випадку робота можна запустити будь-де, хоч на своєму домашньому комп'ютері.

Опираючись на вищеописане, та структуру завдання, можна зробити висновок, що, оскільки робота боту потрібна на час змагань та деякий час після змагань, бота можна запускати на локальній машині з доступом до інтернету. Для цього можна використовувати домашній комп'ютер.

2.3 Проєктування структур бази даних.

Часто для створення звітів у форматі файлів Microsoft Excel потрібна програмна програма. Іноді навіть очікується, що програма отримає файли Excel як вхідні дані. Наприклад, програма, розроблена для фінансового відділу компанії, повинна буде генерувати всі свої результати в Excel.

Будь-який програміст на Java, який бажає створювати файли MS Office як вихідні дані, повинен використовувати для цього зумовлений API тільки для читання.

Apache POI – це популярний API, який дозволяє програмістам створювати, змінювати та відображати файли MS Office за допомогою програм Java. Це бібліотека з відкритим вихідним кодом, розроблена та розповсюджувана Apache Software Foundation для розробки або зміни файлів Microsoft Office за допомогою програми Java. Він містить класи та методи для декодування даних, що вводяться користувачем, або файлу в документи MS Office.

Компоненти Apache POI наведені у таблиці 2.2

Таблиця 2.2. Компоненти Apache POI.

<i>Імя класу</i>	<i>Розшифровка аббревіатури</i>	<i>Для чого використовується</i>
HSSF	<i>Horrible Spreadsheet Format</i>	Компонент читання та запису файлів MS-Excel, формат XLS
XSSF	<i>XML Spreadsheet Format</i>	Компонент читання та запису файлів MS-Excel, формат XLSX

Продовження таблиці 2.1

HPSF	<i>Horrible Property Set Format</i>	Компонент отримання наборів властивостей файлів MS-Office
HWPF	<i>Horrible Word Processor Format</i>	Компонент читання та запису файлів MS-Word, формат DOC
XWPF	<i>XML Word Processor Format</i>	Компонент читання та запису файлів MS-Word, формат DOCX
HSLF	<i>Horrible Slide Layout Format</i>	Компонент читання та запису файлів PowerPoint, формат PPT
XSLF	<i>XML Slide Layout Format</i>	Компонент читання та запису файлів PowerPoint, формат PPTX
HDGF	<i>Horrible DiaGram Format</i>	Компонент роботи із файлами MS-Visio, формат VSD
XDGF	<i>XML DiaGram Format</i>	Компонент роботи із файлами MS-Visio, формат VSDX

Для зчитування даних з електронної таблиці, таблиці приведено до загального вигляду так, щоб дані, які потрібні боту, знаходилися на п'ятому рядку таблиці. Приклад файлу жеребкування на (рис.2.7)

	A	B	C	D	E	F
1		XIV Відкритий Кубок м. Херсон				
2		р. Чайка				
3						
4		ЖЕРЕБКУВАННЯ				
5	№	Учасник	Команда	ТУР 1	ТУР 2	ТУР 3
6	1	Ніколаєнко Дмитро	Істок	A	B	C
7	2	Маліцький Віктор	Істок	B	C	A
8	3	Саригін Олексій	Істок	C	A	B
9	4	Лаптев Дмитро	Південна спепова брс	A	C	B
10	5	Лось Ярослав	Південна спепова брс	B	A	C
11	6	Гречановський Олекс	Південна спепова брс	C	B	A
12	7	Негруца Роман	Грілекс-22-регіон	C	B	A
13	8	Мельниченко Дмитро	Грілекс-22-регіон	B	A	C
14	9	Татаренко Богдан	Грілекс-22-регіон	A	C	B

Рисунок 2.7 – Приклад файлу жеребкування

Вхідні дані для бота будуть знаходитися у шістьох файлах формату .xlsx. Тому для їх обробки потрібно створити клас ReadingExcel, який матиме чотири методи для обробки файлів.

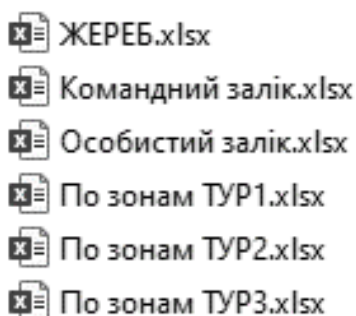


Рисунок 2.8 – Файли.xlsx.

Кожен метод оброблятиме свій вид файлу, оскільки дані таблиці та її форматування різняться між собою. Метод для обробки файлів кожного туру по зонам може бути один для трьох файлів.

Метод readZoneFromXlsx оброблятиме файли турів по зонам та буде повертати список спортсменів та їх результат для подальшого відображення в інтерфейсі бота.

Метод readTossFromXlsx оброблятиме файл жеребкування спортсменів. Його вихідними даними також буде список спортсменів із відповідними жеребкуванням.

Метод readPersonalFromXlsx оброблятиме дані по особистому заліку спортсменів у змаганнях. Адже змагання діляться на особистий та загальнокомандний залік. У кожній команді має бути три спортсмени.

Метод readTeamFromXlsx оброблятиме загальнокомандні дані.

Код класу знаходиться у додатках до проекту.

Оскільки функціонал боту невизначено до кінця, а команду /help потрібно перевизначити, створено клас, який буде вчитувати жарти з файлу формату .txt та у випадковому порядку буде видавати їх користувачеві. Також цей клас спрацьовуватиме тоді, коли користувач введе невідому боту команду.

Файл із жартами та анекдотами можна оновлювати та доповнювати але він має також специфічне форматування для того, щоб розумів кінець одного жарту та початок іншого. Такий підхід створить тимчасову заглушку для боту та підніме настрій спортсменам.

3 РЕАЛІЗАЦІЯ ТЕСТУВАННЯ СИСТЕМИ

1.3 Огляд засобів організації

Майже кожен, навіть якщо ніколи не цікавився програмуванням, швидше за все, чув назву Java. Це одна з найпопулярніших у світі мов програмування: вона була створена в 1995 році, швидко набрала популярності і вже багато років її не втрачає. У рейтингу TIOBE за листопад 2019 Java зайняла перше місце, а за статистикою GitHub – третє: друге місце зайняв Python, а перше – JavaScript.

Java використовують скрізь: ви знайдете її майже у всіх великих компаніях, у тому числі в Netflix, AliExpress, Google, Intel, eBay, TripAdvisor та багатьох інших.

Java – об'єктно-орієнтована мова програмування, що розробляється компанією Sun Microsystems і офіційно випущена 23 травня 1995 року. Щодо вимови російською, як й інших, утворилися дві різні норми — запозичена англomовна і традиційно-національна «Ява», що відповідає традиційному вимові назви острова Ява.

Java – так називають не тільки саму мову, а й платформу для створення додатків рівня підприємств на основі цієї мови.

Спочатку мова програмування називалася Оак (Дуб) і розроблявся Джеймсом Гослінгом для побутової електроніки, але згодом був перейменований в Java і став використовуватися для написання клієнтських програм та серверного програмного забезпечення. Названий на честь марки кави Java, улюбленої програмістами, тому на офіційній емблемі мови Java зображена чашка з кавою, що димиться.

Програми на Java транслюються в байт-код, що виконується віртуальною java-машиною (JVM) – програмою, що обробляє байтовий код і передає інструкції обладнання як інтерпретатор, але з тією відмінністю, що байтовий код, на відміну від тексту, обробляється значно швидше.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		34

Одним з основних переваг такого методу виконання програм є абсолютна незалежність байт-коду від операційної системи та апаратного забезпечення, що дозволяє запускати Java-програми на будь-якому пристрої, який підтримує віртуальну машину. Іншим важливим аспектом технології Java є гнучка система безпеки, завдяки якій виконання програми повністю контролюється віртуальною машиною. Будь-які дії, які перевищують повноваження програми (наприклад, незаконний доступ до даних або з'єднання з іншим комп'ютером), спричиняють негайне припинення виконання. Це дає змогу користувачам завантажувати програми, написані на мові Java, на їх комп'ютери (або інші пристрої, такі як мобільні телефони) з невідомих джерел, не хвилюючись про можливість зараження вірусами або втрату важливої інформації.

Часто вказують на недоліки використання віртуальної машини, які полягають у можливому зниженні продуктивності програм та алгоритмів, реалізованих на мові Java, через виконання байт-коду віртуальною машиною. Це твердження було вірним для ранніх версій віртуальної машини Java, проте воно втратило актуальність у останні часи. Цьому сприяли різні покращення, такі як використання native-коду в стандартних бібліотеках та розширені апаратні можливості, що дозволяють прискорити обробку байт-коду.

Ідеї, закладені в концепцію та різні реалізації JVM (Java Virtual Machine), надихнули безліч ентузіастів на розширення переліку мов, які могли б бути використані для створення програм, що виконуються всередині JVM. Ця ідея перегукується із концепцією CLI (Command Line Interface), закладеною в основу платформи .NET.

Усередині Java існують три основні сімейства технологій:

- J2EE або Java EE (починаючи з v1.5) – Java Enterprise Edition, для створення програмного забезпечення рівня підприємства;
- J2SE або Java SE (починаючи з v1.5) – Java Standard Edition, для створення власних додатків, в першу чергу – для настільних систем;

- J2ME, Java ME або Java Micro Edition, для використання у пристроях, обмежених за обчислювальною потужністю, у тому числі мобільних телефонах, PDA, вбудованих системах.

Найпопулярнішими вважаються серверні технології сімейства J2EE.

Останнім релізом є версія 17, в якій було здійснено покращення системи безпеки, покращення підтримки скриптової мови Mozilla Rhino, покращено інтеграцію з робочим столом, додано деякі нові можливості у створенні графічних інтерфейсів.

Java та Microsoft. Компанією Microsoft, починаючи з Windows 95. MSJVM була близька до специфікації Sun Microsystems v.1.1.3, але не відповідала їй за низкою важливих параметрів (зокрема, за відсутністю підтримки технологій JNI та за наявності нестандартних розширень). Згодом це стало приводом для судових позовів із боку Sun Microsystems. В даний час компанією .NET пропонується Java-подібна мова J#, яка не відповідає офіційній специфікації мови Java.

Сфери застосування мови Java. Java застосовується в широкому спектрі областей, включаючи електронну комерцію, веб-сайти, розробку програм для Android, наукові дослідження, фінанси, ігри (наприклад, Minecraft) та розробку настільних програм, таких як Eclipse, Netbeans і IntelliJ.

Мобільні програми Android. Java Micro Edition – це популярне кросплатформове середовище, що застосовується для створення програм смартфонів. Приклади мобільних програм, написаних на Java: Netflix, Tinder, Google Earth, Uber і т.д.

Ігри на Java. Java-Gaming.org є веб-сайтом, який повністю присвячений розробці ігор на мові Java. Процес створення ігор значно полегшується завдяки використанню бібліотеки LWJGL. Щоб переглянути приклади ігор, написаних на мові Java, можна здійснити пошук в Інтернеті.

Серверні фінансові програми. Глобальні інвестиційні банки, такі як Goldman Sachs, Citigroup, Barclays і Standard Chartered, використовують мову програмування Java для розробки фронт- та бек-офісних систем у сфері електронної торгівлі, систем розрахунків та підтвердження транзакцій, обробки даних та інших схожих систем.

Веб-програми на Java. З наведеної нижче таблиці (рис. 3.1) випливає, що Java - найпопулярніша серверна мова серед провідних технологічних компаній. У порівнянні з іншими мовами програмування, Java має кращу безпеку, адаптованість і масштабованість додатків.

Back-end (Server-side) table in most popular websites																
Websites ⇅	C# ⇅	C ⇅	C++ ⇅	D ⇅	Erlang ⇅	Go ⇅	Hack ⇅	Java ⇅	JavaScript ⇅	Perl ⇅	PHP ⇅	Python ⇅	Ruby ⇅	Scala ⇅	Xhp ⇅	
Google	No	Yes	Yes	No	No	Yes	No	Yes	No	No	No	Yes	No	No	No	
YouTube	No	Yes	Yes	No	No	Yes	No	Yes	No	No	No	Yes	No	No	No	
Facebook	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes	No	No	Yes	
Yahoo	No	Yes	Yes	No	No	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	
Amazon	No	No	Yes	No	No	No	No	Yes	No	Yes	No	No	No	No	No	
Wikipedia	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No	
Twitter	No	No	Yes	No	No	No	No	Yes	No	No	No	No	Yes	Yes	No	
Bing	Yes	No	Yes	No	No	No	No	No	No	No	No	No	No	No	No	
eBay	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes	No	
MSN	Yes	No	No	No	No	No	No	No	No	No	No	No	No	No	No	
LinkedIn	No	No	No	No	No	No	No	Yes	Yes	No	No	No	No	Yes	No	
Pinterest	No	No	No	No	Yes	No	No	No	No	No	No	Yes	No	No	No	
WordPress.com	No	No	No	No	No	No	No	No	No	No	Yes	No	No	No	No	

Рисунок 3.1. – Порівняння мов програмування

Серед компаній, які переважно спеціалізуються на Java (J2EE) технологіях, а не на .NET, можна відзначити IBM і Oracle. Наприклад, у СУБД Oracle використовується JVM як одна з її складових, що дозволяє пряме програмування бази даних мовою Java, включаючи процедури, що зберігаються.

Основні можливості Java:

- автоматичне керування пам'яттю
- розширені можливості обробки виняткових ситуацій;
- багатий набір засобів фільтрації вводу/виводу;

- набір стандартних колекцій, таких як масив, список, стек тощо;
- наявність простих засобів створення мережових програм (у тому числі з використанням протоколу HTTP-запити та обробляти відповіді);
- вбудовані в мову засоби створення багатопотокових додатків;
- уніфікований доступ до баз даних на основі JDBC (англ. Java DataBase Connectivity - з'єднання з базами даних на Java) - платформно незалежний промисловий стандарт взаємодії Java-додатків з різними СУБД, реалізований у вигляді пакета java.sql, що входить до складу Java SE.

Концепція JDBC базується на використанні драйверів, що дозволяють отримувати з'єднання з базою даних шляхом використання спеціально визначеного URL. Драйвери можуть бути динамічно завантажені під час роботи програми. Після завантаження, драйвер самостійно реєструє себе і автоматично викликається, коли програма потребує URL-адреси з певним протоколом, за який драйвер відповідає.

Може здатися, що Java - абсолютний лідер і вона не має суперників, але насправді все зовсім навпаки. Її часто порівнюють із C# та Python, і це лише головні конкуренти.

Головна перевага Python – його простий синтаксис. Справді, написати простий скрипт або випробувати нову ідею серед Python куди простіше. Не потрібно писати і компілювати цілу програму, щоб досягти результату.

Це проявляється під час роботи з будь-якими сутностями коду. Ті ж класи в Python та Java виглядають по-різному. В останньому вони помітно масивніші і складніші в розумінні, ніж ті, що пропонує Python.

На відміну від Python, Java використовується сувора типізація, що змушує розробників бути дисциплінованими і завжди заздалегідь декларувати тип даних, що використовується. Інакше компілятор відкликається помилкою. Python у цьому плані менш прискіпливий.

А ще Python підтримує більше варіантів оформлення коду, тому більшість розробників вважає, що його легше читати.

Незважаючи на наявність спільних рис між мовами в частині синтаксису, Java та C++ мають ряд значних відмінностей.

Наприклад, C++ використовує лише компілятор. Тобто механізм, що перетворює весь код в об'єктну структуру, що безпосередньо прочитується комп'ютером. Java ж, крім компілятора, задіює інтерпретатор, який читає кожен рядок і відразу виконує інструкції, описані в ній.

C++ підтримує перезавантаження операторів і перевантаження методів, і навіть такі типи, як `struct` і `union`. З перерахованого вище Java підтримує тільки перевантаження методів.

C++ частіше використовується для створення музичного програмного забезпечення, наприклад, секвенсорів або емуляторів аналогового обладнання. Також C++ код можна виявити у компонентах операційних систем Windows та macOS.

Багато розробників порівнюють C++ і Java на користь другого, тому що він має схожі можливості, але не містить недоліків першого.

У порівнянні з C і C++ вона, може, і не найшвидша, але Python явно відстає від Java за швидкістю та продуктивністю. В обох мовах програми переводяться в байт-код (це дозволяє їм бути кросплатформенними), але різниця полягає в тому, коли це відбувається: Java компілює заздалегідь, за допомогою JIT-компіляції (динамічної компіляції), а Python - під час виконання програми. У результаті Java значно швидше.

IntelliJ IDEA - це потужне і популярне середовище розробки, яке в першу чергу орієнтоване на роботу з Java, принаймні саме так його позиціонують. Розроблена компанією JetBrains.

IntelliJ IDEA чудово знайома багатьом розробникам Java. Вона входить у трійку найвідоміших середовищ розробки поруч із Eclipse і NetBeans. (рис 3.2.)

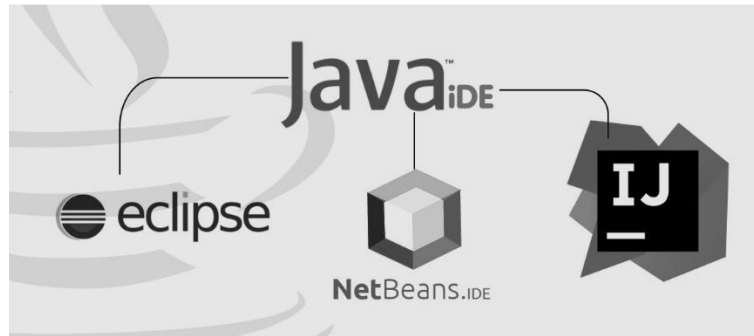


Рисунок 3.2 – Дочірні продукти Java

Спочатку це середовище розробки дійсно було розраховане для роботи з Java. Але на сьогоднішній день навіть у безкоштовній версії цього продукту можна програмувати кількома популярними мовами та фреймворками. Крім цього, всередині цього інструменту вбудовано безліч додаткових функцій, наприклад, інтеграція із системами контролю версій, а також інструменти збирання та розгортання додатків. А в останніх версіях цієї IDE було додано функцію розробки інтерфейсів користувачів, що ще більше розширило можливості цього інструменту.

На сьогоднішній день є дві версії цього продукту:

1. IntelliJ IDEA Community Edition. Безкоштовна версія, яка поширюється на вільну ліцензію. Це ідеальний варіант для програмістів-початківців і тих, хто спеціалізується на Android-розробці та JVM.
2. IntelliJ IDEA Ultimate Edition. Платна версія продукту, в якій можна знайти більше інструменту, ніж у безкоштовній версії. Її можливості вже дозволяють займатися enterprise-розробкою або серйозною веб-розробкою, включаючи фронтенд та бекенд.

IntelliJ IDEA – це робота у трьох найпопулярніших операційних системах: Windows, Linux, MacOS. Найсвіжішу версію цієї IDE для будь-якої з трьох платформ можна завантажити на офіційному сайті.

За замовчуванням інтерфейс IntelliJ IDEA спроектований таким чином, щоб перед очима розробника не було нічого зайвого, крім редактора коду, тому

останній займає максимальну частину інтерфейсу IDE. А додаткові кнопки для роботи з іншими інструментами IntelliJ IDEA розташовані по краях робочого екрану, причому з обох боків та зверху-знизу.

Для зручності користування в IntelliJ IDEA додана крута можливість - програмувати власне поєднання клавіш для швидкого доступу до можливостей самого інструменту або для дій, які часто доводиться виконувати розробнику.

IntelliJ IDEA – це багато дрібних гнучких налаштувань візуальної частини під власні вимоги, тому працювати на ній – одне задоволення. Крім кастомізації зовнішнього, всередині IntelliJ IDEA зібрано багато корисних інструментів та функцій. Ось деякі з них:

- 1) Дебаггер IntelliJ IDEA. Дуже зручно його застосовувати, оскільки він показує свої значення у коді.
- 2) Live Templates. Дозволяє розробнику створювати власні шаблони із конструкцій коду, які він часто використовує. Створивши шаблон, він може застосовувати його будь-де у своїй розробці.
- 3) Першокласний пошук за документом. IntelliJ IDEA завантажує весь документ для аналізу, тому програміст може легко знайти потрібні йому класи, методи та файли. Причому пошук може здійснюватися за ім'ям, директорією, модулем, по всьому проекту або його області.
- 4) Smart Completion. Пропонує розробнику найкраще завершення рядка, що значно прискорює розробку.
- 5) Chain Completion. Пропонує розробнику найбільш підходящі методи класів та змінних у кожному конкретному випадку.
- 6) Інтуїтивний імпорт — імпортує лише потрібні пакети, а все, що не потрібне, моментально видаляє.
- 7) Потужний інструмент рефакторингу.
- 8) Інструменти збирання: Maven, Gradle, Webpack та ін.
- 9) Контроль версій: Git, Subversion, CVS та ін.

- 10) Декомпілятор, якщо потрібно переглянути, що міститься в бібліотеці, що використовується.
- 11) Термінал підійде для тих, хто вміє користуватися Bash.
- 12) Інструменти для роботи з базами даних.

Якщо не вистачає вбудованих інструментів для роботи, ви завжди можете розширити функціонал IntelliJ IDEA за допомогою додаткових плагінів, яких у цього середовища безліч.

Все вищеописане можна віднести до переваг IntelliJ IDEA, то слід зазначити, що дана IDE має один істотний недолік. Вона досить «стара», тому що розробляється ще з 2001 року. Тому в неї досить великий обсяг вихідного коду, що зрештою призводить до наявності багів.

Другий момент: IntelliJ IDEA – це інструмент, що забирає під свої потреби дуже багато апаратних можливостей, отже, щоб працювати за допомогою цієї IDE, у вас має бути достатньо потужний пристрій.

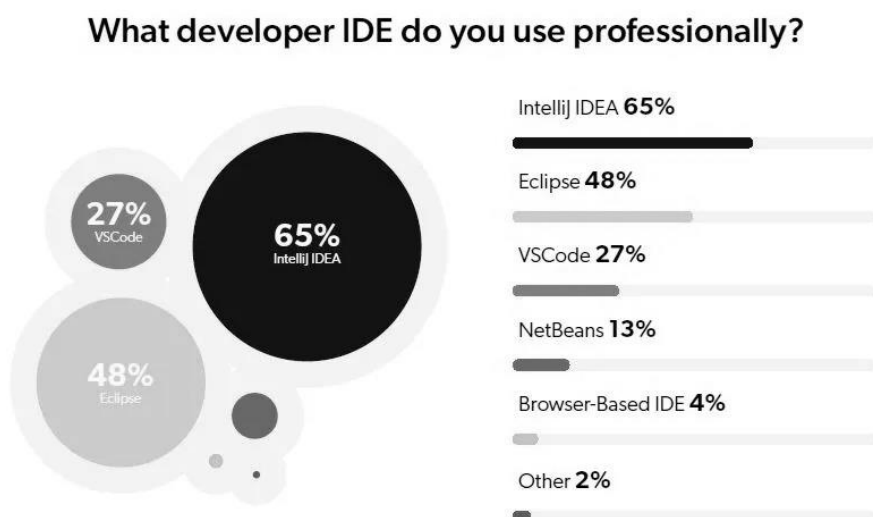


Рисунок 3.3 – Статистика використання IntelliJ IDEA

Не менш важливим інструментом є графічний редактор для створення інтерфейсу. Це значно зручніше, ніж писати шаблонний код. Розробник може лише перетягувати мишкою елементи на робоче поле, тоді як код генерується

самостійно. Серед досвідчених програмістів такий підхід можна вважати дилетантським. Але правда в тому, що таким чином економиться багато часу, а якість коду не стає гіршою. Через це візуальні редактори використовують не лише новачки, а й ті, хто просто хоче оптимізувати свою роботу.

Тому IntelliJ IDEA є однією із найпопулярніших IDE (Рис.3.3).

З'явившись на ринку, API справив справжній прорив і назавжди змінив процес синхронізації інформації між різним програмним забезпеченням. Розглянемо, що являє собою інтерфейс програмування API, область його застосування у сфері написання додатків, приклади його використання та можливі варіанти застосування API у різних розробках.

API (Application Programming Interface) представляє собою набір різноманітних інструментів та функцій, реалізованих у вигляді інтерфейсу, який дозволяє створювати нові програми та забезпечує взаємодію між ними.

Основною метою створення API було надання програмістам зручного способу спрощення процесу розробки різних додатків шляхом використання вже наявного коду (стандартних функцій, процедур, структур або постійних значень), які можуть бути використані у кінцевому продукті.

API визначає можливу функціональність, яку певна програма у формі бібліотеки або модуля зможе здійснювати, при цьому API дозволяє абстрагуватись від способу реалізації функціоналу.

Різні компоненти програми завдяки API отримують можливість взаємодії один з одним, формуючи при цьому, як правило, серед компонентів певну систему ранжування, де абсолютно всі компоненти, які вище за рангом використовують інформацію з нижчих за рангом компонентів, які також використовують інформацію з рангів нижче (рис. 3.4.).

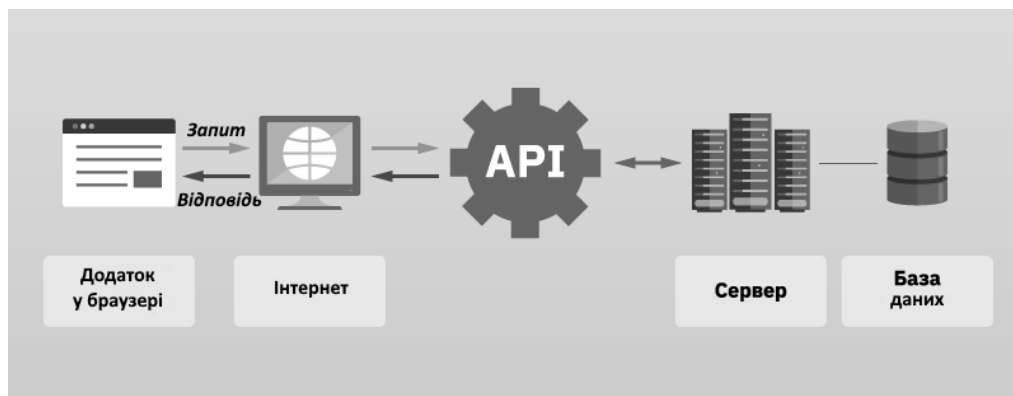


Рисунок 3.4. – Взаємодія компонентів

Слід зазначити, що саме за схожим алгоритмом сформовані протоколи передачі даних через мережу Інтернет, де будь-який рівень використовує функціонал нижчестоящого рівня передачі даних, тим самим надаючи необхідну інформацію та функціональні можливості вищому. Інтернет протоколи схожі за змістом з поняттям API, будучи абстракцією функціональності, лише в одному з випадків суть полягає у передачі даних, а в другому про взаємодію додатків між собою.

Хорошим прикладом API може бути Twitter, інтерфейс за допомогою якого здатний надати Вам інформацію, що відноситься до твітів користувача, кого він читає, хто є його передплатниками, що є невеликою частиною реальних можливостей, які будь-хто зможе здійснити, використовуючи як власний, так і будь-який сторонній. API.

Прикладом використання API може служити процес швидкої реєстрації в різних додатках використовуючи обліковий запис будь-якої із запропонованої соціальної мережі, коли за допомогою спеціальної API соціальної мережі (наприклад, Вконтакте, Facebook) сторонні компанії отримують можливість використовувати спеціальний код та API для надання Вам оперативного та спрощеного доступу до їхнього продукту.

Компанія Google за рахунок використання API дає потенційну можливість розробникам різних програм використовувати інтеграцію інформації зі своїх

сервісів на своїх платформах. Завдяки цьому ви зможете переглянути відео взяті з відео хостингу YouTube.com прямо всередині програми.

Велика кількість комерційних компаній пропонують API як готовий до використання продукт. Так, американська компанія Weather Underground заробляє рахунок продажу доступу до свого API для оперативного отримання метеорологічних даних у будь-якій точці нашої планети.

Інтерфейс є проміжним шаром між двома додатками. Він дозволяє двом програмам обмінюватися інформацією та виконувати функції, не розкриваючи свого внутрішнього API. Приховування частини функцій називається інкапсуляцією.

Є три методи взаємодії з API:

- 1) Процес, який може виконувати програма за допомогою цього інтерфейсу.
- 2) Дані, які потрібно передати інтерфейсу для виконання функції.
- 3) Дані, які програма отримає після роботи з API.

Розробник має повну свободу у вибудовуванні функцій API. Наприклад, окремий набір функцій може визначати можливість реєструватися та авторизуватись у програмі.

API бувають публічні та приватні. Перші призначені для спільного використання із зовнішнім світом, наприклад, API YouTube. Сторонні розробники можуть створювати програми, щоб користуватися можливостями цих інтерфейсів. Другі – це внутрішні програми, розроблені для певної аудиторії чи бази користувача. Вони часто використовуються на підприємствах та всередині компаній. Для роботи з таким API необхідно отримати доступ.

Для чого використовують API. Розробникам програмний інтерфейс дозволяє:

- спростити та прискорити випуск нових продуктів, тому що можна використовувати вже готові API для стандартних функцій;

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		45

- зробити розробку безпечнішою, вивівши ряд функцій в окрему програму, де вони будуть приховані;
- спростити налаштування зв'язків між різними сервісами та програмами та не співпрацювати для розробки свого продукту із творцями різних додатків;
- заощадити гроші, тому що не потрібно розробляти всі програмні рішення з нуля.

До появи Windows та інших графічних операційних систем програмістам до створення вікон на екрані комп'ютера доводилося писати тисячі рядків коду. Коли ж Microsoft надала розробникам API Windows, на створення вікон стало витрачатися всього кілька хвилин роботи.

Бізнесу API потрібні, щоб:

- проводити транзакції;
- інтегрувати потоки даних з клієнтами та партнерськими системами;
- підвищити безпеку автоматизованих процесів;
- розвивати власні програми;
- впроваджувати інновації, наприклад, під час роботи з клієнтами.

У 1990-ті роки організація, яка хотіла запустити систему управління взаємовідносинами з клієнтами (CRM), була змушена вкладати величезні кошти у програмне забезпечення, обладнання та спеціалістів. Тепер компанії використовують хмарні служби на зразок Salesforce. Доступ на рівні API до функцій Salesforce дозволяє бізнесу включити ключові елементи функціональності CRM-системи, наприклад, можливість переглядати історію клієнта.

Урядам API дозволяють:

- обмінюватись даними між відомствами;
- взаємодіяти з громадянами, отримувати зворотний зв'язок.

Вже в 40 містах США використовується безкоштовний API Open311, який дозволяє відслідковувати проблеми на основі розташування користувача. Людині достатньо лише відправити до міської системи фото з вибоїною на дорозі та вказівкою геолокації.

Успіх Телеграму в значній мірі обумовлений наявністю великої кількості ботів - це невеликі сервісні програми, які може створити будь-який користувач з середнім рівнем програмування. Telegram API Bot - це програмний інтерфейс, який надає можливість програмувати власний робот для Телеграму.

API включає набір об'єктів та команд, які визначають поведінку ботів в Telegram. За допомогою цього інтерфейсу ви можете створювати власні програмні коди, які при запуску Telegram стають активними ботами та починають виконувати встановлені завдання.

Бот – скорочений від «робот». Так називають автономну програму, яка працює в інтернеті та виконує за людей деякі завдання. Так виходить звільнити людські ресурси там, де їх недостатньо або їх використання невігідно.

Наприклад, бот може бути консультантом. Його інтегрують у чати та на сайти для створення там активності та відповідей на прості питання. Якщо він не впорається з поставленим завданням, то переведе діалог на реальну людину з підтримки. Так вдається знизити навантаження реальних фахівців, зменшити потік нецільових звернень до них.

Боти в Telegram, наприклад, завантажують фільми (наприклад, @kinomanBot), конвертують файли (@topdf_bot), допомагають засвоювати іноземні мови, знаходити пісні в інтернеті за уривком (@auidbot), навіть заробляти в інтернеті (@Skinve – показує друзям рекламу в Skype, де заробляє користувач). Вони перекладають тексти, розважають, допомагають та розповідають, як витратити час із користю.

Бот може виконувати будь-які завдання, закладені у нього розробником. Наприклад, може надсилати електронні листи, відповідати на запитання

користувачів, надсилати їх на сторінки сайту або до фахівця. За допомогою робота можна дізнатися курс валюти, завантажити музику, знайти книги, поширити якусь інформацію, навіть вплинути на репутацію бренду.

На сьогоднішній день існує два основних інструменти API, якими можна користуватися для використання сервісів Telegram: Telegram Bot API і Telegram API. Перший інструмент призначений для розробки чат-ботів, а другий дозволяє створювати повністю настроювані клієнти Telegram. Розробникам також доступна відкрита бібліотека TDLib (Telegram Database Library), яка дає змогу створювати власні версії месенджера з унікальними можливостями (наприклад, Telegram X, який побудований на базі TDLib). Telegram Bot API є розширенням Telegram API, тому використання Bot API можливе без необхідності знати деталі протоколу MTProto, що використовується.

Для правильної роботи цієї системи використовується проміжний сервер з HTTPS-інтерфейсом, який забезпечує шифрування трафіку та встановлює зв'язок з Telegram API. За допомогою Bot API легко створювати програми, які взаємодіють з інтерфейсом Telegram та виконують код на локальному сервері. Користувачі можуть взаємодіяти з ботами, надсилаючи їм повідомлення, команди та вбудовані запити.

У роботі будь-якого робота присутній принцип, за яким він неперервно посилає запити на сервер і регулярно отримує оновлення. Існують два способи отримання цих оновлень. Перший - використання вебхуків, коли сервер здійснює зворотній виклик на вказану URL-адресу. Другий – активно надсилати запити до Telegram і отримувати постійні відповіді.

Важливо зауважити, що отримання сповіщень про нові повідомлення та події в роботі можливе тільки один раз. Тому, якщо вам важливі дані з чатів, вам потрібно самостійно зберігати список чатів та історію старих повідомлень. Якщо ви випадково втратите або видалите цю інформацію, відновити її буде неможливо.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		48

На відміну від Bot API, де обмеження полягає в одноразовому отриманні оновлень, в Telegram API цю обмеженість можна обійти, використовуючи кілька клієнтів. У цьому випадку робот отримує всі оновлення на кожному з запущених клієнтів. Крім того, в Bot API немає можливості надсилати повідомлення всім користувачам одночасно.

На GitHub (<https://github.com/tdlib/telegram-bot-api>) можна знайти вихідні дані сервера Bot API. Розгорнувши такий сервер, ви отримаєте HTTP URL з можливістю організації роботи через вебхуки. Також можете завантажувати файли (до двох терабайт) на локальний сервер. Сервер Bot API має можливість налаштування порту та локальної IP-адреси для вебхука, а також підтримує до 100000 одночасних підключень - більш ніж достатньо для більшості завдань.

Telegram API підтримується багатьма мовами програмування. Це дає можливість вибору автору.

Любителі JavaScript можуть використовувати Node.js Telegram Bot API. Тут необхідно знання як мови, а й уміння поводитися з цим фреймворком, перетворивши клієнтську мову на повноцінний серверний інтерфейс.

Одним із найпопулярніших для написання ботів з використанням Telegram Bot API є PHP. Ця мова спочатку була призначена для створення серверних веб-додатків. Він відрізняється простотою, логічністю та спеціалізованістю саме для web-середовища.

Часто використовується Telegram Bot API у Python. Ця мова відрізняється мінімалізмом і досить проста у вивченні. Він дуже популярний завдяки своїй продуктивності.

Класикою є застосування Telegram Bot API C++. Мова не можна назвати простою, але вона є базою, на якій були створені всі інші перераховані вище ЯП. Відповідно в ньому не закладено певної спеціалізації. Інструменти дозволяють створювати будь-які програми.

Крім об'єктів API має набір методів, які дозволяють надсилати повідомлення, файл, фото стікери, редагувати та багато іншого. Всі ці команди можна знайти в описі API на офіційному сайті.

Для створення в Telegram є спеціальний сервіс @Botfather. Зайдіть у нього та побачите набір команд, за допомогою яких створюється новий робот. Створення боту буде розглянуто далі.

У сучасний час неможливо уникнути використання API-бібліотек для створення якісних та корисних сервісів. Вони є необхідними для розробки програмного забезпечення та додатків програмістами, а також для надання різноманітних послуг обслуговування клієнтів. Кожним роком роль і область застосування API лише зростають, і вони стають ще більш невід'ємною частиною сучасного сервісного середовища.

Сьогодні це явний тренд, чат-боти можуть створювати продажі не гірше за досвідчених менеджерів, а також беруть на себе масу функцій, які вимагали раніше людських ресурсів, а зокрема, запис на прийом до лікаря, продаж квитків і так далі.

До появи Enterprise Java Beans (EJB) розробникам Java необхідно було використовувати JavaBeans для створення веб-додатків. Хоча JavaBeans допоміг у розробці компонентів інтерфейсу користувача (UI), вони не змогли надати такі послуги, як керування транзакціями та безпека, які були необхідні для розробки надійних і безпечних корпоративних програм. Поява EJB розглядалося як рішення цієї проблеми. EJB розширює компоненти Java, такі як веб- та корпоративні компоненти, і надає послуги, які допомагають у розробці корпоративних додатків. Однак розробка корпоративного додатка за допомогою EJB була нелегкою, оскільки розробнику потрібно було виконувати різні завдання, такі як створення домашнього та віддаленого інтерфейсу та впровадження методів зворотного виклику життєвого циклу, що призводило до складності надання коду для EJB через цю складність,

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		50

Фреймворк Spring з'явився як рішення для всіх цих ускладнень. Ця структура використовує різні нові методи, такі як аспектно-орієнтоване програмування (AOP), звичайний старий Java-об'єкт (POJO) та ін'єкція залежностей (DI), для розробки корпоративних програм, видаляючи тим самим У зв'язку зі складнощами, пов'язаними з розробкою корпоративних додатків за допомогою EJB, Spring є легкою структурою з відкритим вихідним кодом, яка дозволяє розробникам Java EE 7 створювати прості, надійні та масштабовані корпоративні програми. Ця структура в основному зосереджена на наданні різних способів, які допоможуть вам керувати своїми бізнес-об'єктами. Це значно полегшило розробку веб-додатків у порівнянні з класичними фреймворками Java та інтерфейсами програмного забезпечення (API), такими як підключення до бази даних Java (JDBC), JavaServer Pages (JSP) і Java Servlet.

Фреймворк Spring можна розглядати як набір субфреймворків, також званих шарами, наприклад Spring AOP. Spring Object-Relational Mapping (Spring ORM). Spring Web Flow і Spring Web MVC. Ви можете використовувати будь-який з цих модулів окремо під час створення веб-програми. Модулі також можуть бути згруповані разом, щоб забезпечити кращу функціональність у веб-додатку.

Функції фреймворку Spring, такі як IoC, AOP та управління транзакціями, роблять його унікальним серед переліку фреймворків. Ось деякі з найважливіших особливостей фреймворку Spring:

Контейнер IoC:

Відноситься до основного контейнера, який використовує шаблон DI або IoC для неявного надання посилання на об'єкт у класі під час виконання. Цей шаблон діє як альтернатива шаблону локатора служби. Контейнер IoC містить код асемблера, який обробляє керування конфігурацією об'єктів програми.

Фреймворк Spring надає два пакети, а саме `org.springframework.beans` і `org.springframework.context`, які допомагають забезпечити функціональність контейнера IoC.

Фреймворк доступу до даних:

є набором інструментів, які дозволяють розробникам використовувати API для зберігання даних у базі даних, такі як JDBC і Hibernate. Використання цих інструментів допомагає розв'язати різні проблеми розробників, такі як взаємодія з підключенням до бази даних, забезпечення правильного закриття з'єднання, робота з винятками та реалізація управління транзакціями. Крім того, цей фреймворк дозволяє легко писати код для доступу до даних у всій програмі.

Фреймворк Spring MVC:

дозволяє створювати веб-додатки, використовуючи архітектуру MVC. Кожен запит, зроблений користувачем, спочатку обробляється контролером, а потім передається до різних представлень, таких як JSP або сервлети. Функції, пов'язані з обробкою форм та їх перевіркою, легко інтегруються з різними технологіями для відображення, такими як ISP, Jasper Report, FreeMarker і Velocity, у фреймворку Spring MVC.

Керування транзакціями:

допомагає управляти транзакціями програми, не впливаючи на її код. Ця структура надає API транзакцій Java (JTA) для глобальних транзакцій, якими керує сервер додатків, і локальних транзакцій, керованих за допомогою JDBC Hibernate, Java Data Objects (JDO) або інших API доступу до даних. Це дозволяє розробнику моделювати широкий спектр транзакцій на основі декларативного та програмного управління транзакціями Spring.

Spring Web Service:

генерує кінцеві точки та визначення веб-сервісів на основі класів Java, але керувати ними в додатку важко. Щоб вирішити цю проблему, Spring Web Service надає багатопарові підходи, які окремо керуються розширюваною

					ДП.КН 23.519.03.000 ПЗ	Анк.
						52
Зм.	Арк.	№ докум.	Підпис.	Дата		

мовою розмітки (XML) (техніка читання та маніпулювання XML). Spring забезпечує ефективне відображення для передачі вхідного запиту XML-повідомлення до об'єкта і розробнику для легкого розподілу XML-повідомлення (об'єкта) між двома машинами.

Рівень абстракції JDBC:

допомагає користувачам легко та ефективно обробляти помилки. Програмний код JDBC можна зменшити, якщо цей рівень абстракції реалізовано у веб-додатку. Цей рівень обробляє винятки, такі як `DriverNotFound`. Усі `SQLExceptions` перекладаються в клас `DataAccessException`. Виняток для доступу до даних Spring не стосується JDBC, тому об'єкти доступу до даних (DAO) не прив'язані лише до JDBC.

Spring TestContext Framework:

надає засоби модульного та інтеграційного тестування для додатків Spring. Крім того, фреймворк Spring TestContext надає конкретні функції тестування інтеграції, такі як керування контекстом і кешування DI тестових приладів, а також керування транзакційними тестами із семантикою відкату за замовчуванням.

Spring Framework був вперше випущений у 2004 році. Після цього було внесено значну редакцію, наприклад, Spring 2.0 надав простори імен XML і підтримку AspectJ, Spring 2.5 забезпечує конфігурацію на основі анотацій, Spring 3.0 надає модель `@Configuration` на основі Java. Останній випуск spring framework 4.0. він випускається з підтримкою технологій Java 8 і Java EE 7. Хоча ви все ще можете використовувати Spring зі старішою версією Java, мінімальна вимога обмежена Java SE 6. Spring 4.0 також підтримує технології Java EE 7, такі як java message service (JMS) 2.0, java persistence API (JPA) 2.1, Перевірка компонента 1.1, сервлет 3.1 і JCache.

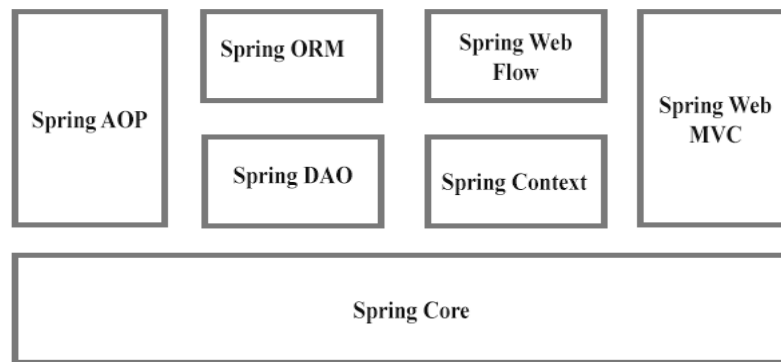


Рисунок 3.5. – Структура Spring

Структура Spring складається з семи модулів, які показані на (рис. 3.5.) Такими модулями є Spring Core, Spring AOP, Spring Web MVC, Spring DAO, Spring ORM, Spring context і Spring Web flow. Ці модулі забезпечують різні платформи для розробки різних корпоративних додатків; наприклад, ви можете використовувати модуль Spring Web MVC для розробки додатків на основі MVC.

Spring Core Module:

Модуль Spring Core, який є основним компонентом фреймворку Spring, забезпечує контейнер IoC. Є два типи реалізацій контейнера Spring, а саме, фабрика bean і контекст програми. Фабрика бінов визначається за допомогою інтерфейсу `org.springframework.beans.factory.BeanFactory` і діє як контейнер для beans. Заводський контейнер Bean дозволяє відокремити конфігурацію та специфікацію залежностей від логіки програми. У фреймворку Spring фабрика Bean діє як центральний контейнер IoC, який відповідає за створення екземплярів об'єктів програми. Він також налаштовує та збирає залежності між цими об'єктами. Існує безліч реалізацій інтерфейсу `BeanFactory`. Клас `XmlBeanFactory` є найпоширенішою реалізацією інтерфейсу `BeanFactory`.

Модуль Spring AOP:

Подібно до об'єктно-орієнтованого програмування (ООП), яке розбиває програми на ієрархію об'єктів, AOP розбиває програми на аспекти або проблеми. Модуль Spring AOP дозволяє вам реалізувати проблеми або аспекти в

додатку Spring у Spring AOP, аспектами є звичайні Spring bean або звичайні класи, анотовані анотацією @Aspect. Ці аспекти допомагають в управлінні транзакціями, веденні журналів і моніторингу збоїв програми. Наприклад, управління транзакціями потрібне в банківських операціях, таких як передача суми з одного рахунку на інший. Модуль Spring AOP забезпечує рівень абстракції управління транзакціями, який можна застосувати до API транзакцій.

Модуль Spring ORM:

Модуль Spring ORM використовується для доступу до даних із баз даних у програмі. Він надає API для маніпулювання базами даних за допомогою JDO, Hibernate та iBatis. Spring ORM підтримує DAO, що забезпечує зручний спосіб створення таких рішень ORM на основі DAO:

- просте декларативне управління транзакціями;
- прозора обробка винятків;
- потокобезпечні, легкі класи шаблонів;
- класи підтримки DAO;
- управління ресурсами.

Модуль Spring Web MVC:

Web MVC Spring реалізує архітектуру MVC для створення веб-додатків. Він розділяє код моделі та компоненти перегляду веб-додатка. У Spring MVC, коли запит генерується з браузера, він спочатку надходить до класу DispatcherServlet (Front Controller), який відправляє запит на контролер (клас SimpleFormController або клас AbstractWizardformController) за допомогою набору зіставлень обробників. Контролер витягує та обробляє інформацію, вбудовану в запит, і надсилає результат до класу DispatcherServlet у вигляді об'єкта моделі. Нарешті, клас DispatcherServlet використовує класи ViewResolver для надсилання результатів у представлення, яке відображає ці результати користувачам.

Spring Web Flow:

Модуль Spring Web Flow є розширенням модуля Spring Web MVC. Фреймворк Spring Web MVC надає контролери форм, такі як клас SimpleFormController і клас AbstractWizardFormController, для реалізації попередньо визначеного робочого процесу. Spring Web Flow допомагає визначити файл XML або клас Java, який керує робочим процесом між різними сторінками веб-програми. Spring Web Flow розповсюджується окремо і може бути завантажений з веб-сайту <http://www.springframework.org>.

Нижче наведено переваги Spring Web Flow:

- Потік між різними інтерфейсами інтерфейсу програми чітко забезпечується визначенням веб-потoku у файлі XML.
- Визначення веб-потoku допомагають вам віртуально розділити програму на різні модулі та повторно використовувати ці модулі в різних ситуаціях.
- Життєвим циклом Spring Web Flow можна керувати автоматично.

Модуль Spring Web DAO:

Пакет DAO у фреймворку Spring забезпечує підтримку DAO за допомогою технологій доступу до даних, таких як JDBC, Hibernate або JDO. Цей модуль представляє рівень абстракції JDBC, усуваючи потребу в наданні втомливого кодування JDBC. Він також надає програмні та декларативні класи керування транзакціями. Пакет Spring DAO підтримує гетерогенну підключення до бази даних Java та відображення O/R, що допомагає Spring працювати з кількома технологіями доступу до даних. Для легкого та швидкого доступу до ресурсів бази даних фреймворк Spring надає абстрактні базові класи DAO. Для кожної технології доступу до даних, яка підтримується фреймворком Spring, доступно кілька реалізацій. Наприклад, у JDBC клас JdbcDaoSupport та його методи використовуються для доступу до екземпляра DataSource і попередньо налаштованого екземпляра JdbcTemplate.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		56

Spring Application Context Module:

Контекстний модуль програми Spring заснований на модулі Core. Контекст програми `org.springframework.context.ApplicationContext` — це інтерфейс `BeanFactory`. Цей модуль отримав свою функцію з пакета `org.springframework.beans`, а також підтримує такі функції, як інтернаціоналізація (I18N), перевірка, поширення подій і завантаження ресурсів. Контекст програми реалізує інтерфейс `MessageSource` і надає програмі функціональні можливості обміну повідомленнями.

Spring Framework може бути розглянутий як колекція менших фреймворків або фреймворків у фреймворку. Більшість цих фреймворків може працювати незалежно один від одного, однак вони забезпечують більшу функціональність при спільному їх використанні.

Особливості ядра Spring Framework застосовні в будь-якій Java-додатку, і існує безліч розширень та удосконалень для побудови веб-додатків на Java Enterprise платформі. З цих причин Spring набув великої популярності і визнається розробниками як стратегічно важливий фреймворк.

Ручне складання проектів на Java досить трудомісткий процес. Потрібно правильно вказати потрібні проекту бібліотеки та фреймворки, від яких проект залежить.

Gradle — система для автоматизації складання додатків та збору статистики про використання програмних бібліотек, що застосовує мови Groovy, Java, JavaScript, Kotlin тощо, а також рішення з фреймворків Apache Ant та Apache Maven.

Gradle написана мовами Java, Groovy та Kotlin, також підтримує плагіни на JavaScript, C++, Swift, Scala. Система розповсюджується як програмне забезпечення з відкритим кодом за ліцензією Apache License 2.0.



Рисунок 3.6. Gradle

Ручне складання програми для запуску на пристрої потребує багато часу та ресурсів. Вона затягує процес розробки та релізу.

Щоб прискорити його, програмісти створюють скрипти, що автоматизують виконання типових завдань. Це скорочує час збирання, але ускладнює роботу над командним проектом іншим розробникам.

Проблему вирішили системи автоматичного збирання додатків. Перша – Ant – з'явилася у 2000 році. 2004 року вийшла Maven. Для опису сценаріїв збирання системи використовували мову розмітки XML. Його недолік – неконтрольоване розростання під час спроб зібрати великий багатомодульний проект.

У 2006 року з'явилася перша версія Gradle. Вона дозволила вирішити основні проблеми:

- реалізацію великих багатомодульних збірок шляхом послідовного та паралельного виконання основних та додаткових модулів;
- прискорення збирання за рахунок кешування компонентів через спільну мережу з використанням Gradle Build Cache;
- управління залежностями, тобто фреймворками, бібліотеками та іншими компонентами, що підключаються до проекту;
- спрощення робочого процесу за рахунок автоматизації типових завдань та веб-візуалізації складання.

Gradle використовує багато рішень, які застосовуються в Ant та Maven. Три системи широко інтегровані одна з одною: проект, розроблений для однієї з

них, можна зібрати в іншій. Наприклад, основним джерелом зовнішніх компонентів, що підключаються в Gradle, є сховище Maven.

У Gradle використовується діалект мов Groove та Kotlin замість XML. Обидві об'єктно-орієнтовані мови розроблені як доповнення до Java, використовують схожий синтаксис, сумісні з тими самими бібліотеками та іншим Java-кодом. Завдяки йому стало простіше писати сценарії складання та порівняно швидко створювати складні проекти.

Виконання завдань у Gradle базується на спрямованих ациклічних графах. Вони вузли пов'язані не циклами, а шляхами, які можуть виходити з початкового вузла і в кінцевий пункт приходити різними способами. Ациклічні графи оптимізують внутрішню об'єктну модель програми та управління залежностями, підвищують гнучкість складання.

Два приклади описують виконання однієї й тієї самої задачі (рис .3.7.).

Код на Gradle простіший і менший. Різниця помітна при реалізації збірок із великою кількістю залежностей.

Gradle має декларативний характер складання. Тобто розробник просто ставить цілі, бажаний кінцевий результат, а система сама підбирає шляхи його досягнення. Завдяки використанню загальних принципів проектування Gradle дозволяє створити зручний, зрозумілий та швидко реалізований проект.

Також система має детальний і добре продуманий програмний інтерфейс спрощує відстеження та налаштування конфігурації складання, контроль виконання.



Рисунок 3.7. Включення залежностей у проект з використанням Maven та Gradle.

Система сумісна з завданнями Ant, інфраструктурою репозиторію (мережевого сховища) Maven та Ivy, де можна опублікувати та отримати залежні компоненти. Є також конвертери для перетворення Maven pom.xml на скрипт Gradle.

Gradle легко пристосовується до будь-якої структури, завдяки чому можна переносити з нього та на нього проекти, розроблені для інших систем складання.

Система розповсюджується за вільною ліцензією Apache Software License (ASL), тому сторонні розробники змогли написати багато корисних плагінів, бібліотек та інших компонентів. Вони розширюють функціонал та створюють повноцінну Gradle-екосистему.

Gradle Wrapper – опція дозволяє реалізацію збірок, створених у Gradle, на машинах, де система не встановлена. Це спрощує безперервну інтеграцію серверів.

Groove, на відміну від жорстко заданих XML-ієрархій Ant та Maven, дає розробнику більше варіантів дії, що дозволяє оптимізувати проект, що показує гнучкість мови.

Підтримка каскадної моделі у Gradle означає, що система сама визначає, які компоненти дерева розробки були змінені чи залишилися незмінними. І, відповідно, які залежать від них завдання потрібно перезапустити та оновити.

У загальному вигляді Gradle створює текстовий файл сценарію збирання (BuildScript), що складається з наступних компонентів:

- **Project (проект).** Це кінцевий результат роботи сценарію. Проект являє собою JAR-файл, веб-додаток або zip-файл, що включає Java-архіви, створені іншими проектами. Проект може складатися як із одного модуля (одномодульний проект), так і з кількох підмодулів (багатомодульний проект). У багатомодульному проекті кожен підмодуль можна як самостійний Gradle-проект.
- **Task (завдання).** Це набір дій із проектом. Завдання можуть бути пов'язані один з одним. Під складання проекту розуміється виконання однієї чи кількох пов'язаних завдань. Проект вважається успішно зібраним, коли виконується завдання чи набір пов'язаних завдань. Як правило, результат складання – набір артефактів, дистрибутивів, їх розміщення у репозиторії, встановлення та запуск на іншому комп'ютері в Мережі.
- **Dependency (залежності).** У ході збирання можуть бути використані залежності. Вони бувають зовнішніми та внутрішніми. Зовнішні - бібліотеки, створені розробниками з усього світу та розташовані в репозиторіях (repositories). Внутрішні (підмодулі у багатомодульних проектах) - бібліотеки, розроблені всередині поточного проекту та оформлені у вигляді підмодуля.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		61

Опис проекту знаходиться у файлі `build.gradle` (рис. 3.8.), налаштування вказуються у файлі `gradle.settings`. Функціональність скрипта `build.gradle` може бути розширена за допомогою службового підпроекту `buildSrc` всередині проекту, а також модулів, що підключаються, – плагінів. Плагіни поділяються на вбудовані та сторонні. Сторонні плагіни можна знайти на офіційному сайті.

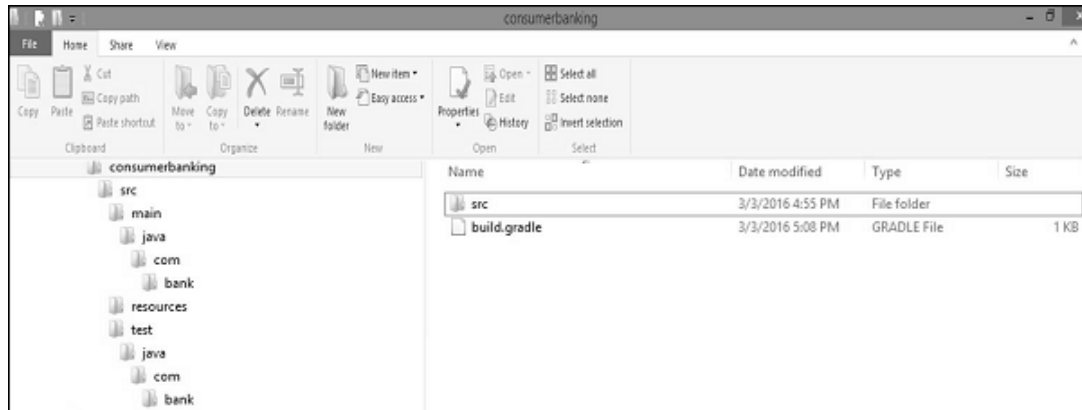


Рисунок 3.8. Структура каталогів та розміщення файлу `build.gradle`.

На відміну від Apache Maven, заснованого на концепції життєвого циклу проекту, і Apache Ant, у якому порядок виконання завдань (targets) визначається відносинами залежності (depends-on), Gradle використовує спрямований ациклічний граф визначення порядку виконання завдань.

Gradle був розроблений для багатопроєктних зборок, що розширюються, і підтримує інкрементальні зборки, визначаючи, які компоненти дерева збірки не змінилися і які завдання, залежні від цих частин, не вимагають перезапуску.

Gradle – є хорошим вибором як системи складання проєктів. Підтвердженням цього є те, що його використовують розробники таких відомих проєктів як Spring та Hibernate. Вище було розглянуто лише самі базові речі. За ними прихований мільйон особливостей та можливостей, які з'являються у розробників. Gradle також підтримує створення багатомодульних проєктів.

3.2 Реалізація функціоналу чат-боту

Для створення клавiш боту у Телеграм iснує два рiзні види клавiатур. Для початку варто зрозуміти, в чому основна вiдмiннiсть ReplyKeyboardMarkup i InlineKeyboardMarkup.

ReplyKeyboardMarkup – це шаблони повiдомлень. Наприклад, бот ставить користувачевi питання та пропонує варіанти вiдповiдi. Користувач може самостійно надрукувати вiдповiдь або натиснути на готову кнопку. Така клавiатура вiдображається замість основної i не прив'язана до жодного повiдомлення. У кнопки такої клавiатури не можна закласти жодної iнформації. Якщо користувач натискає кнопку з текстом «abc» вiдправити текст «qwerty» неможливо, надiслано буде те, що написано на кнопцi (рис. 3.9.)

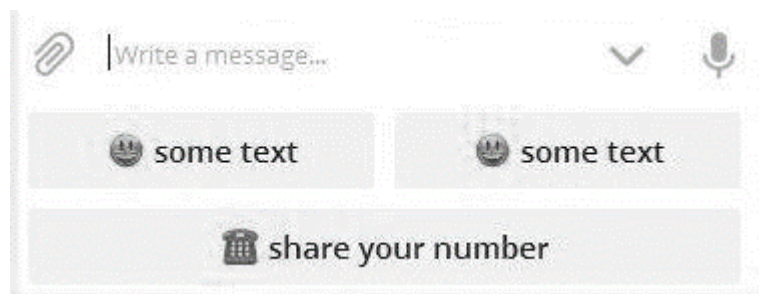


Рисунок 3.9. – Приклад клавiатури ReplyKeyboardMarkup.

InlineKeyboardMarkup – це вже справжня кастомна клавiатура. З її допомогою можна виконувати складніші дії. Вона прив'язується до повiдомлення, з яким було вiдправлено. У кнопки можна закласти будь-який текст розміром вiд 1 до 64 байт. iнлайн кнопки дозволяють приховати в собі внутрішні посилання, посилання на зовнішній ресурс, а також шорткат для iнлайн запиту.



Рисунок 3.10 – Приклад клавіатури InlineKeyboardMarkup.

І ту й іншу клавіатуру можна редагувати, але у різний спосіб. Перша оновлюється при надсиланні повідомлення з новою клавіатурою типу ReplyKeyboardMarkup, другу можна редагувати разом із повідомленням, до якого вона прикріплена (або лише розмітку).

Для виконання вимог проекту використовуватимуться обидва типи клавіатур. Детальний код клавіатур приведено у додатках до роботи.

4 ТЕХНІКО-ЕКОНОМІЧНЕ ОБГРУНТУВАННЯ

Тема даного дипломного проекту полягає в створенні Telegram бота, який буде використовуватися для перегляду та публікації результатів змагань з ловлі хижої риби. Метою економічної частини цього проекту є проведення розрахунків з метою визначення економічної ефективності розробки Telegram бота та прийняття рішення щодо його подальшого розвитку та впровадження.

Оцінюватимуться переваги та недоліки роботи бота, а також будуть розглянуті альтернативні варіанти. На основі отриманих результатів буде прийняте рішення про доцільність подальшої роботи над проектом.

Об'єктом розробки є Telegram бот.

Розрахунок вартості розробки виконується в декілька етапів:

- опис технологічного процесу розробки із зазначенням трудомісткості кожної операції;
- визначення суми витрат на оплату праці основного і допоміжного персоналу, включаючи відрахування на соціальні заходи;
- обчислення витрат на електроенергію для науково-виробничих цілей;
- складення кошторису та визначення собівартості робіт;
- розрахування ціни робіт;

4.1 Визначення стадій технологічного процесу та загальної тривалості проведення робіт

Базуючись на даних сформованого технічного завдання виділено 9 стадій розробки Telegram боту.

Дані про витрати часу на проведення робіт наведені в таблиці 4.1.

Таблиця 4.3 – Середній час виконання робіт по обслуговуванню та стадії (операції) технологічного процесу.

№ п/п	Назва операції (стадії)	Виконавець	Середній час виконання операції, год.
1.	Постановка задачі та формування технічного завдання	Керівник проекту	4
2.	Розробка дизайну Telegram боту	Програміст	5
3.	Затвердження дизайну	Керівник проекту	2
4.	Розробка frontend та backend	Програміст	25
5.	Тестування Telegram боту	Програміст	3
6.	Документування	Програміст	10
7.	Здача Telegram боту	Програміст	2
		Керівник проекту	2
Разом		-	53

Сумарний час виконання операцій технічного процесу становить 53 годин.

4.2 Оцінка витрат на оплату праці та відрахувань на соціальні заходи

Угідно до законодавства України, "Про оплату праці", заробітна плата визначається як "винагорода, яка, як правило, виражається у грошовій формі". Розмір заробітної плати залежить від складності та умов виконуваної роботи, професійних навичок працівника, результатів його праці та ефективності діяльності підприємства. Заробітна плата складається з основної та додаткової частини оплати праці. Основна заробітна плата – це винагорода за виконану

роботу відповідно до встановлених норм праці (норми часу, виробітку, обслуговування, посадові обов'язки).

Додаткова заробітна плата – це винагорода за працю понад установлені норми за трудові успіхи та винахідливість і за особливі умови праці, що включає доплати, надбавки, гарантійні і компенсаційні виплати, передбачені чинним законодавством; премії, пов'язані з виконанням виробничих завдань і функцій.

Джерелом додаткової оплати праці є фонд матеріального стимулювання, який створюється за рахунок прибутку.

Основна заробітна плата розраховується за формулою:

$$Z_{осн.} = T_c \cdot K_g, (4.1)$$

де: T_c – тарифна ставка, грн.;

K_g – кількість відпрацьованих годин.

Отже основна заробітна плата для:

- Керівника проекту $Z_{осн1} = 130 \cdot 8 = 1040$ грн.
- програміст $Z_{осн2} = 170 \cdot 45 = 7650$ грн.

Сумарна основна заробітна плата становить:

$$Z_{осн} = 1040 + 7650 = 8690 \text{ грн.}$$

Додаткова заробітна плата становить 10 – 15 % від суми основної заробітної плати.

$$Z_{дод.} = Z_{осн.} \cdot K_{допл.}, (4.2)$$

де $K_{допл.}$ – коефіцієнт додаткових виплат працівникам, 0,1–0,15.

Отже додаткова заробітна плата по категоріях працівників становить:

- Керівника проекту $Z_{дод1} = 1040 \cdot 0,1 = 104$ грн.
- інженера $Z_{дод2} = 7650 \cdot 0,1 = 765$ грн.

Загальна додаткова заробітна плата становить:

					ДП.КН 23.519.03.000 ПЗ	Анк.
						67
Зм.	Арк.	№ докум.	Підпис.	Дата		

$$З_{\text{дод}} = 104 + 765 = 869 \text{ грн.}$$

Звідси загальні витрати на оплату праці (Во.п.) визначаються за формулою:

$$B_{\text{ол.}} = З_{\text{осн.}} + З_{\text{дод.}},$$

$$B_{\text{ол.}} = 8690 + 869 = 9559 \text{ грн} \quad (4.3)$$

Крім того, слід визначити відрахування на соціальні заходи:

- єдиний соціальний внесок – 22%

Отже, сума відрахувань на соціальні заходи буде становити:

$$B_{\text{с.з.}} = \text{ФОП} \cdot 0,22 \quad (4.4)$$

де ФОП – фонд оплати праці, грн.

$$B_{\text{с.з.}} = 9559 \cdot 0,22 = 2103 \text{ грн.}$$

Проведені розрахунки витрат на оплату праці зведемо у таблицю 4.4.

Таблиця 4.4 - Зведені розрахунки витрат на оплату праці

№ п/п	Категорія працівників	Основна заробітна плата, грн.			Додаткова заробітна плата, грн.	Нарахування на ФОП, грн.	Всього витрати на оплату праці, грн.
		Тарифна ставка, грн.	К-сть від-працьо в. год.	Фактично нараховано з/пл., грн.			
1	Керівник проекту	130	8	1040	104	-	-
2	Програміст	170	45	7650	765	-	-
Разом				8690	869	2103	11662

Отже, загальні витрати на оплату праці становлять 11662 грн.

4.3 Розрахунок витрат на електроенергію

Затрати на електроенергію 1-ці обладнання визначаються за формулою:

$$Z_e = W \cdot T \cdot S, \quad (4.5)$$

де: W – необхідна потужність, кВт;

T – кількість годин роботи обладнання;

S – вартість кіловат-години електроенергії.

Час роботи ПК над даним проектом становить 53 годин, споживана потужність - 0.5 кВт/год.

$$Z_e = 0.5 \cdot 53 \cdot 1,80 = 47,7 \text{ грн.}$$

Загальні витрати на електроенергію становлять $Z_e = 47,7$ грн.

4.4 Розрахунок матеріальних витрат

Матеріальні витрати — це вартість витрачених матеріалів, малоцінних та швидкозношуваних предметів на виробництво продукції, робіт або послуг, а також матеріалів, витрачених на адміністративні, збутові та інші потреби.

Загальна сума витрат на матеріальні ресурси (B_M) визначається за

Формулою

$$B_M = \sum_{i=1}^n K_i \cdot C_i \quad (4.6)$$

де K_i - витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i - ціна за одиницю i -го типу матеріалу, грн.;

i - тип матеріального ресурсу;

n - кількість типів матеріальних ресурсів.

Звідси, витрати на матеріальні ресурси дорівнюватимуть:

$$B_M = 332 \text{ грн.}$$

Проведені розрахунки занесемо у таблицю 4.5.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		69

Таблиця 4.5 - Зведені розрахунки покупних виробів

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Загальна сума, грн
1	Папір (формат А4)	уп	180	1	180	220,0
3	Зшивання диплому	ч.р	110	1	110,0	110,0
Разом						332

4.5 Обчислення накладних витрат

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 20 – 60 % від суми основної та додаткової заробітної плати працівників.

$$H_{\varepsilon} = B_{o.n.} \cdot 0,2 \dots 0,6, \quad (4.7)$$

де НВ – накладні витрати.

$$H_{\varepsilon} = 11200 \cdot 0,3 = 3360 \text{ грн.}$$

4.6 Складання кошторису витрат та визначення собівартості робіт

Результати проведених розрахунків витрат зведемо у таблицю 4.4.

Таблиця 4.6 - Кошторис витрат

Зміст витрат	Сума, грн.	В % до загальної суми
1	2	3
Витрати на оплату праці (основну і додаткову заробітну плату)	9559	61,53
Відрахування на соціальні заходи	2103	13,53
Витрати на електроенергію	47,7	0,31
1	2	3
Накладні витрати	3823,6	24,61
Собівартість	15533,3	99.98

Собівартість (СВ) НДР розраховуємо за формулою:

$$C_B = B_{o.n.} + B_{c.z.} + Z_{m.o.} + Z_e + T_e + A + H_e : \quad (4.8)$$

Отже, собівартість дорівнює СВ= 15533.3грн.

ВИСНОВКИ

В рамках дипломного проєкту детально розглядається процес створення Telegram-бота з використанням мови програмування Java. Основна мета цього проєкту полягає в створенні інтерактивної платформи, яка надає користувачам можливість отримувати результати змагань з ловлі риби. Для досягнення цієї мети була розроблена та використана база даних, яка забезпечує зберігання інформації про змагання та їх результати.

У рамках дипломного проєкту акцентується увага на тому, що створення Telegram-бота є не тільки ефективним, але й інноваційним способом створення інтерактивної платформи для отримання різноманітної інформації. Проведене дослідження включає детальний огляд поняття Telegram-бота та процесу його розробки з використанням мови програмування Java. Завдяки цьому боту користувачі мають зручну можливість отримувати результати змагань з ловлі риби, що робить його незамінним інструментом для спілкування та отримання актуальної інформації для рибалок.

Разом з тим, дипломний проєкт висвітлює різноманітні можливості подальшого розвитку та вдосконалення Telegram-бота. За допомогою гнучкої та розширюваної платформи Telegram, бот може бути розширений функціоналом, таким як можливість взаємодії з іншими користувачами, надання рекомендацій щодо вибору місць для лову риби, а також розширеної статистики та аналітики.

Важливо відзначити, що процес розробки Telegram-бота детально проаналізований в дипломному проєкті. Він включає кроки, починаючи від вибору необхідних бібліотек та інструментів, налаштування середовища розробки, аналізу вимог користувача та створення архітектури бота. Застосовані технології та методики програмування детально розглядаються, зокрема, робота

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		72

з API Telegram, взаємодія з базою даних, обробка команд користувача та надсилання результатів змагань.

Окрім того, дипломний проєкт розглядає потенціал Telegram-бота як інструменту для покращення користувацького досвіду та забезпечення більш широкого спектру функцій. Він надає можливість взаємодії з різними сторонніми сервісами та API, реалізацію механізму автоматизованої обробки даних та розширення функціоналу за допомогою модульної архітектури.

Загалом, дипломний проєкт пропонує комплексний підхід до створення Telegram-бота з використанням мови програмування Java для отримання результатів змагань з ловлі риби. Він підкреслює ефективність та значимість такого бота як інструменту для створення інтерактивної платформи та надає рекомендації щодо подальшого розвитку та вдосконалення.

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		73

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Алгоритми і структура даних: Навчальний посібник / В.М.Ткачук. - ІваноФранківськ : Видавництво Прикарпатського національного університету імені Василя Стефаника, 2016. 286 с.
2. Алгоритми та структури даних. Навчальний посібник / Т. О. Коротєєва. Львів : Видавництво Львівської політехніки, 2014. - 280 с.
3. Глоба Л. С. Розробка інформаційних ресурсів та систем [Електронний ресурс] : конспект лекцій / Л. С. Глоба, Т. М. Кот. - Київ : НТУУ "КПІ", 2014. - 318 с.
4. Грязнова В. О., Єфіменко С. В. Основи методології програмування. - К.: ВПЦ "Київський університет", 2010.
5. . Інженерія якості програмного забезпечення: навч. посібник / Г.В Табунщик, Р.К. Кудерметов, Т.І. Брагіна. - Запоріжжя: ЗНТУ, 2013. - 180 с.
6. Шевчук І. Б. Інформаційні технології в регіональній економіці: теорія і практика впровадження та використання : монографія. Львів : Видавництво ННВК "АТБ", 2018. 448 с.
7. Ковалюк Т.В. Основи програмування. / Ковалюк Т.В. Київ: ВНУ Києв, 2005. 400 с.
8. Николайчук Я. М. Проектування спеціалізованих комп'ютерних систем : навч. посібник / Я. М. Николайчук, Н. Я. Возна, І. Р. Пітух. - Тернопіль : ТзОВ "Терно-граф", 2010. - 392 с.
9. Java Учебник для начинающих программистов : [Електрон. ресурс]. - Режим доступу: <http://proglang.su/java>
10. Популярные технологии программирования в 2017 году : [Електрон. ресурс]. - Режим доступу: <https://vc.ru/dev/21483-what-language-2017>

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		74

ДОДАТКИ

Додаток А

Реалізація класу обробки команд боту.

```
package com.telegabot.spinning.handlers;
import com.telegabot.spinning.ReadingExcel;
import com.telegabot.spinning.Solution;
import com.telegabot.spinning.messagesender.MessageSender;
import com.telegabot.spinning.objects.Person;
import com.telegabot.spinning.services.FileWorker;
import com.telegabot.spinning.services.SaveUserService;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.Message;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.InlineKeyboardMarkup;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.ReplyKeyboardMarkup;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.InlineKeyboardBu
tton;
import
org.telegram.telegrambots.meta.api.objects.replykeyboard.buttons.KeyboardRow;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;
@Component
```

					ДП.КН 23.519.03.000 ПЗ	Анк.
						75
Зм.	Арк.	№ докум.	Підпис.	Дата		

```

public class MessageHandler implements Handler<Message> {
    private final MessageSender messageSender;
    public MessageHandler(MessageSender messageSender) {
        this.messageSender = messageSender;
    }
    @Override
    public void choose(Message message) {
        InlineKeyboardMarkup inlineKeyboardMarkup = new InlineKeyboardMarkup();
        List<InlineKeyboardButton> keyboard = new ArrayList<>();
        var readingFromExcel = new ReadingExcel();
        FileWorker fileWorker = new FileWorker();
        String result;
        if (message.hasText()) {
            System.out.println();
            SendMessage sendMessage = new SendMessage();
            sendMessage.setParseMode("HTML");
            sendMessage.setChatId(String.valueOf(message.getChatId()));
            String text = message.getText();
            if (!text.equals("TYP1") && !text.equals("TYP2") && !text.equals("TYP3"))
        {
            List<Person> list;
            List<Person> list2;
            StringBuilder str2;
            StringBuilder str;
            switch (text) {
                case "/start":
                    new SaveUserService(message);// USER SAVE
                    sendMessage.setReplyMarkup(setReplyKeyboardMarkup());

```

```

sendMessage.setParseMode("HTML");
sendMessage.setText(Solution.START_TEXT);
break;
case "Жеребьёвка":
    list = readingFromExcel.readTossFromXlsx(Solution.EXCEL_TOSS);
    if (list.isEmpty()) {
        sendMessage.setText("Такого файла нет. Или результаты еще
обрабатываются.");
    } else {
        str = new StringBuilder("<b><u>Жеребьёвка: </u></b>\n\n");
        for (Person person:list) {
            str.append(person.toStringToss()).append("\n");
        }
        sendMessage.setText(str.toString());
    }
    break;
case "Личка ТУР1":
    list =
readingFromExcel.readPersonalFromXlsx(Solution.EXCEL_PERSONAL, 0);
    if (list.isEmpty()) {
        sendMessage.setText("Такого файла нет. Или результаты еще
обрабатываются.");
    } else {
        System.out.println(list.size());
        //Чтобы не превышать символы в сообщении
        if (list.size() > 50) {
            list2 = list.subList(50, list.size());
            list = list.subList(0, 50);

```

					ДП.КН 23.519.03.000 ПЗ	Анк.
						77
Зм.	Арк.	№ докум.	Підпис.	Дата		

```

        str = new StringBuilder("<b><u>Лучный
ТУП1:</u></b>\n\n");
        for (Person person:list) {
            str.append(person.toStringPersonalCredit()).append("\n");
        }
        str2 = new StringBuilder();
        for (Person person:list2) {
            str2.append(person.toStringPersonalCredit()).append("\n");
        }
        sendMessage.setText(str.toString());
        messageSender.sendMessage(sendMessage);
        sendMessage.setText(str2.toString());
    } else {
        str = new StringBuilder("<b><u>Лучный
ТУП1:</u></b>\n\n");
        for (Person person:list) {
            str.append(person.toStringPersonalCredit()).append("\n");
        }
        sendMessage.setText(str.toString());
    }
    //TODO
}
break;
case "Лучка ТУП1+2" :
    list =
readingFromExcel.readPersonalFromXlsx(Solution.EXCEL_PERSONAL, 1);
    if (list.isEmpty()) {

```

sendMessage.setText("Такого файла нет. Или результаты еще обрабатываются.");

} else {

if (list.size() > 50) {

list2 = list.subList(50, list.size());

list = list.subList(0, 50);

*str = new StringBuilder("<u>Личный
ТУР1+2:</u>\n\n");*

for (Person person : list) {

str.append(person.toStringPersonalCredit()).append("\n");

}

str2 = new StringBuilder();

for (Person person : list2) {

str2.append(person.toStringPersonalCredit()).append("\n");

}

sendMessage.setText(str.toString());

messageSender.sendMessage(sendMessage);

sendMessage.setText(str2.toString());

} else {

*str = new StringBuilder("<u>Личный
ТУР1+2:</u>\n\n");*

for (Person person : list) {

str.append(person.toStringPersonalCredit()).append("\n");

}

sendMessage.setText(str.toString());

}

					ДП.КН 23.519.03.000 ПЗ	Анк.
						79
Зм.	Арк.	№ докум.	Підпис.	Дата		

```

    }
    break;
    case Solution.BUTTON_PERSONAL_FINISH:
        list =
readingFromExcel.readPersonalFromXlsx(Solution.EXCEL_PERSONAL, 2);
        if (list.isEmpty()) {
            sendMessage.setText("Такого файла нет. Или результаты еще
обрабатываются.");
        } else {
            if (list.size() > 50) {
                list2 = list.subList(50, list.size());
                list = list.subList(0, 50);
                str = new StringBuilder("<b><u>Личка
ИТОГ:</u></b>\n\n");
                for (Person person : list) {
                    str.append(person.toStringPersonalCredit()).append("\n");
                }
                str2 = new StringBuilder();
                for (Person person : list2) {
                    str2.append(person.toStringPersonalCredit()).append("\n");
                }
                sendMessage.setText(str.toString());
                messageSender.sendMessage(sendMessage);
                sendMessage.setText(str2.toString());
            } else {
                str = new StringBuilder("<b><u>Личка
ИТОГ:</u></b>\n\n");
                for (Person person : list) {

```

					ДП.КН 23.519.03.000 ПЗ	Анк.
						80
Зм.	Арк.	№ докум.	Підпис.	Дата		

```

        str.append(person.toStringPersonalCredit()).append("\n");
    }
    sendMessage.setText(str.toString());
}
}
break;

case "Коман.Тип1":
    list = readingFromExcel.readTeamFromXlsx(Solution.EXCEL_TEAM,
0);

    if (list.isEmpty()) {
        sendMessage.setText("Такого файла нет. Или результаты еще
обрабатываются.");
    } else {
        str = new StringBuilder("<b><u>Командный
ТУП1:</u></b>\n\n");
        for (Person person:list) {
            str.append(person.toStringTeamResult()).append("\n");
        }
        sendMessage.setText(str.toString());
    }
    break;
case "Коман.Тип1+2" :
    list = readingFromExcel.readTeamFromXlsx(Solution.EXCEL_TEAM,
1);

    if (list.isEmpty()) {
        sendMessage.setText("Такого файла нет. Или результаты еще
обрабатываются.");
    }

```

					ДП.КН 23.519.03.000 ПЗ	Анк.
						81
Зм.	Арк.	№ докум.	Підпис.	Дата		

```

        } else {
            str = new StringBuilder("<b><u>Командный
ТУП1+2:</u></b>\n\n");
            for (Person person:list) {
                str.append(person.toStringTeamResult()).append("\n");
            }
            sendMessage.setText(str.toString());
        }
        break;
case Solution.BUTTON_TEAM_FINISH:
    list = readingFromExcel.readTeamFromXlsx(Solution.EXCEL_TEAM,
2);
    if (list.isEmpty()) {
        sendMessage.setText("Такого файла нет. Или результаты еще
обрабатываются.");
    } else {
        str = new StringBuilder("<b><u>Командный
ИТОГ:</u></b>\n\n");
        for (Person person:list) {
            str.append(person.toStringTeamResult()).append("\n");
        }
        sendMessage.setText(str.toString());
    }
    break;
default:
    str = new StringBuilder(fileWorker.getStringFromFile());
    sendMessage.setText(str.toString());
}

```

```

    } else {
        result = "Виберіть зону";
        switch (text) {
            case "TYP1":
                keyboard.add(InlineKeyboardButton.builder().text("TYP_1 Зона-
A").callbackData("tour1zoneA").build());
                keyboard.add(InlineKeyboardButton.builder().text("TYP_1 Зона-
B").callbackData("tour1zoneB").build());
                keyboard.add(InlineKeyboardButton.builder().text("TYP_1 Зона-
C").callbackData("tour1zoneC").build());

                inlineKeyboardMarkup.setKeyboard(Collections.singletonList(keyboard));
                sendMessage.setReplyMarkup(inlineKeyboardMarkup);
                break;
            case "TYP2":
                keyboard.add(InlineKeyboardButton.builder().text("TYP_2 Зона-
A").callbackData("tour2zoneA").build());
                keyboard.add(InlineKeyboardButton.builder().text("TYP_2 Зона-
B").callbackData("tour2zoneB").build());
                keyboard.add(InlineKeyboardButton.builder().text("TYP_2 Зона-
C").callbackData("tour2zoneC").build());

                inlineKeyboardMarkup.setKeyboard(Collections.singletonList(keyboard));
                sendMessage.setReplyMarkup(inlineKeyboardMarkup);
                break;
            case "TYP3":
                keyboard.add(InlineKeyboardButton.builder().text("TYP_3 Зона-
A").callbackData("tour3zoneA").build());

```

```
        keyboard.add(InlineKeyboardButton.builder().text("TYP_3 Зона-  
B").callbackData("tour3zoneB").build());
```

```
        keyboard.add(InlineKeyboardButton.builder().text("TYP_3 Зона-  
C").callbackData("tour3zoneC").build());
```

```
inlineKeyboardMarkup.setKeyboard(Collections.singletonList(keyboard));
```

```
sendMessage.setReplyMarkup(inlineKeyboardMarkup);
```

```
break;
```

```
default:
```

```
result = fileWorker.getStringFromFile();
```

```
}
```

```
sendMessage.setText(result);
```

```
}
```

```
sendMessage.setParseMode("HTML");
```

```
messageSender.sendMessage(sendMessage);
```

```
}
```

```
}
```

```
// create keyboard on start
```

```
public ReplyKeyboardMarkup setReplyKeyboardMarkup() {
```

```
    var markup = new ReplyKeyboardMarkup();
```

```
    var keyboardRows = new ArrayList<KeyboardRow>();
```

```
    KeyboardRow row1 = new KeyboardRow();
```

```
    KeyboardRow row2 = new KeyboardRow();
```

```
    KeyboardRow row3 = new KeyboardRow();
```

```
    KeyboardRow row4 = new KeyboardRow();
```

```
    row1.add("TYP1");
```

```
    row1.add("TYP2");
```

```
    row1.add("TYP3");
```

```

    row2.add("Личка ТУР1");
    row2.add("Личка ТУР1+2");
    row2.add(Solution.BUTTON_PERSONAL_FINISH);
    row3.add("Коман.Тур1");
    row3.add("Коман.Тур1+2");
    row3.add(Solution.BUTTON_TEAM_FINISH);
    row4.add("Жеребьёвка");
    row4.add(Solution.BUTTON_STATISTIC);
    keyboardRows.add(row1);
    keyboardRows.add(row2);
    keyboardRows.add(row3);
    keyboardRows.add(row4);
    markup.setKeyboard(keyboardRows);
    markup.setResizeKeyboard(true);
    markup.setOneTimeKeyboard(true);

    return markup;
}
}

```

Додаток Б
Реалізація класу обробки CallbackQuery.

```
package com.telegabot.spinning.handlers;

import com.telegabot.spinning.ReadingExcel;
import com.telegabot.spinning.Solution;
import com.telegabot.spinning.messagesender.MessageSender;
import com.telegabot.spinning.objects.Person;
import org.springframework.stereotype.Component;
import org.telegram.telegrambots.meta.api.methods.send.SendMessage;
import org.telegram.telegrambots.meta.api.objects.CallbackQuery;

import java.util.List;

@Component
public class CallbackQueryHandler implements Handler<CallbackQuery> {

    private final MessageSender messageSender;

    public CallbackQueryHandler(MessageSender messageSender) {
        this.messageSender = messageSender;
    }

    @Override
    public void choose(CallbackQuery callbackQuery) {
        SendMessage sendMessage = new SendMessage();

        sendMessage.setChatId(String.valueOf(callbackQuery.getMessage().getChatId()));
```

```

String callbackQueryData = callbackQuery.getData();
String result="";

switch (callbackQueryData) {
    //TYP1 - ЗОНЫ
    case "tour1zoneA":
        result = getDataTourAndZone(Solution.EXCEL_TOUR1, 0);
        break;
    case "tour1zoneB":
        result = getDataTourAndZone(Solution.EXCEL_TOUR1, 1);
        break;
    case "tour1zoneC":
        result = getDataTourAndZone(Solution.EXCEL_TOUR1, 2);
        break;
    //TYP2 - ЗОНЫ
    case "tour2zoneA":
        result = getDataTourAndZone(Solution.EXCEL_TOUR2, 0);
        break;
    case "tour2zoneB":
        result = getDataTourAndZone(Solution.EXCEL_TOUR2, 1);
        break;
    case "tour2zoneC":
        result = getDataTourAndZone(Solution.EXCEL_TOUR2, 2);
        break;
    //TYP3 - ЗОНЫ
    case "tour3zoneA":
        result = getDataTourAndZone(Solution.EXCEL_TOUR3, 0);
        break;

```

```

    case "tour3zoneB":
        result = getDataTourAndZone(Solution.EXCEL_TOUR3, 1);
        break;
    case "tour3zoneC":
        result = getDataTourAndZone(Solution.EXCEL_TOUR3, 2);
        break;

    default: result = "Ой, ошибка...";
}

sendMessage.setReplyToMessageId(callbackQuery.getMessage().getMessageId());
sendMessage.setParseMode("HTML");
System.out.println(result.length());
sendMessage.setText(result);
messageSender.sendMessage(sendMessage);
}

String getDataTourAndZone(String pathToExcelFile, int sheetIndex) {
    List<Person> personList;
    String result;
    if (pathToExcelFile.contains("TYP1.xlsx")){
        if (sheetIndex == 0)
            result = "<b><u>TYP - 1, Зона - A</u></b>";
        else if (sheetIndex == 1)
            result = "<b><u>TYP - 1, Зона - B</u></b>";
        else if (sheetIndex == 2)
            result = "<b><u>TYP - 1, Зона - C</u></b>";
        else result = "Что-то пошло не так...";
    } else if (pathToExcelFile.contains("TYP2.xlsx")) {
        if (sheetIndex == 0)

```

```

        result = "<b><u>ТУР - 2, Зона - А</u></b>";
    else if (sheetIndex == 1)
        result = "<b><u>ТУР - 2, Зона - В</u></b>";
    else if (sheetIndex == 2)
        result = "<b><u>ТУР - 2, Зона - С</u></b>";
    else result = "Що-то пошло не так...";
} else if (pathToExcelFile.contains("ТУР3.xlsx")) {
    if (sheetIndex == 0)
        result = "<b><u>ТУР - 3, Зона - А</u></b>";
    else if (sheetIndex == 1)
        result = "<b><u>ТУР - 3, Зона - В</u></b>";
    else if (sheetIndex == 2)
        result = "<b><u>ТУР - 3, Зона - С</u></b>";
    else result = "Що-то пошло не так...";
} else
    result = "Наверное, результаты еще не готовы...";
personList = new ReadingExcel().readZoneFromXlsx(pathToExcelFile,
sheetIndex);
if (!personList.isEmpty()) {
    for (Person person : personList) {
        result = result + "\n\n" + person.toStringZone();
    }
}
else result = "Наверное, результаты еще не готовы...";

return result;
}
}

```

Додаток В

Реалізація класу з шляхами до класів та деякими константними значеннями змінних.

```
package com.telegabot.spinning;

import org.springframework.stereotype.Component;

@Component

public class Solution {

    //Стартовый текст бота

    public static final String START_TEXT = "Привет. Я - бот, что запущен в
тестовом режиме. " +

        "Надеюсь, что скоро я буду уметь больше... А пока развлекайся!\n\n\n" +

        "<a href='https://ligaspinning.com.ua'>Наш сайт</a>";

    //Пути к файлам туров соревнований. Внутри туров лежат зоны (А, В, С)

    public static final String EXCEL_TOUR1 = "..\\spinning\\src\\docsexcel\\По зонам
ТУР1.xlsx";

    public static final String EXCEL_TOUR2 = "..\\spinning\\src\\docsexcel\\По зонам
ТУР2.xlsx";

    public static final String EXCEL_TOUR3 = "..\\spinning\\src\\docsexcel\\По зонам
ТУР3.xlsx";

    public static final String EXCEL_TOSS =

    "..\\spinning\\src\\docsexcel\\ЖЕПЕБ.xlsx";

    public static final String EXCEL_PERSONAL =

    "..\\spinning\\src\\docsexcel\\Особистий залік.xlsx";

    public static final String EXCEL_TEAM =

    "..\\spinning\\src\\docsexcel\\Командний залік.xlsx";

    //для кнопок
```

					ДП.КН 23.519.03.000 ПЗ	Анк.
Зм.	Арк.	№ докум.	Підпис.	Дата		90

ВІДГУК
дипломний проект
відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола
студента IV курсу групи КН-41

Огньова Ігора
(прізвище та ініціали)

Спеціальність 122 „Комп'ютерні науки”

Керівник ДП Чубей О.О.

Тема: Створення телеграм-боту змагань ловлі хижої риби.

1. Загальна характеристика студента: *студент на достатньому рівні володіє засобами проектування, проявив себе старанним та відповідальним. Робота проводилась із дотриманням усіх відповідних термінів.*
2. Практична або теоретична цінність опрацьованих питань: *робота полягає у проектуванні телеграм-боту, з метою задоволення потреб приватного захоплення та бізнесу. Тема є практичною і значимою на даному етапі розвитку мобільних технологій. Проблематику проаналізовано та здійснено огляд існуючих рішень. Даний проект готовий до застосування за призначенням, оскільки є нескладним у використанні.*
3. Недоліки роботи: *тема дипломного проекту частково відрізняється від теми затвердженої у наказі освітнього закладу; перелік використаних джерел містить посилання на літературу російського видання.*
4. Загальний висновок: *дипломний проект виконано згідно вимог і заслуговує на задовільну оцінку.*

Керівник дипломного проекту Чубей О.О.
(прізвище та ініціали)

РЕЦЕНЗІЯ

на дипломний проєкт
студента відділення комп'ютерних технологій
Галицького фахового коледжу імені В'ячеслава Чорновола
студента IV курсу групи КН-41
Огньова Ігора Ігоровича

Спеціальність 122 „Комп'ютерні науки”

Обсяг дипломного проєкту: 69 стор.
Кількість сторінок записки: 69 стор.

Тема: «Створення телеграм-боту змагань з ловлі хижої риби»

Актуальність теми акцентується на тому, що створення Telegram-бота є не тільки ефективним, але й інноваційним способом створення інтерактивної платформи для отримання різноманітної інформації.

Практична або теоретична цінність опрацьованих питань. Проведене дослідження включає детальний огляд поняття Telegram-бота та процесу його розробки з використанням мови програмування Java. Завдяки цьому боту користувачі мають зручну можливість отримувати результати змагань з ловлі риби, що робить його незамінним інструментом для спілкування та отримання актуальної інформації для рибалок.

Застосовані технології та методики програмування детально розглядаються, зокрема, робота з API Telegram, взаємодія з базою даних, обробка команд користувача та надсилання результатів змагань.

Недоліки роботи: в роботі варто привести діаграму класів розробленого ПЗ. В дипломі варто відобразити скріни програмного засобу на етапі розробки.

Загальний висновок: робота відповідає вимогам до дипломних проєктів освітньо кваліфікаційного рівня «молодший спеціаліст» зі спеціальності 122 «Комп'ютерні науки» та заслуговує оцінку «задовільно».

Рецензент

 Н.Г.Гавришків

« 23 » 06 2023р.

Ім'я користувача:
Василь Кузик

Дата перевірки:
19.06.2023 22:16:50 EEST

Дата звіту:
19.06.2023 22:22:34 EEST

ID перевірки:
1015650463

Тип перевірки:
Doc vs Internet + Library

ID користувача:
100012366

Назва документа: Огньов(1)

Кількість сторінок: 67 Кількість слів: 11892 Кількість символів: 88410 Розмір файлу: 4.49 MB ID файлу: 1015296205

19.2% Схожість

Найбільша схожість: 3.42% з джерелом з Бібліотеки (ID файлу: 1015267429)

18.3% Джерела з Інтернету

636

Сторінка 69

4.72% Джерела з Бібліотеки

13

Сторінка 75

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

73.3% Вилучень

Деякі джерела вилучено автоматично (фільтри вилучення: кількість знайдених слів є меншою за 8 слів та 0%)

Немає вилучених Інтернет-джерел

73.3% Вилученого тексту з Бібліотеки

1

Сторінка 75

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи

5