

НАЦИОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

УКРАЇНИ «Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованої обробки інформації та управління

КУРСОВА РОБОТА

по дисципліні

Об'єктно-орієнтоване програмування

(назва дисципліни)

Варіант № 15

Виконав: студент групи ІІ-82
Обруч Я.В
(Прізвище та ініціали)

Прийняла доцент Іванова Л.М.

Київ – 2020

ЗМІСТ

ЗАВДАННЯ.....	3
1 ОПИС ЗАСТОСУВАННЯ.....	4
АРХІТЕКТУРА ЗАСТОСУВАННЯ.....	4
ДІАГРАМА ВАРІАНТІВ ЗАСТОСУВАННЯ.....	6
ДІАГРАМА КЛАСІВ І ТАБЛИЦЬ.....	7
2 РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ ЗАСТОСУВАННЯ.....	17
2.1 ТЕХНІЧНІ ХАРАКТЕРИСТИКИ	17
2.2 РЕКОМЕНДАЦІЇ ПО ВСТАНОВЛЕННЮ ТА НАЛАШТУВАННЯ...	17
2.3 РЕКОМЕНДАЦІЇ КОРИСТУВАЧЕВІ.....	19
3 ВИХІДНИЙ КОД ЗАСТОСУВАННЯ.....	22
СПИСОК ЛІТЕРАТУРИ.....	23

ЗАВДАННЯ

Написати консольне застосування за архітектурним шаблоном MVC з розподіленням відповідальності між компонентами, яке виконує обробку набору даних згідно з варіантом завдання (додаток А) та забезпечує наступне:

1. Збереження даних у файлі (формат файлу будь-який);
2. Читання даних у пам'ять при запуску застосунку;
3. Збереження даних у той же файл при завершенні роботи застосунку, якщо дані були змінені;
4. Збереження проміжних даних у будь-який файл (користувач вводить ім'я файлу);
5. Інтерактивність з користувачем (мова інтерфейсу має обиратися при запуску застосунку на виконання);
6. Логування подій та помилок в роботі застосунку.

1 ОПИС ЗАСТОСУВАННЯ

1.1 Архітектура застосування

В даній курсовій роботі було використано архітектурний шаблон MVC . Три ключові терміни : Model , View , Controller . Головною ідеєю цього шаблону є модульність . Механізм роботи MVC наступний . Користувач починає працювати на сайті і таким чином відправляє запити до серверу , тобто на Controller .Controller в свою чергу обробляє запити і передає до Model дані для подальших маніпуляцій з ними . Controller виконує роль зв'язку між користувачем та самою системою.Варто зауважити що Model не залежить від View(не знає як дані візуалізувати) і Controller (не має точок взаємодії з користувачем), просто надає доступ до даних і управління ними(тобто бізнес-логіка проекту). Після того як Model виконала маніпуляції з даними , Controller відправляє на View користувача результати роботи Model .

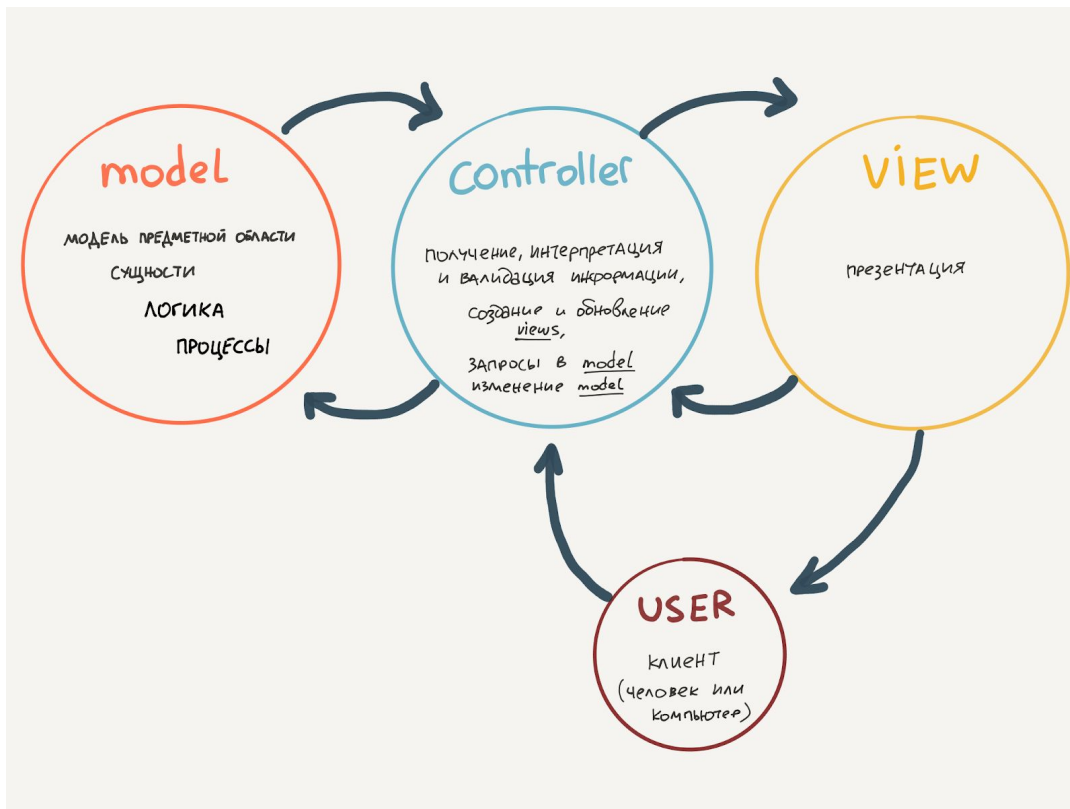


Рис. 1 - Архітектурний шаблон MVC

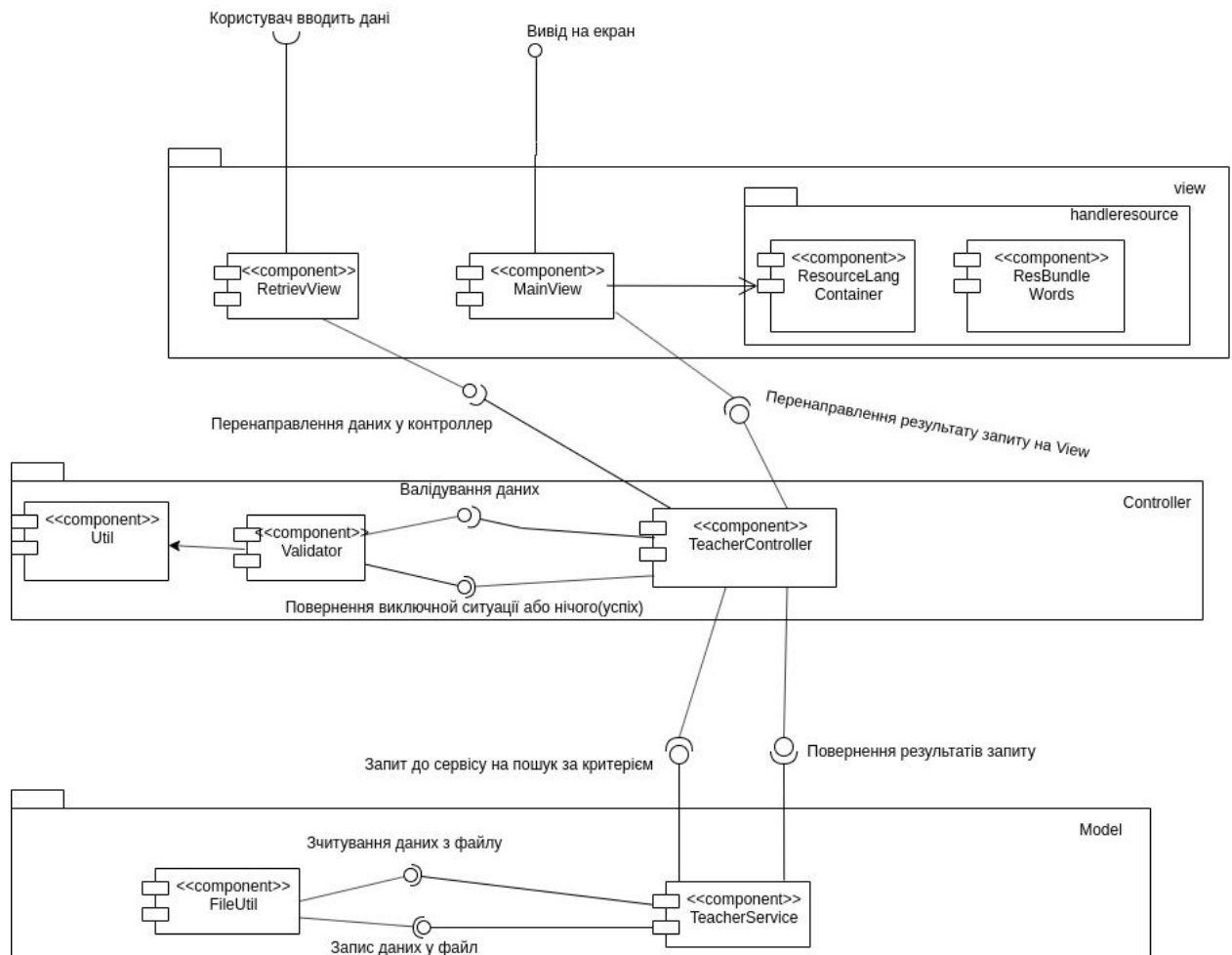


Рис 2 - Діаграма компонентів програми

На даній діаграмі зображено механіку роботи курсового проекту. Першим кроком буде отримання запиту від користувача за допомогою класу RetriewView який знаходиться в пакеті View . Запит направляється до Controller де в свою чергу відразу валідується за допомогою класу Validator на наявність неприпустимих символів . Якщо відповідь користувача пройшла валідацію , то від Controller до Model буде відправлена інформація яка повинна бути оброблена . У разі якщо дані не пройшли валідацію то з валідуючого методу на

Controller будет кинута виключну ситуацію яка в свою чергу буде сигналізувати що користувач ввів некоректні дані .В такому випадку Controller поверне на View помилку та чому вона сталася . В цьому проекті також використовується робота з файлами і перед тим як обробляти інформацію , Model (а саме клас TeacherService) повинен проініціалізувати файл з інформацією , перевівши його у масив об'єктів для зручної роботи . Клас TeacherService є основним в бізнес-логіці курсового проекту . Він виконує пошук викладачів за певними критеріями . Після того як Model провів обробку запиту , від Model до Controller відправляються результати виконання . В свою чергу Controller відправляю View(а саме класу MainView) результати виконання Model . Клас MainView має методи для форматowanego виводу на екран ,які і будуть виводити кінцевий результат запиту користувача .

1.2 Діаграма варіантів застосування

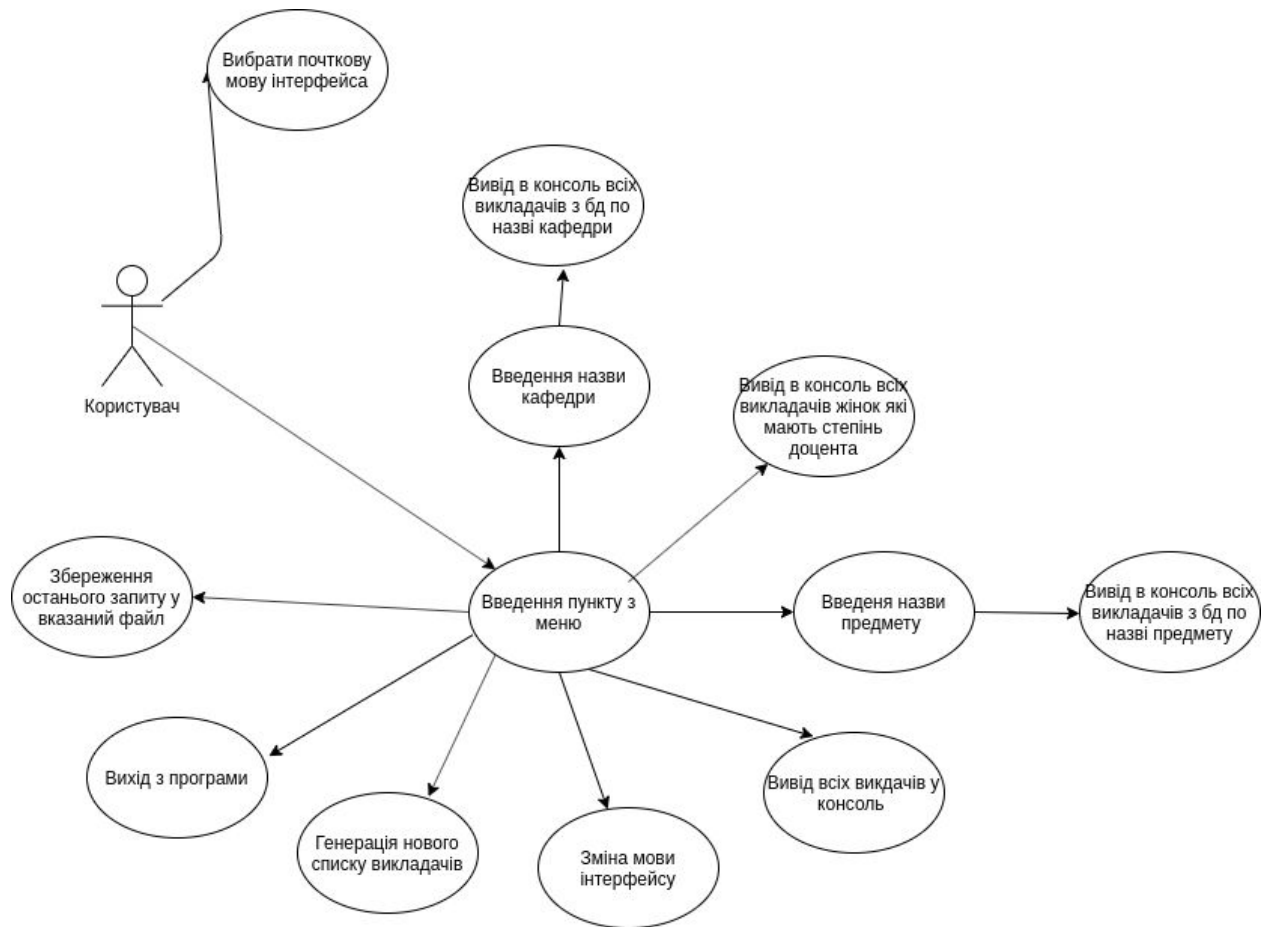


Рис. 3 - Use case diagram

Для візуалізації узагальнення опису предметної області нашого програмного забезпечення, представлена діаграма прецедентів системи (рис 2), яка також має назву “use case diagram” – діаграма варіантів використання програми. На діаграмі зображено відношення між користувачем та прецедентами (діями, які вони можуть зробити, використовуючи наше програмне забезпечення).

1.3 Діаграма класів і таблиця

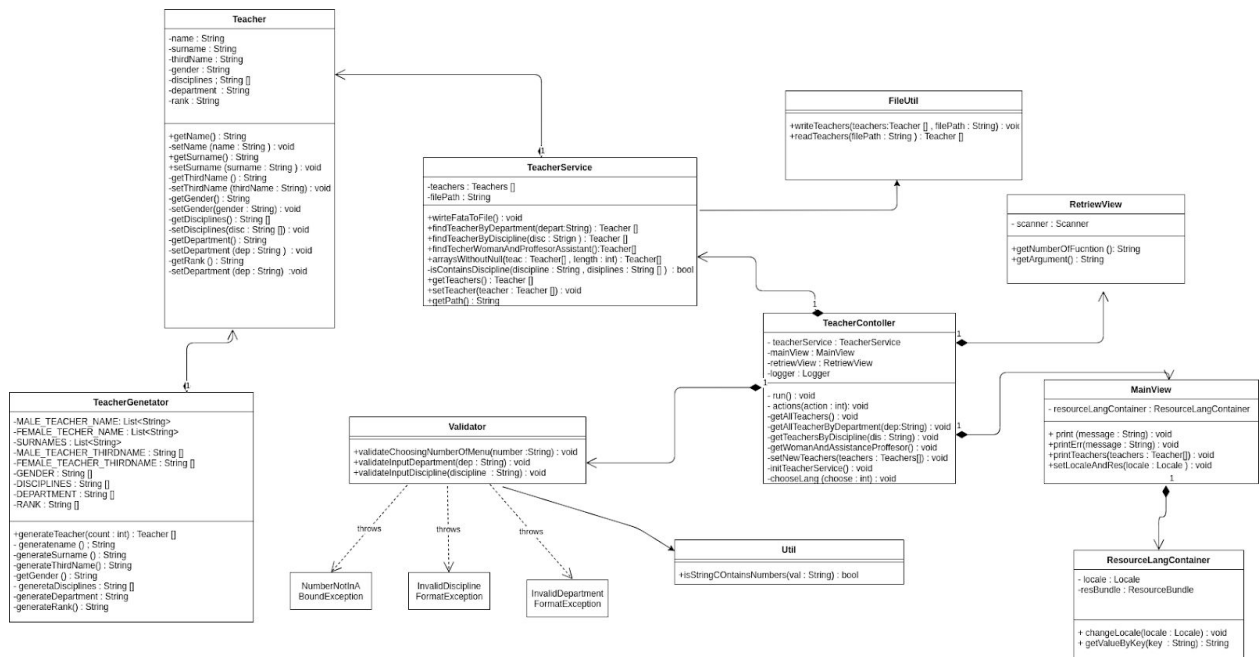


Рис. 4 - Class diagram

Ми на цій діаграмі можемо бачити архітектуру курсового проекту . Оскільки було обрано MVC , то логічним буде розділення всіх компонентів на 3 групи : Controller , Model , View . Також в проекті є допоміжні пакети : FileUtil , Validator , ResourceLangContainer , Util.

В пакеті controller знаходиться клас TeacherController який є головним в управлінні нашим застосунком . Він також містить такі класи як : RetriewView,TeacherService , Validator , MainView .Клас TeacherService має роль БД , при ініціалізації екземпляра цього класу , в конструкторі автоматично читається файл з серіалізованими даними викладачів та вся зчитана інформація конвертується в масив об'єктів(викладачів) . В разі якщо файл не було знайдено або пошкоджено , то користувачеві виводить помилку. Також цей

клас відповідає за пошук по критеріям . Користувач обирає за яким критерієм він хоче зробити вибірку викладачів та вводить ключове слово для пошуку .Клас Validator має статичні методи , які перевіряють валідність введених даних користувачем . В разі якщо користувач ввів невалідне значення , на бік TeacherContoller буде проброшено помилку і контроллер в блоці try catch обробе її та пропонує користувачеві подальші дії для використання застосунку. В пакеті view знаходяться класи MainView , RetriewView та інший пакет який відповідає за мову інтерфейсу - resourcehandler .За допомогою класу RetriewView отримуємо інформацію від користувача .Клас MainView відповідає за формат виводу повідомлень у консоль та відображення мови інтерфейса . Клас ResourceLangContainer відповідає за поточну мову інтерфейсу . В пакеті util знаходяться допоміжні класи такі як Utils , FileUtils і TeacherGenerator . FileUtils є класом що працює з файлами ,який має статичні методи для запису даних в файл та зчитування даних з файлу за вказаним шляхом . Utils має всього один допоміжний метод для валідації даних . Клас TeacherGenerator створений для автоматичного(рандомного) генерування викладачів .

Табл. 1.3.1 - Опис класів застосування

# п/ п	Назва класу	Назва функції	Призначення функції	Опис вхідних параметрів	Опис вихідних параметр ів
1	MainView	print	Виводи в консоль текстове повідомлення користувачеві	Повідомлен ня типу String	void
2	MainView	printErr	Виводить в консоль повідомлення-помилку(Повідомлен ня-помилка	void

			текст червоним кольором)	типу String	
3	MainView	printTeachers	Виводить користувачеві всі данні про викладачів	Массив викладачів	void
4	Retriew View	getArgument	Отримати відповідь користувача	-	String
5	Teacher Service	findTeachers ByDepartment	Функція призначена для того щоб знайти всіх викладачів за конкретною кафедрою яку ввів користувач	Назва кафедри типу String	Массив викладачів Teacher[]
6	Teacher Service	findTeachers ByDiscipline	Функція призначена для того щоб знайти всіх викладачів за конкретним предметом викладання який ввів користувач	Назва предмету типу String	Массив викладачів Teacher[]
7	Teacher Service	findTeachers WomansAnd AssistanceProfessor	Функція призначена для того щоб знайти всіх викладачів жінок які також є доцентами	-	Массив викладачів Teacher[]
8	Teacher Service	writeDataTo File	Функція призначена для того щоб записати поточний список всіх викладачів в текстовий файл для подальшого використання	-	void

9	Validator	validateChoosingNumberOfMenu	Функція призначена для того щоб провалідувати дані які ввів користувач для обрання пункту в меню.	Пункт меню типу String	void
10	Validator	ValidateInputDepartment	Функція призначення для того щоб провалідувати введену назву кафедри на вміст цифрових значень	Назва кафедри типу String	void
11	Validator	validateInputDiscipline	Функція призначення для того щоб провалідувати введену назву предмету на вміст цифрових значень	Назва предмети типу String	void
12	Teacher Controller	run	Ця функція є головною(з неї починається запуск програми та всі процеси які пов'язані з введенням виведенням , валідування даних)	-	void
13	Teacher Controller	actions	Ця функція виступає в ролі мапінгу , тобто слугує для запуску конкретних функцій по введеним даним від користувача	Номер пункту в меню типу int	void
14	Teacher Controller	getAllTeachers	Функція виводить всіх викладачів які знаходяться в поточному списку викладачів у сервісі	-	void

15	Teacher Controller	getTeachersByDepartment	Функція призначена для того щоб провалідувати введену кафедру та в разі якщо дані не пройдуть валідацію , функція знову викличе саму себе для повторного введення . Якщо дані пройшли валідацію , то знаходить всіх викладачів за певною кафедрою	-	void
16	Teacher Controller	getTeachersByDiscipline	Функція призначена для того щоб провалідувати введену дисципліну та в разі якщо дані не пройдуть валідацію , функція знову викличе саму себе для повторного введення . Якщо дані пройшли валідацію , то знаходить всіх викладачів за певною кафедрою	-	void
17	Teachers Controller	getWomanAndAssProfessor	Функція виводить в консоль користувачу всіх викладачів жінок які також мають степінь доцента.	-	void
18	Teacher Controller	chooseLang	Функція приймає на вхід назву мови , яку вибрав користувач для того щоб змінити мову інтерфейсу	Назва мови типу String	void

19	ResourceLangContainer	changeLocale	Функція змінює значення Locale та встановлює новий Bundle , що призводить до змінення мови інтерфейсу	Locale	void
20	ResourceLangContainer	getValueByKey	Функція повертає значення із поточного Bundle за ключем	Значення ключа типу String	String
21	TeacherController	saveLastQueryToFile	Функція зберігає останній запит до вказаного користувачем файлу	-	void
22	Util	isStringHasNumber	Функція повертає true якщо слово містить числові значення , інакше false	String	boolean
23	FileUtil	writeTeachers	Функція серіалізує вхідний масив викладачів та записує результат до вказаного файлу в параметрах	Teacher[], шлях до файлу типу String	void
24	FileUtil	readTeachers	Функція десиріалізує з файлу масив викладачів який знаходиться за шляхом який вказаний в параметрах до функції	шлях до файлу типу String	Teacher []
25	TeacherGenerator	generateName	Функція рандомно вибирає ім'я з масиву імен	-	String

26	Teacher Generator	generateSurname	Функція рандомно вибирає фамілію з масиву фамілій	-	String
27	Teacher Generator	generateThirdName	Функція рандомно вибирає по батькові з масиву	-	String
28	Teacher Generator	getGender	Функція в залежності від ім'я вибирає стать	-	String
29	Teacher Generator	generateDisciplines	Функція рандомно генерує кількість дисциплін і також самі дисципліни викладача	-	String[]
30	Teacher Generator	generateDepartment	Функція рандомним шляхом генерує кафедру викладача з масиву кафедр	-	String
31	Teacher Generator	generateRank	Функція рандомно генерує звання викладача із масиву звань	-	String
32	Teacher	equals	Функція для порівняння викладачів за значеннями	Object teacher	boolean
33	Teacher	hashCode	Функція для генерації хеш коду	-	int
34	Teacher	toString	Функція для форматowanego виводу Teacher	-	String
35	Teacher	getName	Функція повертає name поточного екземпляра класа Teacher	-	String
36	Teacher	setName	Функція встановлює поточному екземпляру класа Teacher поле name	-	void
37	Teacher	getSurname	Функція повертає surname поточного екземпляра класа	-	String

			Teacher		
38	Teacher	setSurname	Функція встановлює поточному екземпляру класа Teacher поле surname	-	void
39	Teacher	getThirdName	Функція повертає thirdName поточного екземпляра класа Teacher	-	String
40	Teacher	setThirdName	Функція встановлює поточному екземпляру класа Teacher поле thirdName	-	void
41	Teacher	getGender	Функція повертає gender поточного екземпляра класа Teacher	-	String
42	Teacher	setGender	Функція встановлює поточному екземпляру класа Teacher поле gender	-	void
43	Teacher	getDisciplines	Функція повертає disciplines поточного екземпляра класа Teacher	-	String []
44	Teacher	setDisciplines	Функція встановлює поточному екземпляру класа Teacher поле disciplines	-	void
45	Teacher	getDepartment	Функція повертає department поточного екземпляра класа Teacher	-	String

46	Teacher	setDepartment	Функція встановлює поточному екземпляру класа Teacher поле department	-	void
47	Teacher	setRank	Функція встановлює поточному екземпляру класа Teacher поле rank	-	void
48	Teacher	getRank	Функція повертає rank поточного екземпляра класа Teacher	-	void

2 РЕКОМЕНДАЦІЇ ЩОДО ВИКОРИСТАННЯ ЗАСТОСУВАННЯ

2.1 Технічні характеристики

ПЗ яке було розроблено може використовуватися на ПК з такими мінімальними характеристиками :

- CPU - 2х ядерний (4 потоки) Intel Core i3 3-го покоління (i3-3220)
- RAM - 512 MB
- Відеокарта - інтегрована в процесор
- Пам'ять - HDD 256 GB
- Операційна система - Linux Ubuntu 18.04
- Монітор - будь-який
- Мишка - будь-яка
- Клавіатура - будь-яка
- Браузер - Chrome , Opera , FireFox

2.2 Рекомендації по встановленню та налаштуванню

Ця версія встановлення та налаштування адаптована під Linux Ubuntu 18.04 . Для встановлення програмного забезпечення необхідно мати JDK на вашому ПК . Що б перевірити потрібно :

```
yarik@TopLaptop:~$ java -version
openjdk version "11.0.7" 2020-04-14
OpenJDK Runtime Environment (build 11.0.7+10-post-Ubuntu-2ubuntu218.04)
OpenJDK 64-Bit Server VM (build 11.0.7+10-post-Ubuntu-2ubuntu218.04, mixed mode,
sharing)
```

Рис. 5 - Перевірка наявності JDK

Якщо у вас є таких напис , то на вашому ПК встановлено все для запуску програми .Інакше вам потрібно перейти за посиланням та встановити JDK згідно з вашою ОС :

<https://www.oracle.com/java/technologies/javase-jdk14-downloads.html>

Наступним кроком буде скачування .jar файлу з репозиторія розробника ПЗ . Для того щоб скачати архів вам потрібно :

1. Перейти за посиланням <https://github.com/neopite/JavaJarProject>

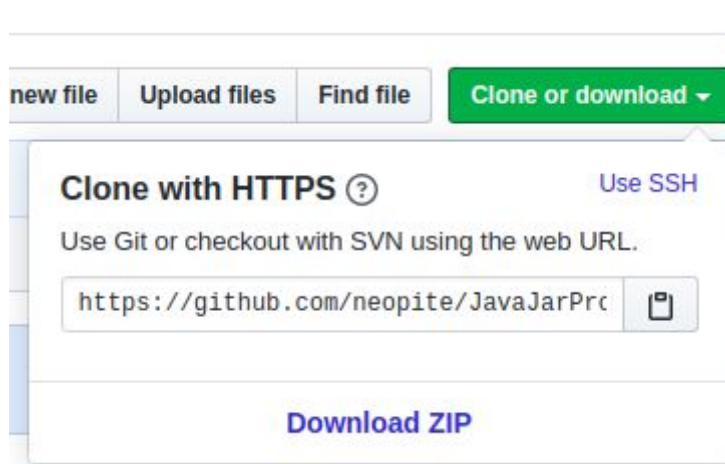


Рис. 6 - Позначення для скачування

2. Натиснути на “Download ZIP”
3. Створіть окрему папку в яку ви будете розархівувати скачаний ZIP архив.
4. Розархівувати

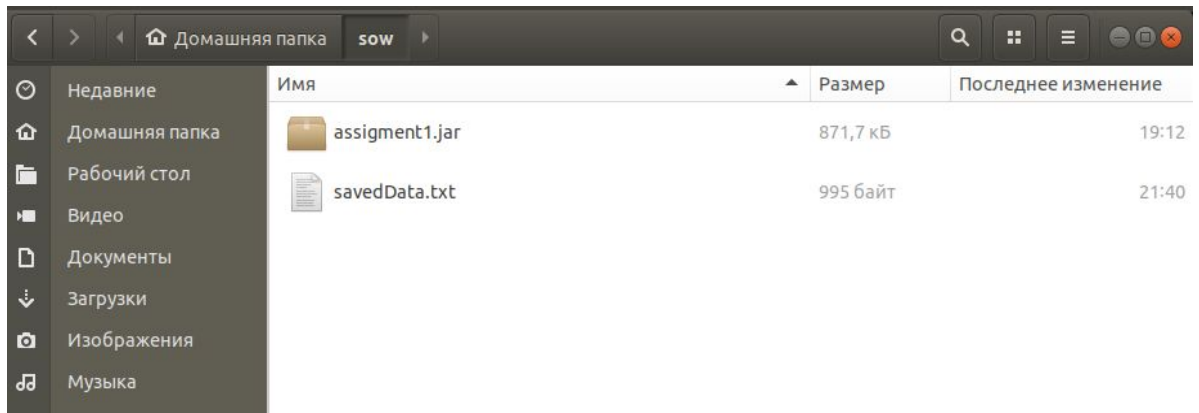


Рис. 7 - Приклад коректної установки ПЗ

5. Зайдіть в папку з .jar файлом через термінал за допомогою команди `cd`
6. Потрібно надати .jar права на виконання . Це можна зробити за допомогою програмної утиліти `chmod` для Linux . Для того щоб надати файлу права , потрібно прописати в консолі `chmod +x assignment.jar`

```
yarik@TopLaptop:~/Загрузки/Copy.zip (Unzipped Files)/Copy/assignment1/out/artifacts/assignment1_jar$ chmod +x assignment1.jar
yarik@TopLaptop:~/Загрузки/Copy.zip (Unzipped Files)/Copy/assignment1/out/artifacts/assignment1_jar$
```

Рис. 8 - Надання файлу прав на виконання

2.3 Рекомендації користувачеві

Після всіх дій можна запускати програмою за допомогою команди :

“`sudo java -jar assignment1.jar`”.

```
yarik@TopLaptop:~/Загрузки/Copy.zip (Unzipped Files)/Copy/assignment1/out/artifacts/assignment1_jar$ sudo java -jar assignment1.jar
```

Рис. 9 - Запуск програми через консоль

Після виконання попередньої команди у вас повинно з'явитися меню з мовами :

```
yarik@topLaptop:~/Зарпубки/copy.zip (Unzipped Files)/Copy/assignment1/out/artifacts/assignment1_jar$ sudo java -jar assignment1.jar
Change your language (type a number of option):
1. English (default).
2. Russian.
3. Ukrainian language.
3
1 : Вивести всіх викладачів
2 : Знайти виклада за кафедрою
3 : Знайти викладача за предметом
4 : Знайти викладача жіночої статі яка є також доцентом
5 : Вихід з програми
6 : Створити новий масив викладачів
7 : Змінити мову інтерфейсу
8 : Зберегти результати останнього запиту в вказаний файл
3
Введіть предмет
Math


| Ім'я      | Прізвище  | По-батькові | Стать | Предмети                | Кафедра | Звання          |
|-----------|-----------|-------------|-------|-------------------------|---------|-----------------|
| Yana      | Kolesnik  | Petrivna    | Woman | [Deutch, Math]          | AUTS    | Professor       |
| Stanislav | Kolesnik  | Petrovich   | Man   | [Deutch, Math]          | ASOIU   | Professor       |
| Marya     | Knish     | Bogdanivna  | Woman | [English, Deutch, Math] | VT      | Senior Teacheer |
| Ivan      | Stetcenko | Petrovich   | Man   | [Deutch, Deutch, Math]  | AUTS    | Senior Teacheer |


1 : Вивести всіх викладачів
2 : Знайти виклада за кафедрою
3 : Знайти викладача за предметом
4 : Знайти викладача жіночої статі яка є також доцентом
5 : Вихід з програми
6 : Створити новий масив викладачів
7 : Змінити мову інтерфейсу
8 : Зберегти результати останнього запиту в вказаний файл
```

Рис. 10 - Коректна робота програми

Також в ході роботи з програмою можуть виникнути виключні ситуації,нижче всі вони будут описані :

1. FileForReadingNotFound . Ця виключна ситуація виникає коли користувач забув завантажити додатковий файл з назвою “savedData.txt”.Якщо ви забули його завантажити , ви просто можете створити в одній директорії разом з assignment.jar текстовый файл з назвою “savedData.txt”.
2. InvalidDisciplineFormat . Виникає коли користувач введе значення яке не проходить валідацію (назва дисципліни має цифрові символи). Ви можете заново ввести назву дисципліни після некоректного вводу.
3. InvalidDepartmentFormat . Виникають в тій же ситуації що і InvalidDisciplineFormat

4. `NumberNotInBound` . Виникає коли користувач під час того як знаходиться в головному меню , ввів некоректні дані (це можуть бути букви та номери які не входять в діапазон допустимих значень пунктів меню)
5. `DataCannotBeReadFromFile` . Виникає коли файл пошкоджено та прочитати вміст файлу не є можливим .
6. `NotFoundPathFileForSave` . Ця виключна ситуація виникає коли користувач хоче самостійно ввести шлях до файлу щоб зберегти останній виконаний запит . Для того щоб уникнути цього , ви повинні скопіювати відносний шлях файлу та вставити у консоль.

3 ВИХІДНИЙ КОД ЗАСТОСУВАННЯ

Переглянути код застосування можна на платформі GitHub посиланням :

<https://github.com/neopite/JavaTrack/tree/master/src/main/java/com/company/lab5>

4 СПИСОК ЛІТЕРАТУРИ

1. Архітектурний шаблон MVC :

<https://habr.com/ru/post/181772/>

2. Лямбда вирази в Java 8 :

<https://habr.com/ru/post/224593/>

3. Документація по log4 :

<https://logging.apache.org/log4j/1.2/>

4. Офіційна документаціяOracle:

<https://docs.oracle.com/javase/8/docs/api/index.html>