**Міністерство освіти і науки України**

**Національний технічний університет**

**«Дніпровська політехніка»**



**Звіт лабораторних робіт**

**з дисципліни «Програмування в середовищі Java»**

**Виконала:**

студентка гр. 125-20-1

Середняк Катерина

Олександрівна

**Прийняв:**

доцент каф. Мінєєв О.С.

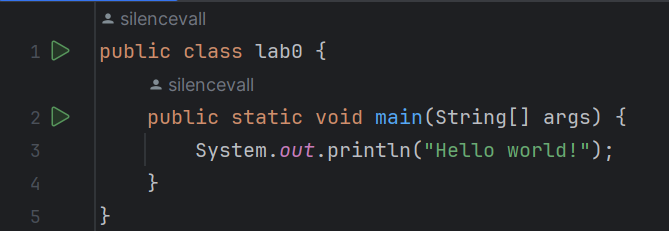
**Дніпро**

**2024**

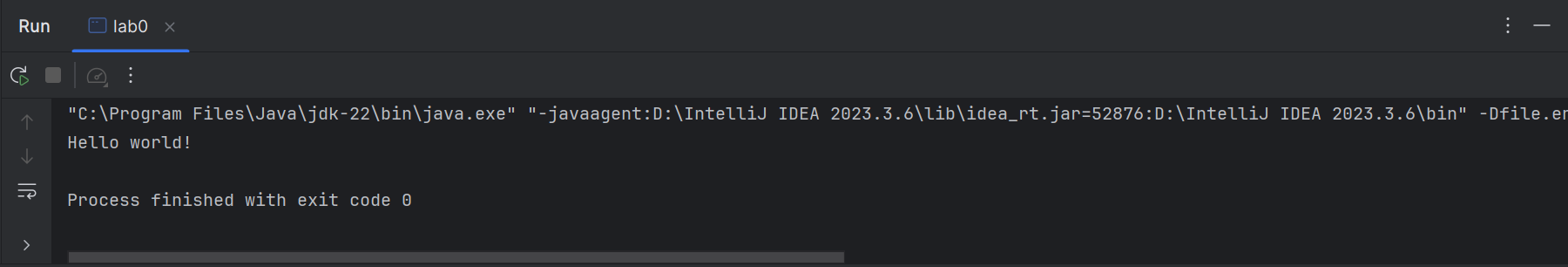
**Лабораторна робота номер 0.** Hello world

Встановити Intellij Idea та Java jdk останньої версії. Cтворити maven проект та розробити в цьому проекті типову програму Hello world.  Програма повинна видавати на екран напис Hello world та закінчувати свою роботу.

**Код:**



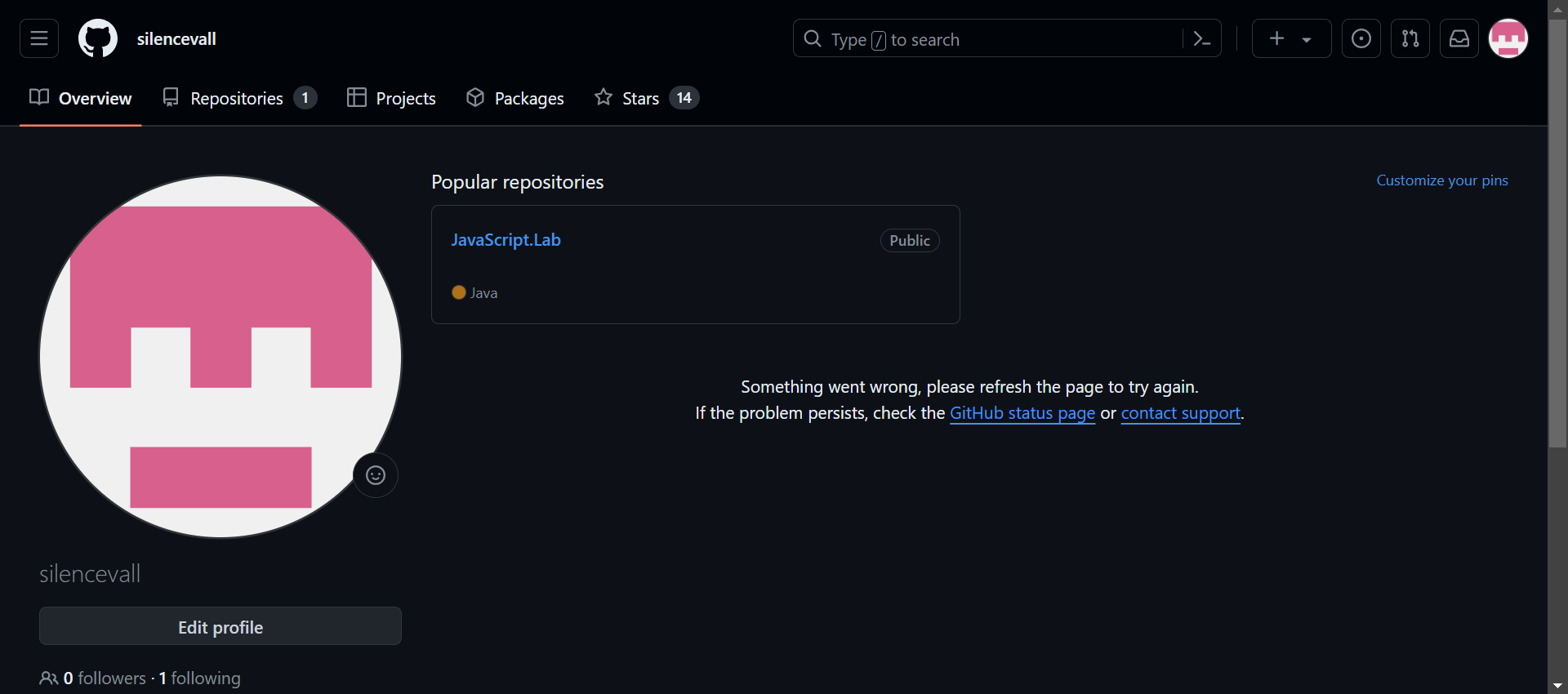
**Результат виконання коду:**

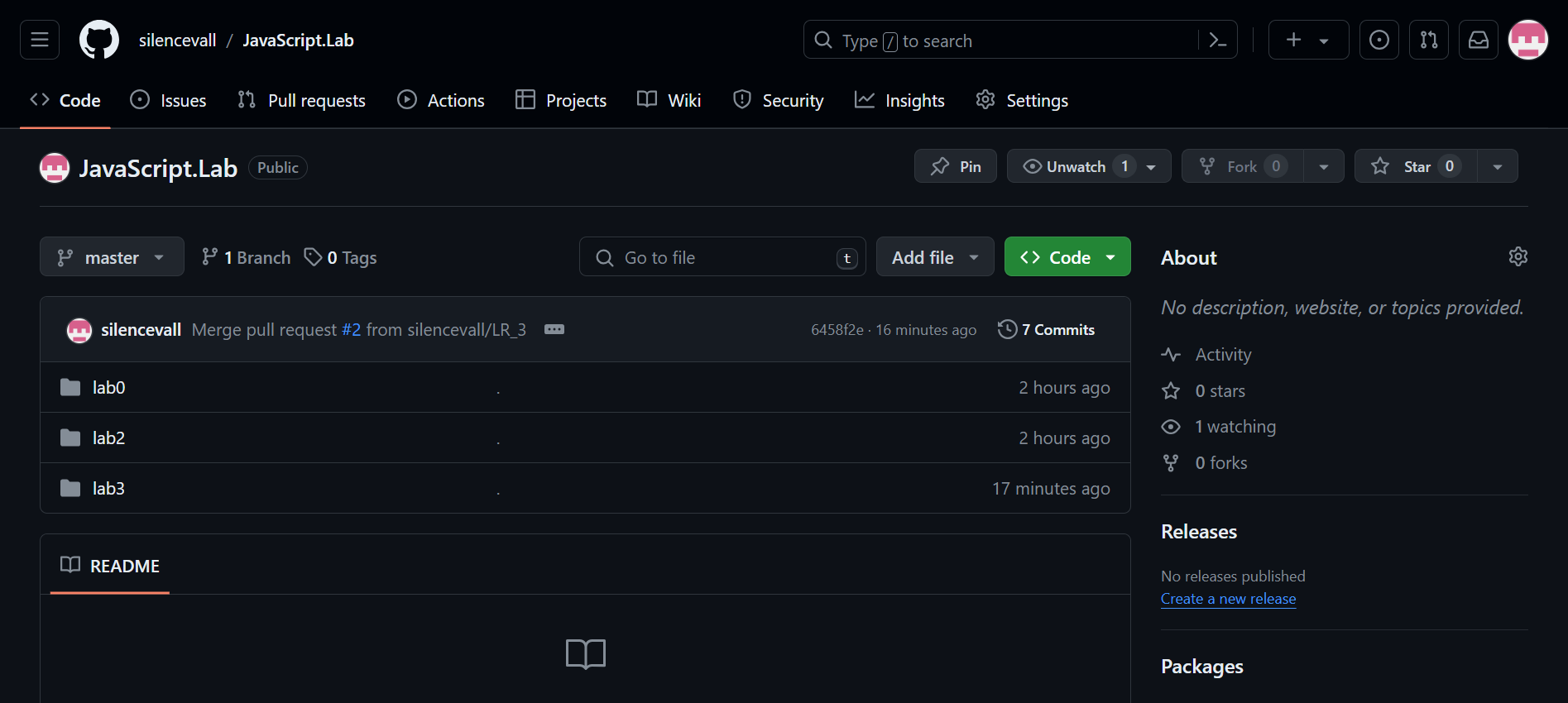


**Лабораторна робота номер 1.** CVS. GIT

Під'єднати до intellij Idea систему CVS. А саме GIT. Створити аккаунт в хмарному середовищі github, під'єднати свій проект в intellij Idea до свого аккаунту github та завантажити нульову лабораторну роботу на github аккаунт. Кожну нову лабораторну роботу робити в окремій гілці(з іменем лабораторної наприклад «LR\_3») а потім після того як її написали мержити гілку до мастера.

Обов'язково перевірте, щоб ваш проект лабораторних робіт на GitHub мав вільний доступ усіх бажаючих до вашого коду. Обмеження доступу до ваших лабораторних робіт буде розцінено, як помилка при виконанні лабораторної роботи.

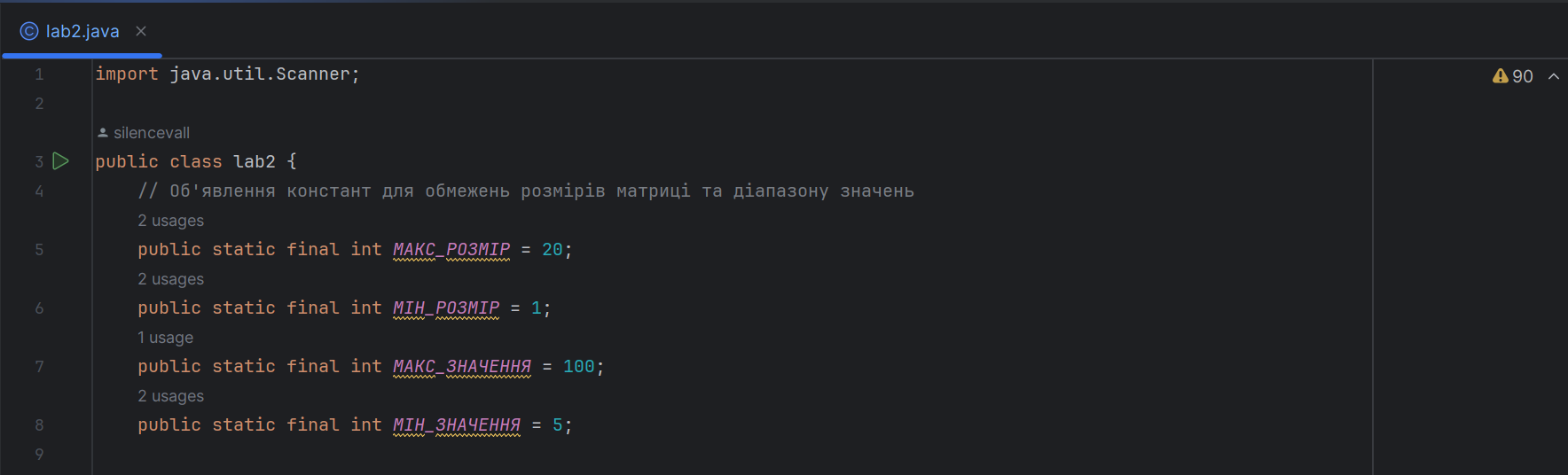


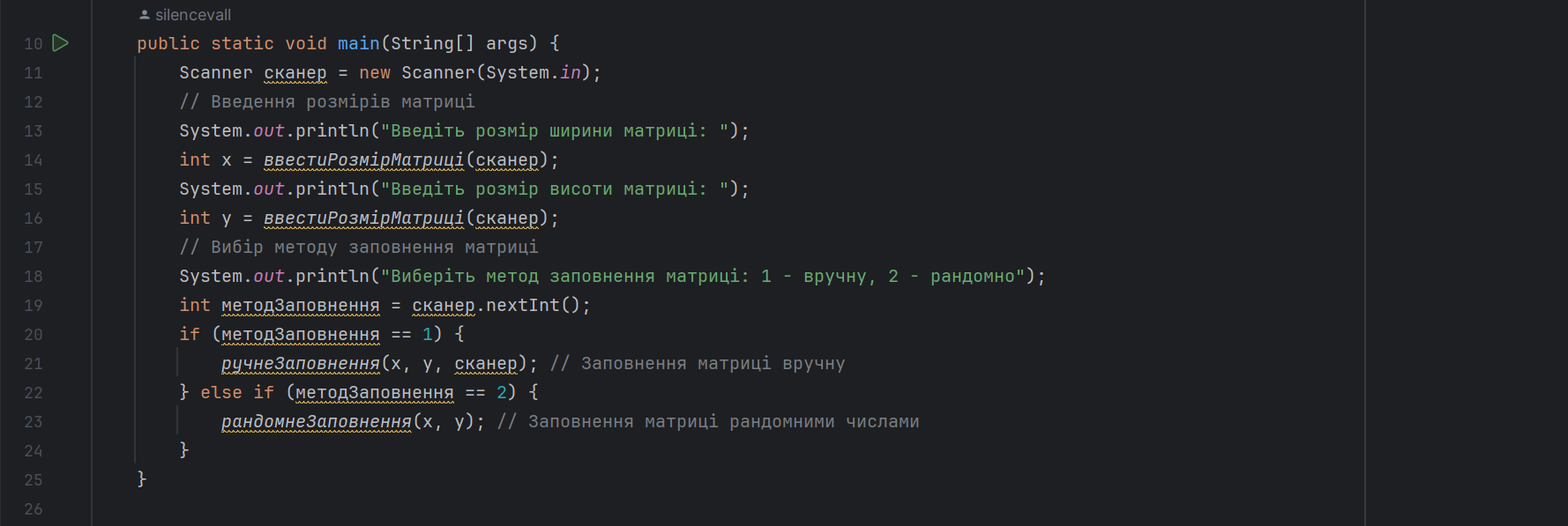


**Лабораторна робота номер 2.** Основи.

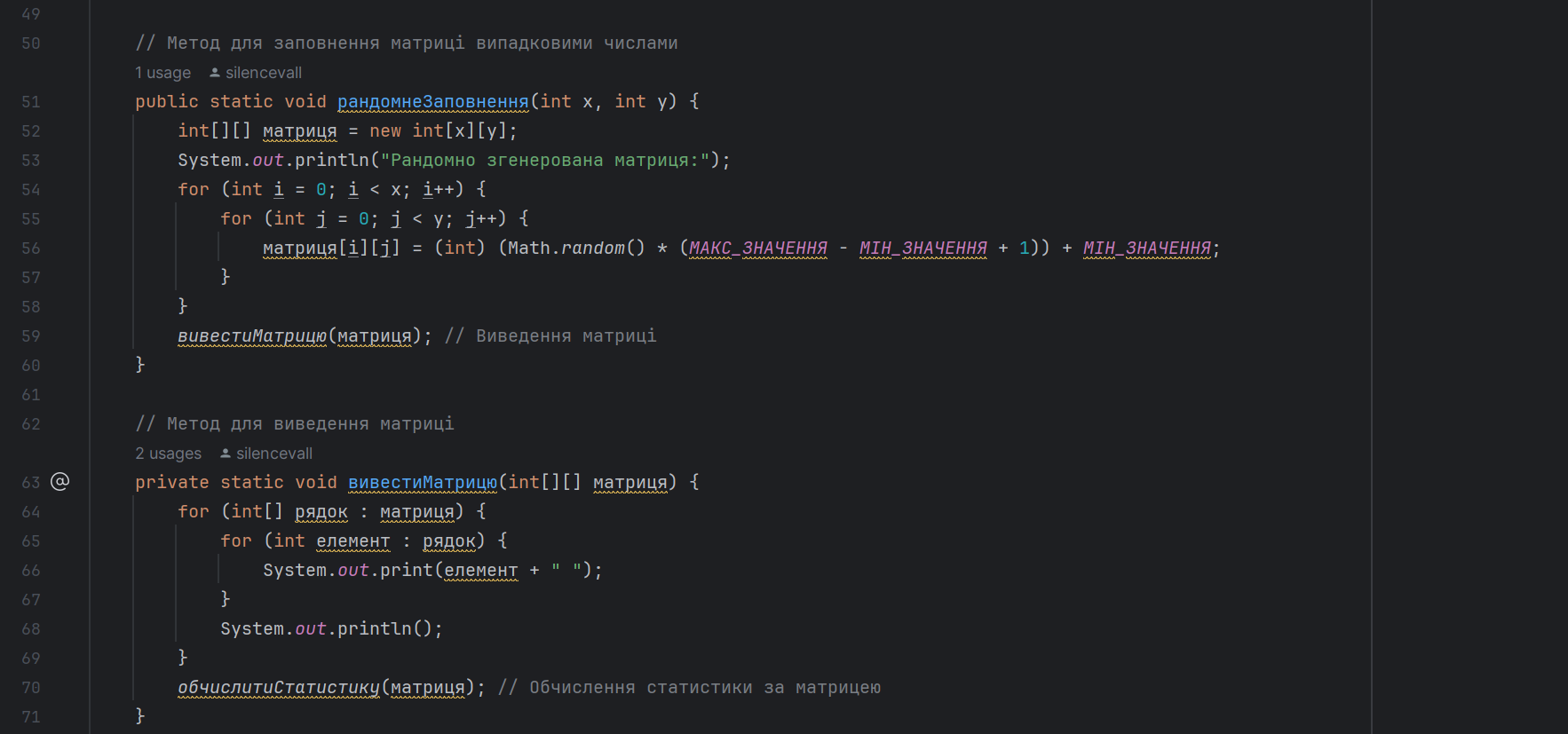
Розробити програму, що дозволить вам створити, як з клавіатури так і рандомно матрицю цілих чисел типу int заданої ширини та висоти(ввести з клавіатури), але не більше 20 на 20. Створити можливість пошуку в цій матриці мінімального і максимального елементу та розрахунок середнього арифметичного. Програма може бути написана в одному класі, обов'язково розбиття на методи. Обов'язкове використання клавіатури, під час вибору ручного чи рандомного створення матриці. Створення системи зчитування з клавіатури зробити будь-яким способом, наприклад завдяки класу Scanner. Scanner являє собою найпростішу систему сканування клавіатури. Діапазон рандомних чисел для створення елементів матриці повинен зберігатись в спеціальних константах. Як завдання підвищеної складності додати розрахунок середнього геометричного елементів матриці.

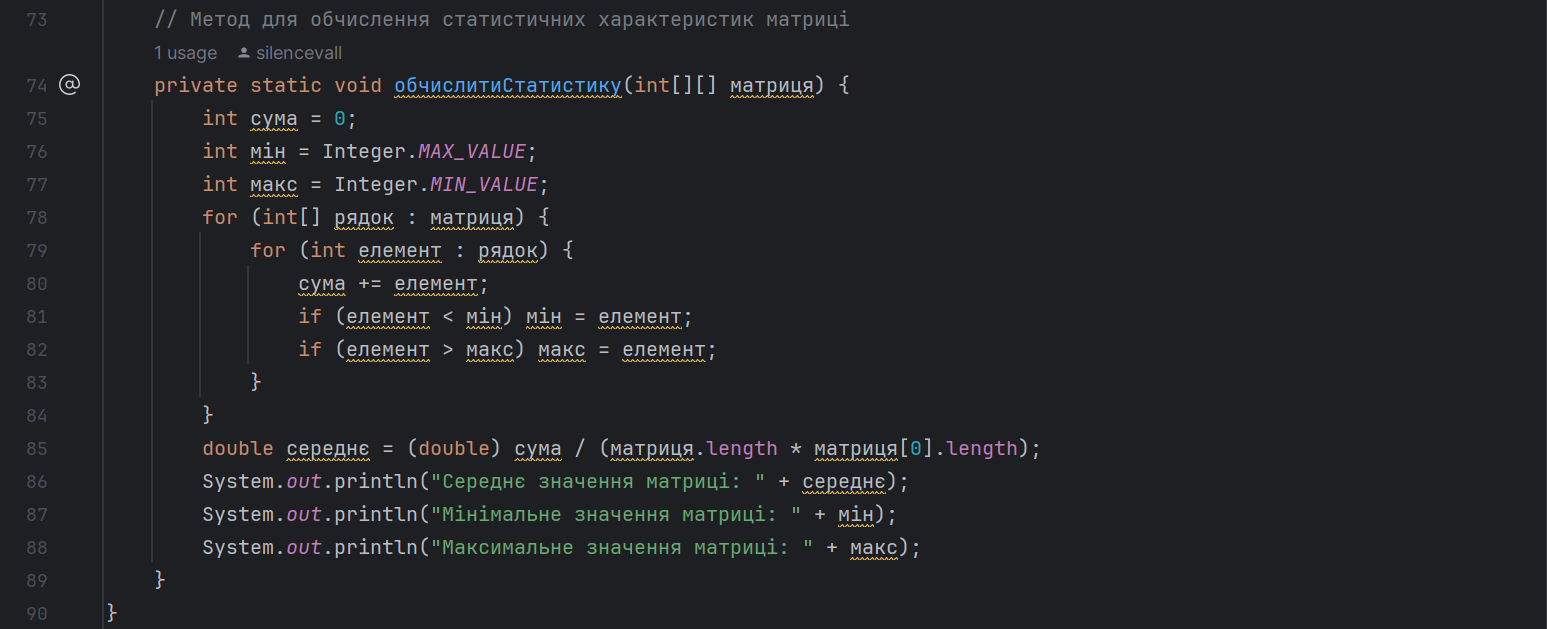
**Код:**





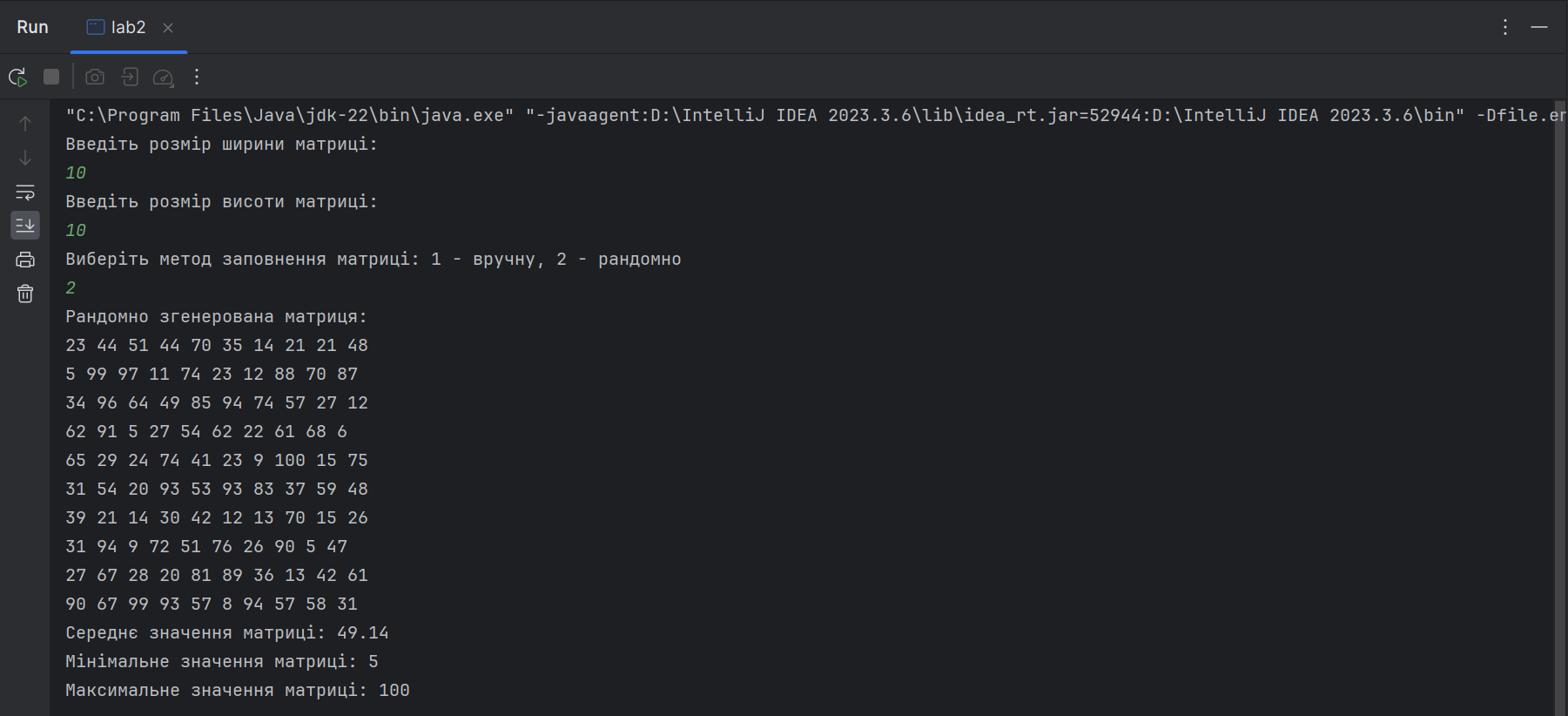




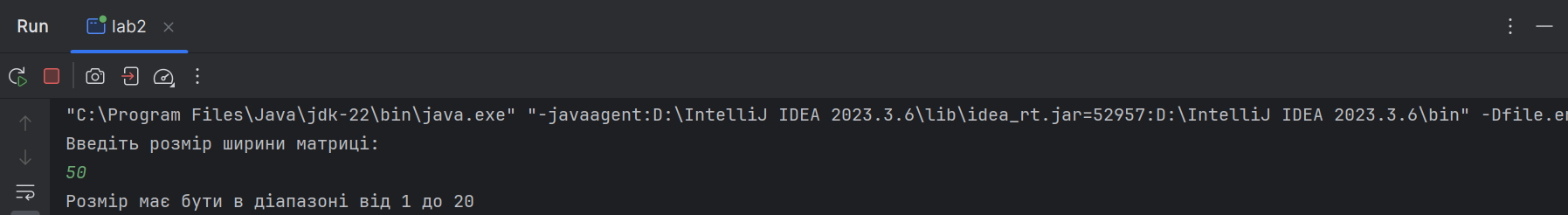


**Результат виконання коду:**

1. **Розмір відповідає нормі:**



1. **Розмір не відповідає нормі:**



**Лабораторна робота номер 3.** ООП.

Створити програму що буде створювати та обробляти комплексний об'єкт під назвою університет(university). Програма повинна складатися з трьох частин: модель вид та контролер згідно з парадигмою mvc (Model View Controller). Кожній з цих груп повинна відповідати package з відповідною назвою. В моделі повинні знаходитись усі класи що відповідають за структурні підрозділи університету. Серед них: університет, факультет, кафедра, група, студент, людина (Human). Усі вони повинні містити назву типу string та голову типу Human. Студент також повинен бути породжений від Human. Human повинен мати поля ім'я, прізвище, по-батькові та стать. Усі поля повинні бути строковими окрім поля стать. Стать повинна використовувати спеціальний enum типу Sex(стать). В цій лабораторній роботі група View Нам не потрібна.

Що стосується групи контролер (controller) то вона повинна містити менеджери що дозволяють нам створити відповідні підрозділи наприклад StudentCreator, FacultyCreator, GroupCreator та інші, кожен з яких повинен використовувати можливості нижчого за рівнем створювача. Програма повинна також містити клас Run, в якому буде знаходитись точка входу та методи, що повинні дати можливість створити університет. Процес створення університету повинен бути зроблений в методі createTypycalUniversity.

В програмі активно рекомендується використовувати абстрактні класи та інтерфейси

**Код:**

1. Папка ModuleLab3:

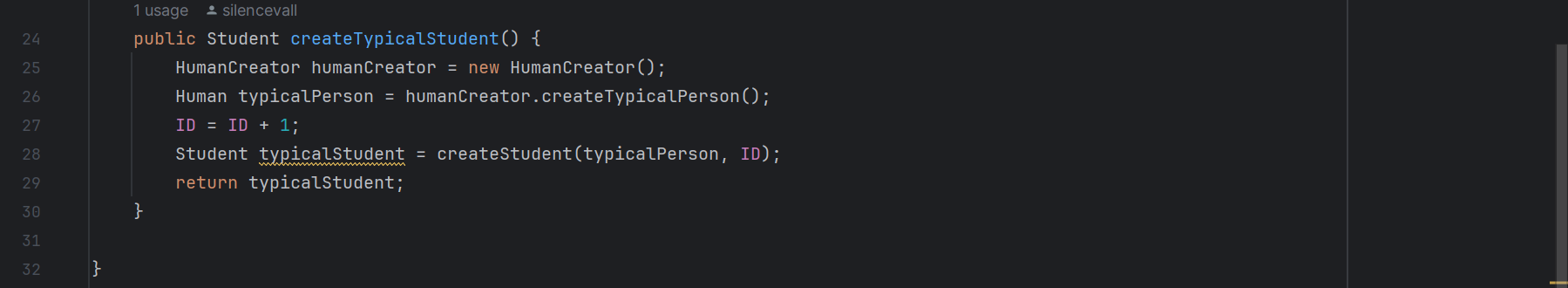
UniversityCreator:





StudentCreator:





HumanCreator:



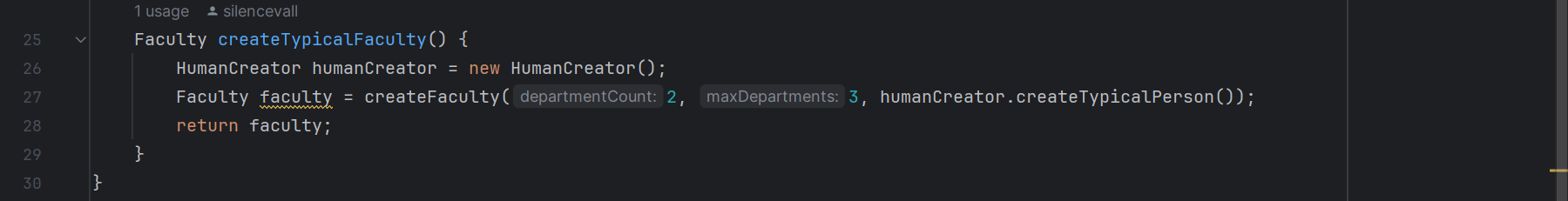
GroupCreator:





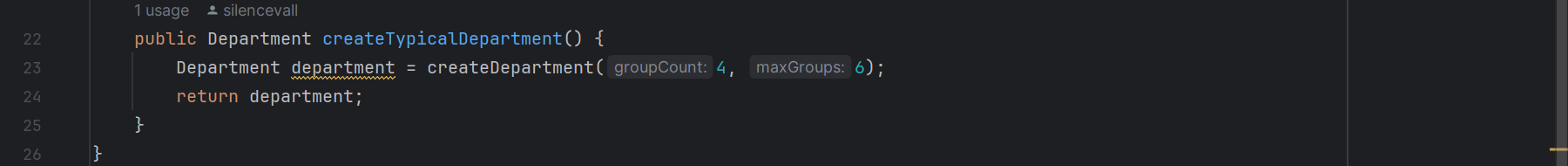
FacultyCreator:





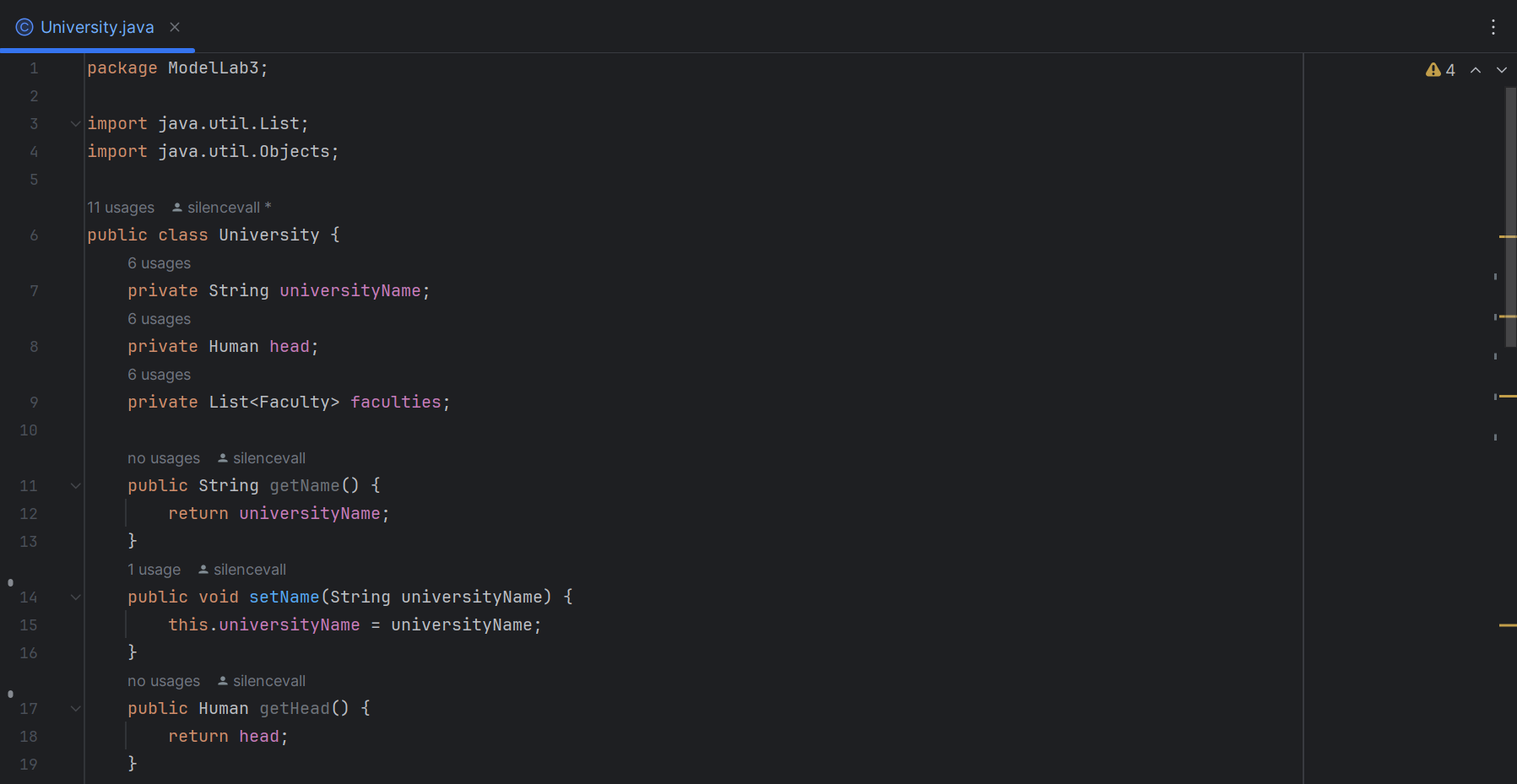
DepartmentCreator:





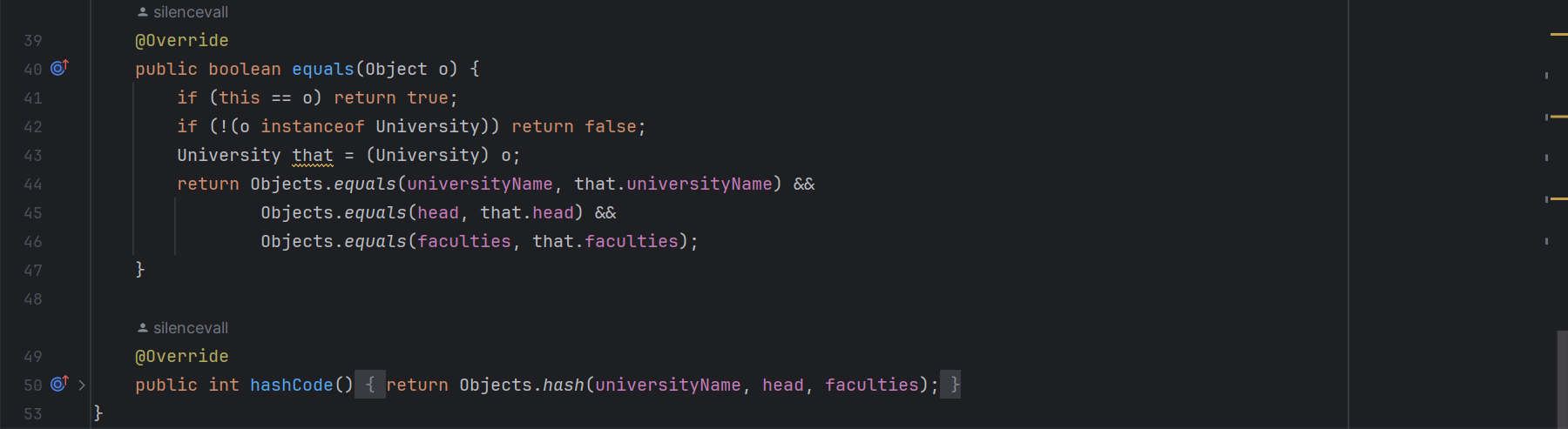
1. Папка ModelLab3:

University:

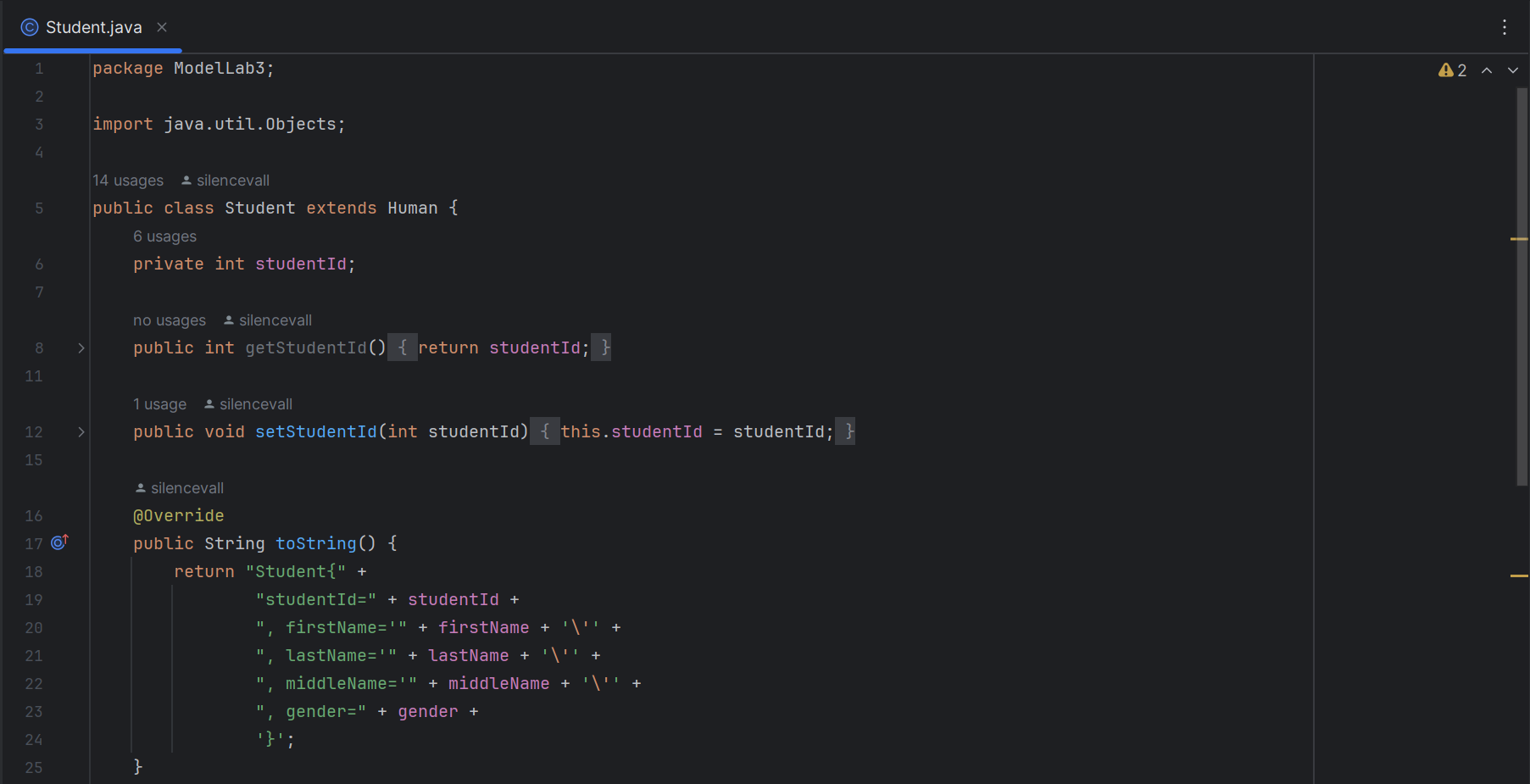


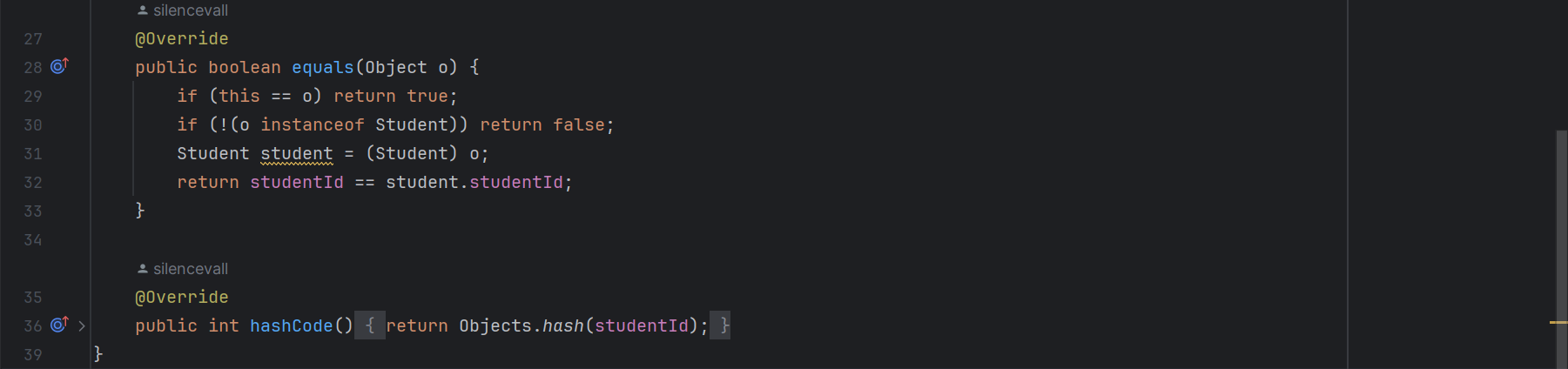




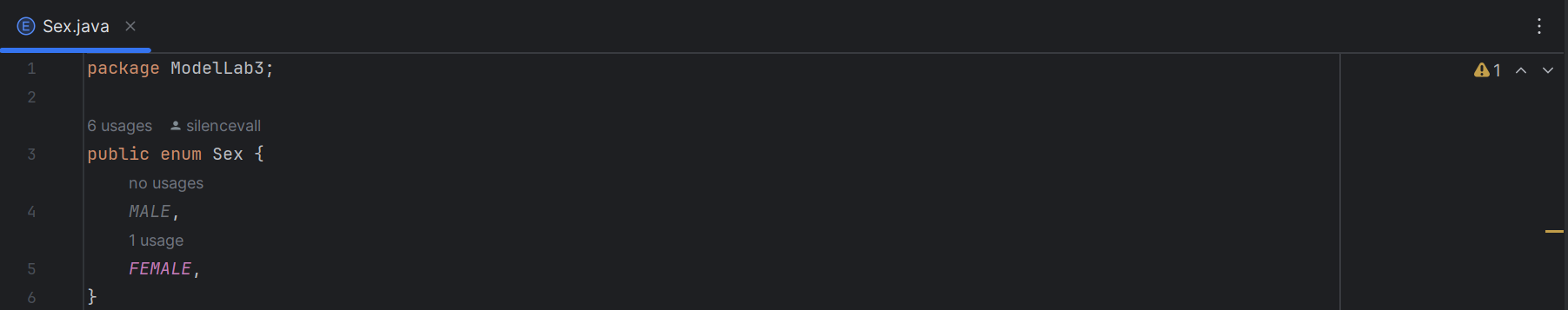


Student:

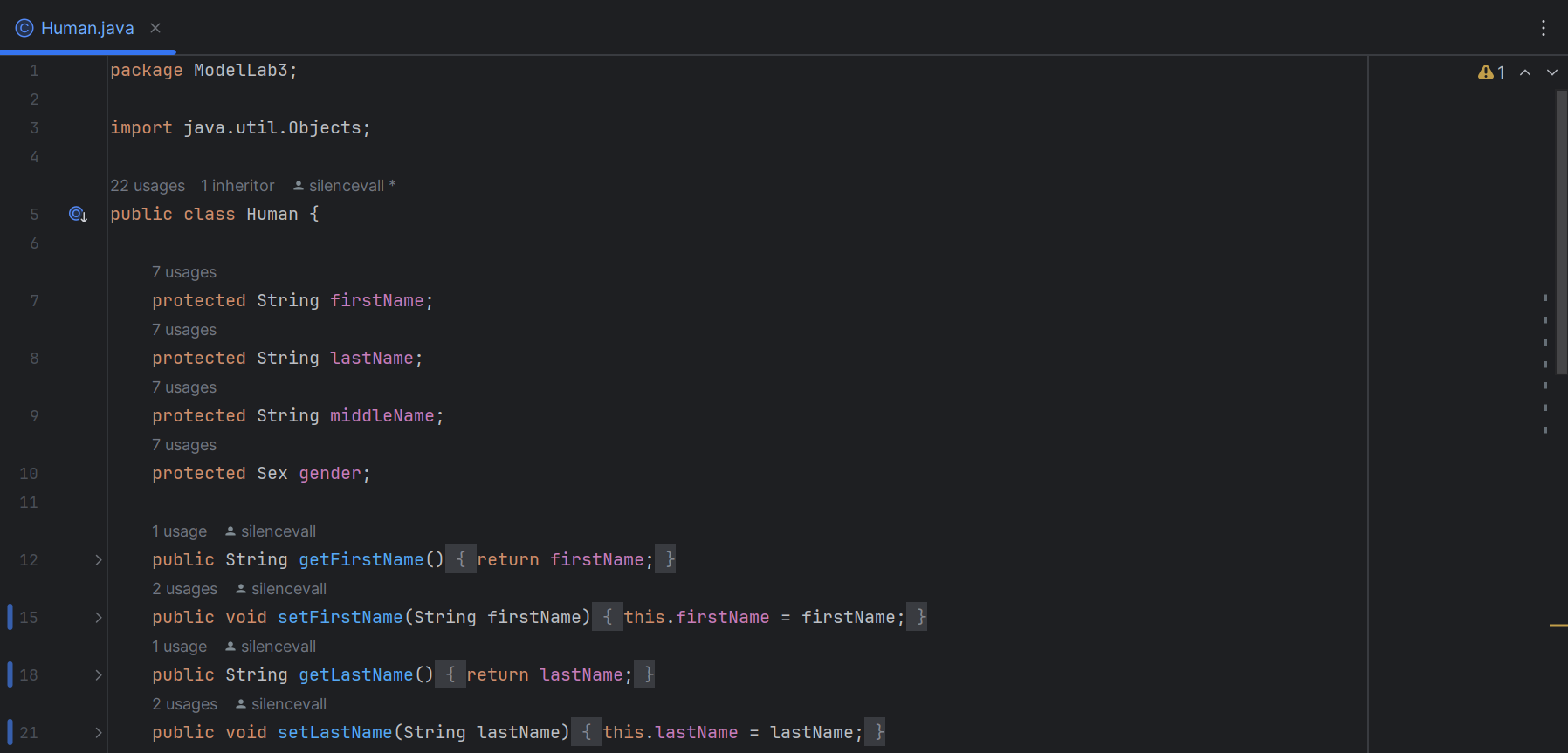




Sex:



Human:

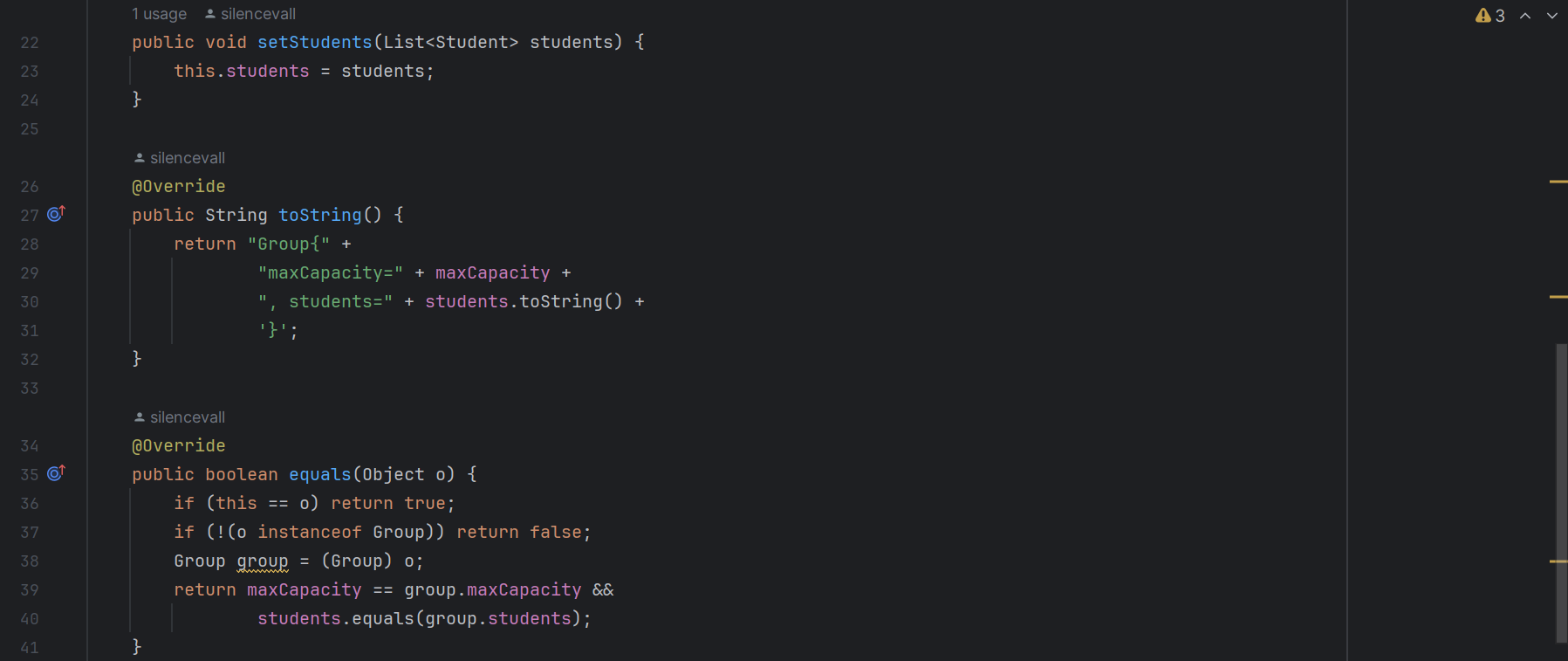


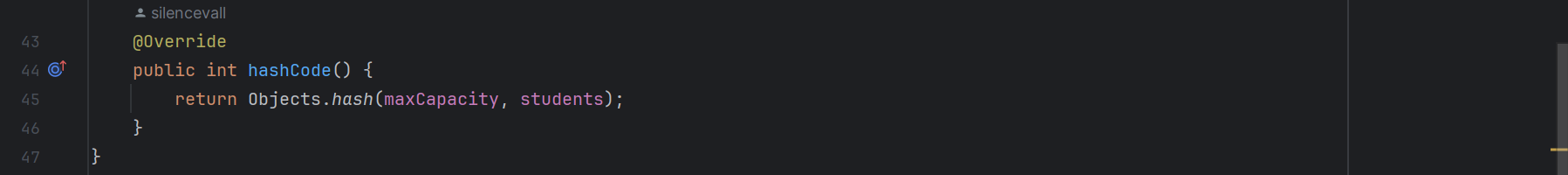




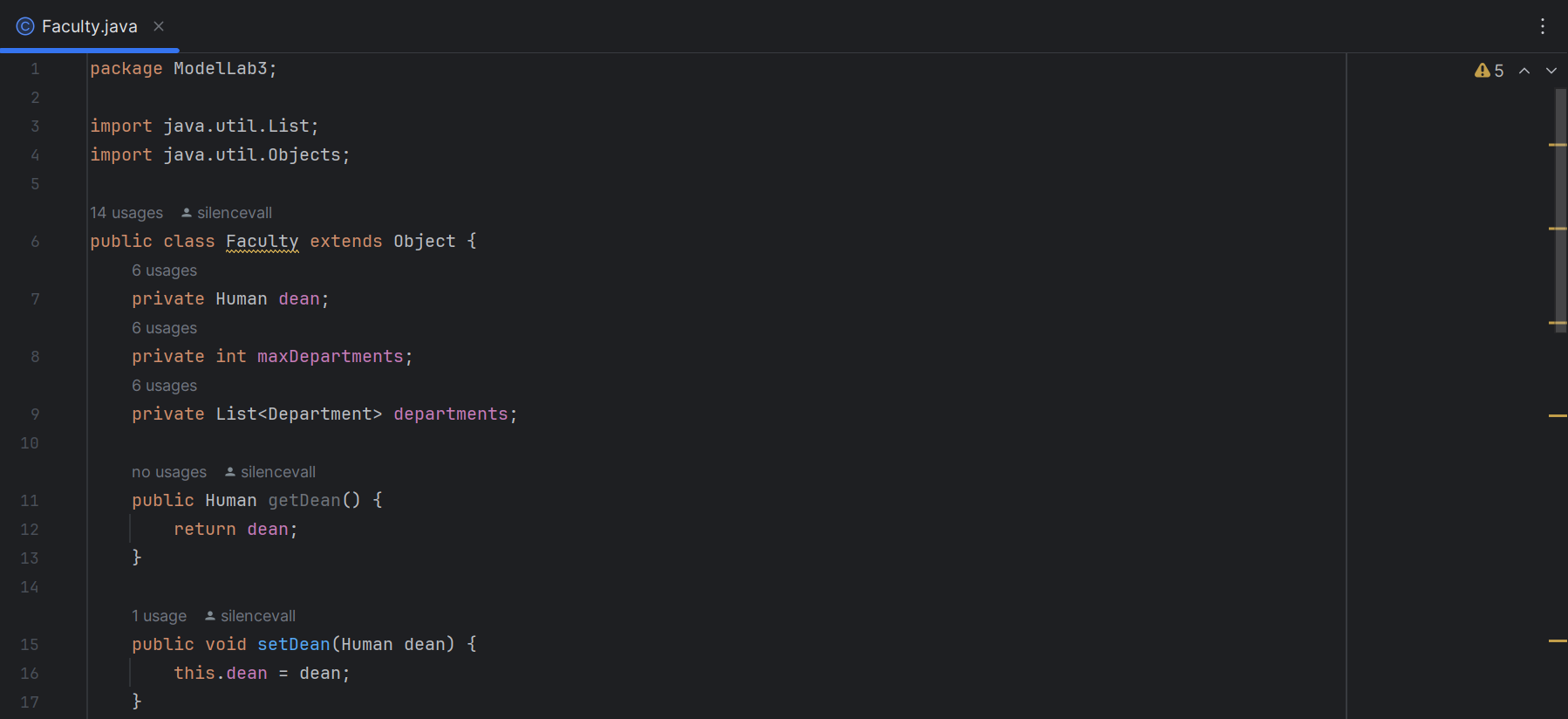
Group:





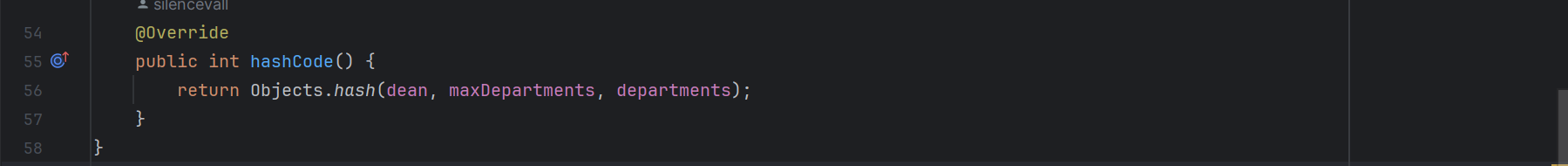


Faculty:

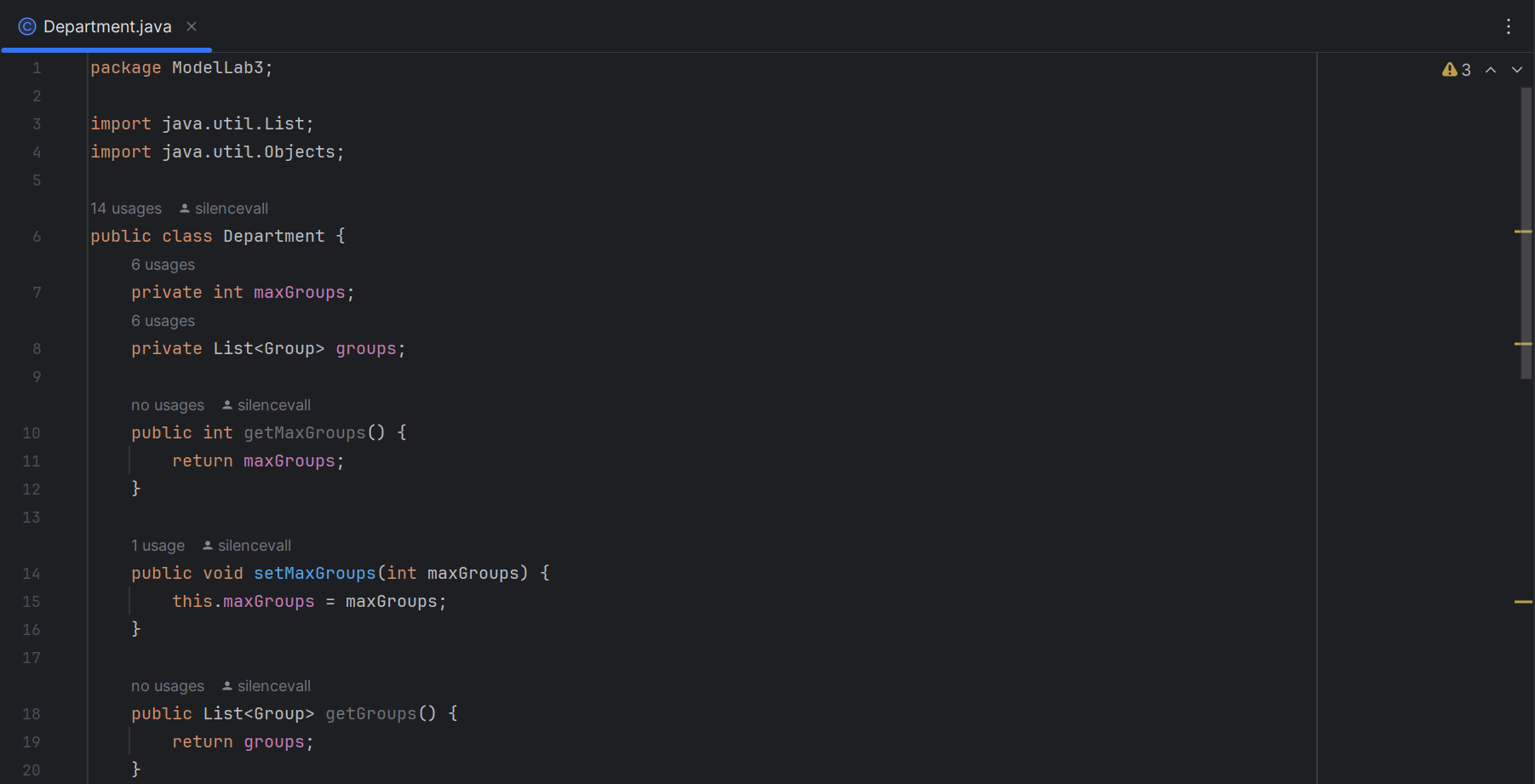




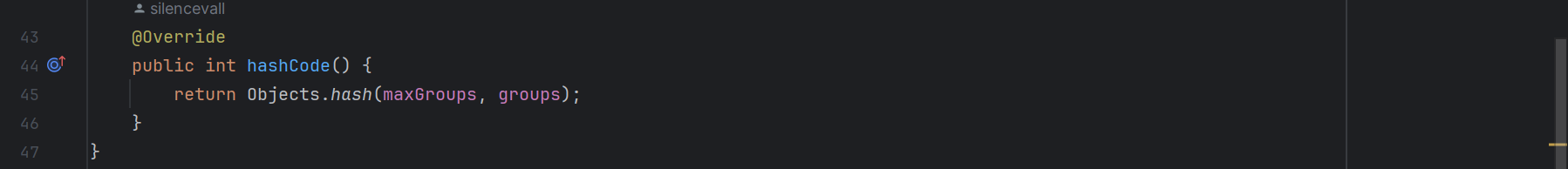




Department:



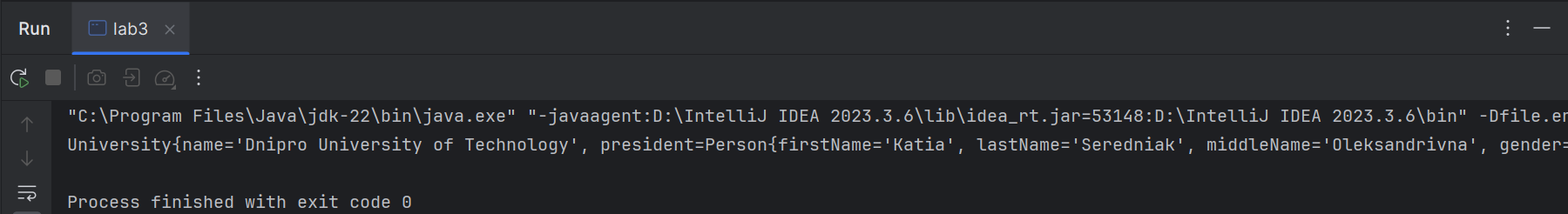




1. Файл lab3.java:



**Результат виконання коду:**



**Лабораторна робота номер 4**. JUnit. Json

Додати до лабораторної роботи 3 можливість запису університету у формат json, запис цього формату у файл, зчитування цього формату файлу, та створення об'єкту з текстового формату json. В проекті повинен бути зроблений JUnit тест, який буде виглядати наступним чином: створити об'єкт університет(oldUniversity), в якому в кожному підрозділі маються два підрозділи нижчого рівня. Наприклад на факультеті дві кафедри, на кожній кафедрі дві групи, на кожній групі два студенти. Цей об'єкт повинен бути записаний в файл у форматі json. Потім з цього файлу зчитаний та відновлений як newUniversity. В тесті повинні бути порівняні newUniversity та oldUniversity за допомогою методу equals. Якщо все зроблено правильно то університети повинні бути еквівалентні, а метод equals повинен повернути True. Для запису та зчитування університету у форматі json повинен бути зроблений клас JsonManager. Для безпосереднього перетворення університету у формат json та його відновлення цього формату, можливо використання сторонніх бібліотек наприклад Gson, Jackson чи будь-яких інших. Для початку розробки лабораторної роботи номер 4 повністю скопіювати програмний код лабораторної роботи номер 3. Не змішувати ці роботи не в якому разі.

**Код:**

1. JsonManager.java







1. JsonTest.java



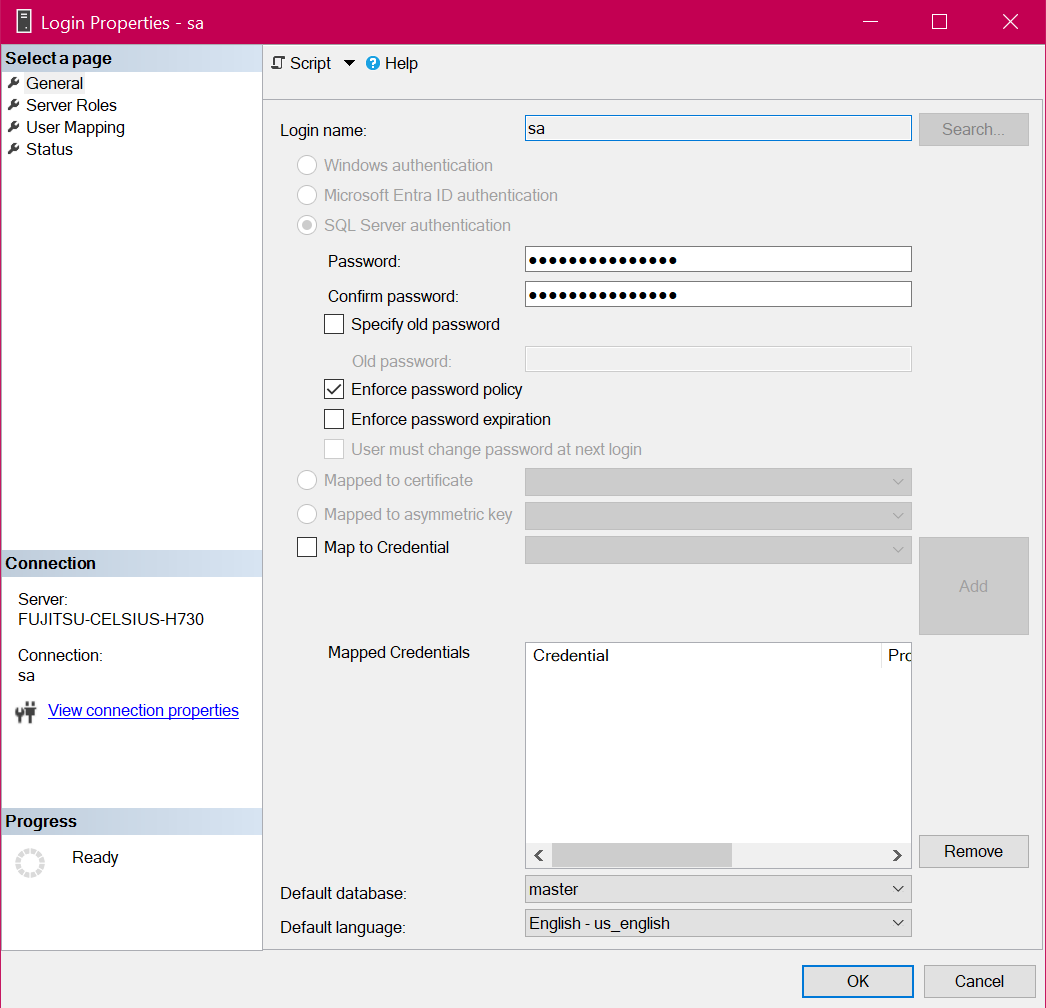
**Лабораторна робота номер 5.** Jdbc

Створити базу даних в будь якому сервері баз даних. Створити таблицю з переліком студентів вказати їх прізвище, ім'я, по батькові, день народження номер залікової книжки та ID. Створити програму що буде дозволяти виводити на екран інформацію про студентів які народилися в тому чи іншому місяці року. Програма повинна завдяки системі jdbc під'єднатися до вашої бази даних та робити до неї запроси. Вимог до розробки бази даних немає. Програма ж має бути написана за усіма стандартами ООП. Та може бути спроектована за двох принципів:

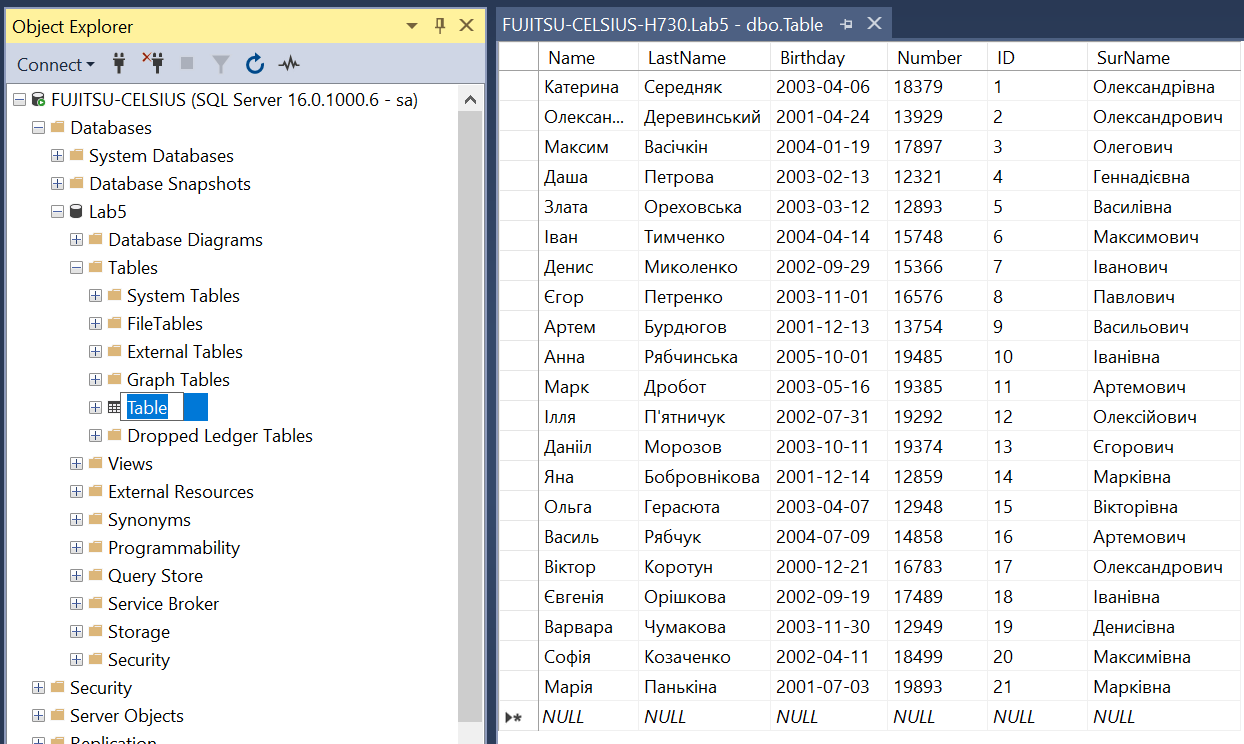
- при будь-якій ситуації буде забиратися весь перелік студентів, а вже на стороні java буде зроблено пошук необхідного

- SQL запрос буде сформований згідно запросу який зробив користувач і вже сервер управління баз даних буде вирішувати, які самі студенти народилися в тому чи іншому місяці. У висновку обов'язково пояснити чому вибрали той чи інший принцип, які в нього переваги та недоліки. Оцінка не залежить від того який сервер управління баз даних вибрали. Перелік студентів зробити не менше 20 людей. Місяць червень зробити місяцем, коли в жодного зі студентів немає дня народження. SQL код створення бази даних розмістити проекті 6 лабораторної роботи в файлі database в пекеджі resources. Для використання цієї лабораторної роботи рекомендується активно використовувати знання отримані на дисципліні що стосуються розробки баз даних. До паперового звіту обов'язково додати прінтскрін з програми в якій ви дивитесь інформацію вашого сервера управління баз даних, де показати створену таблицю, її ім'я та загальні відомості бази даних, наприклад назва, ім'я, назва користувача адміністратора, пароль тощо. Для роботи з сервером управління баз даних рекомендуємо використовувати програмне забезпечення компанії jetbrains datagrip. Або вбудовану панель користування базами даних, що міститься у середовищі intellij Idea, яка на сьогоднішній день підтримує майже всі сервери управління баз даних.

**Створений користувач:**

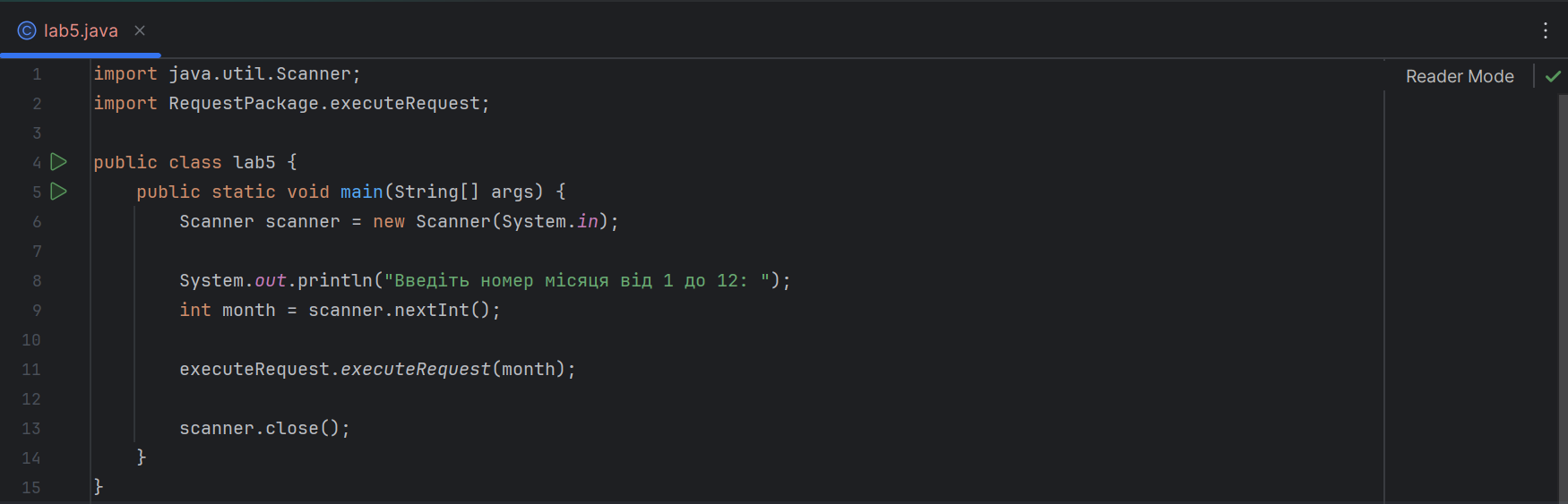


**Таблиця з даними студентів:**

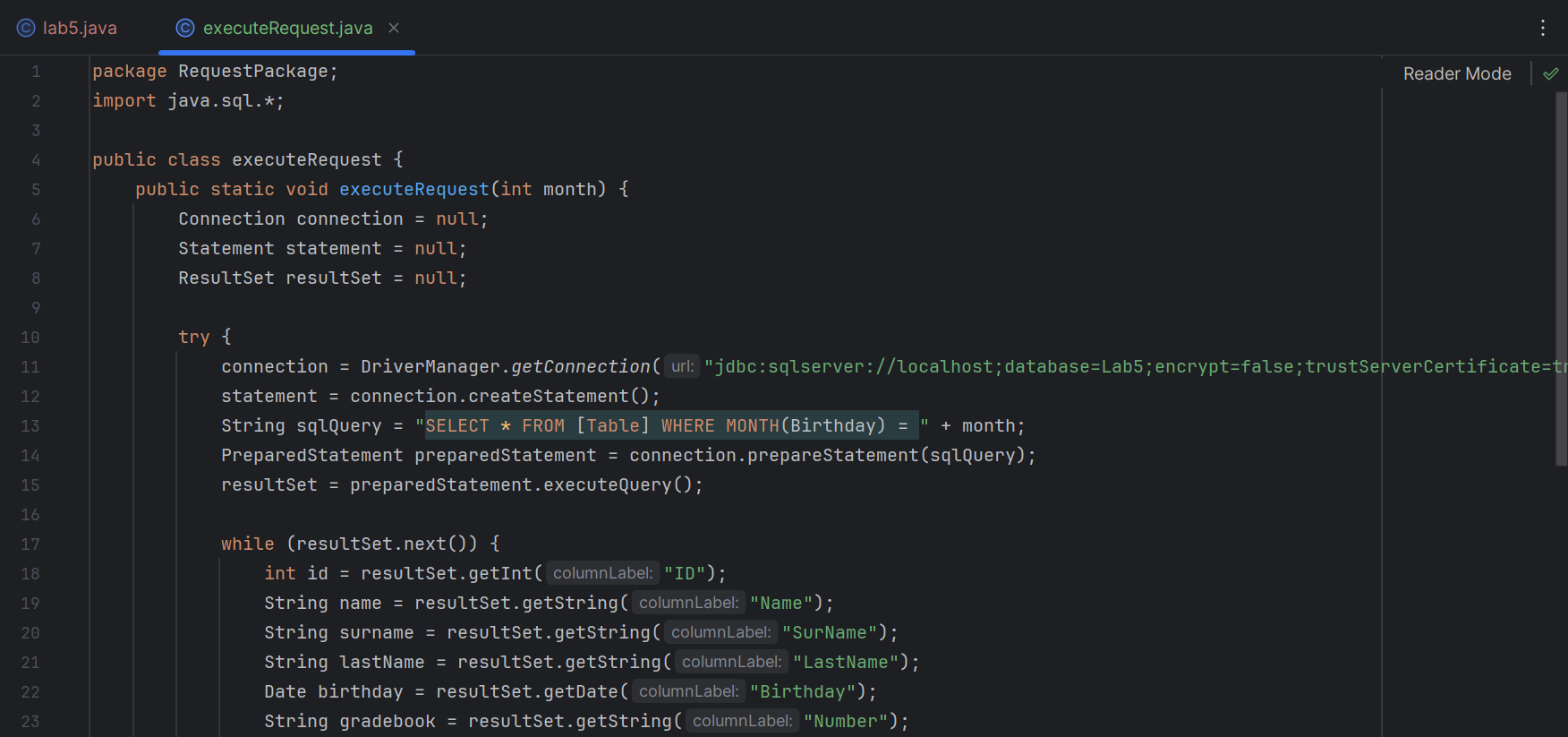
****

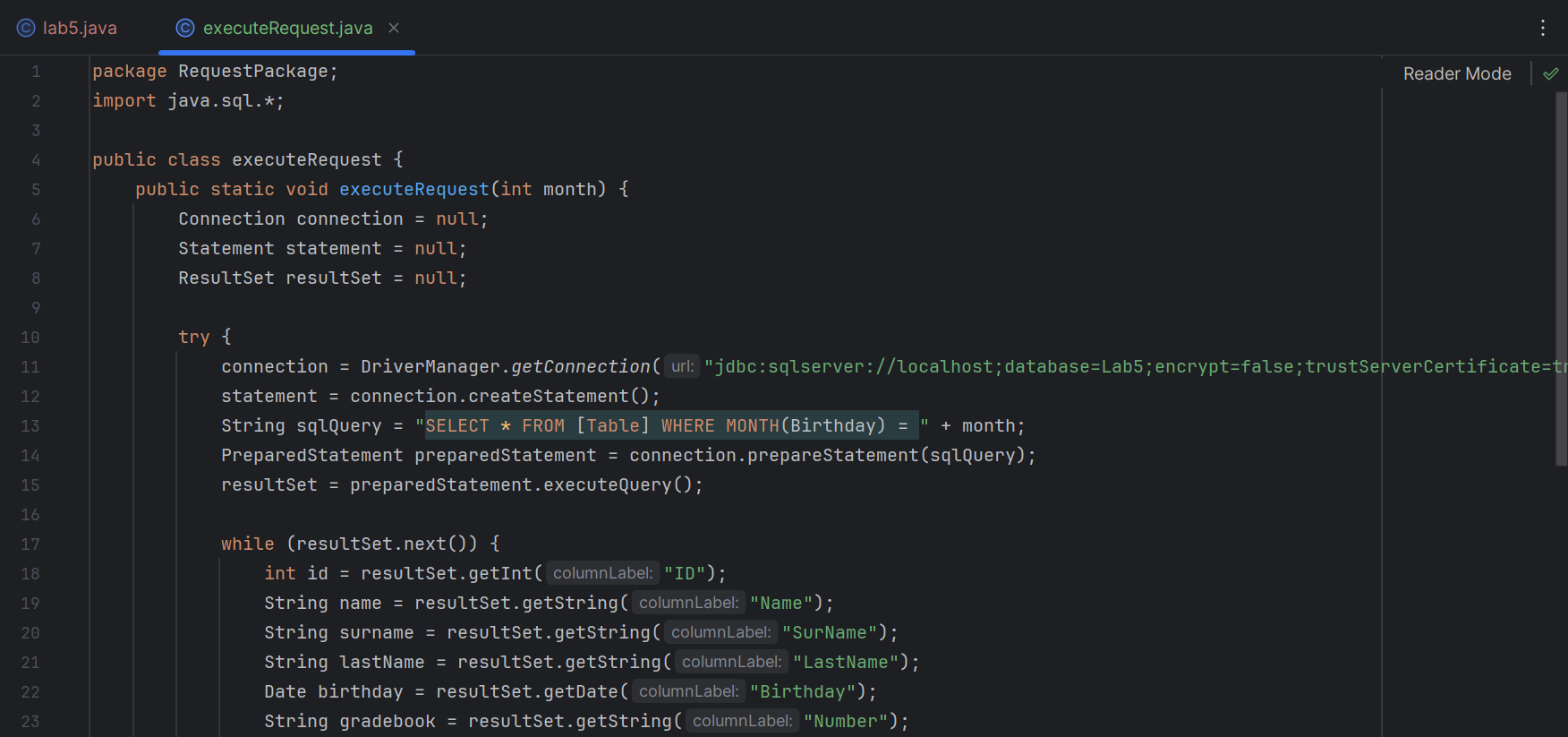
**Код:**

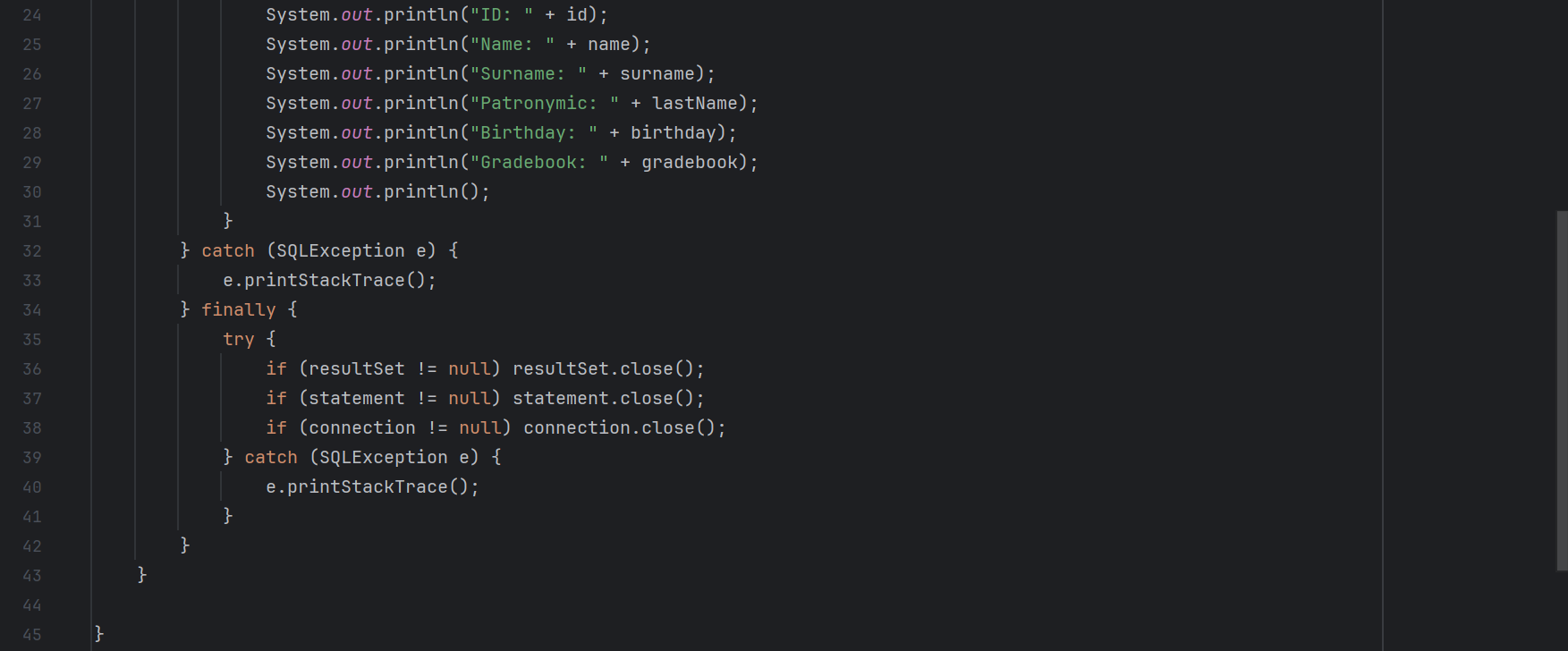
1. lab5.java



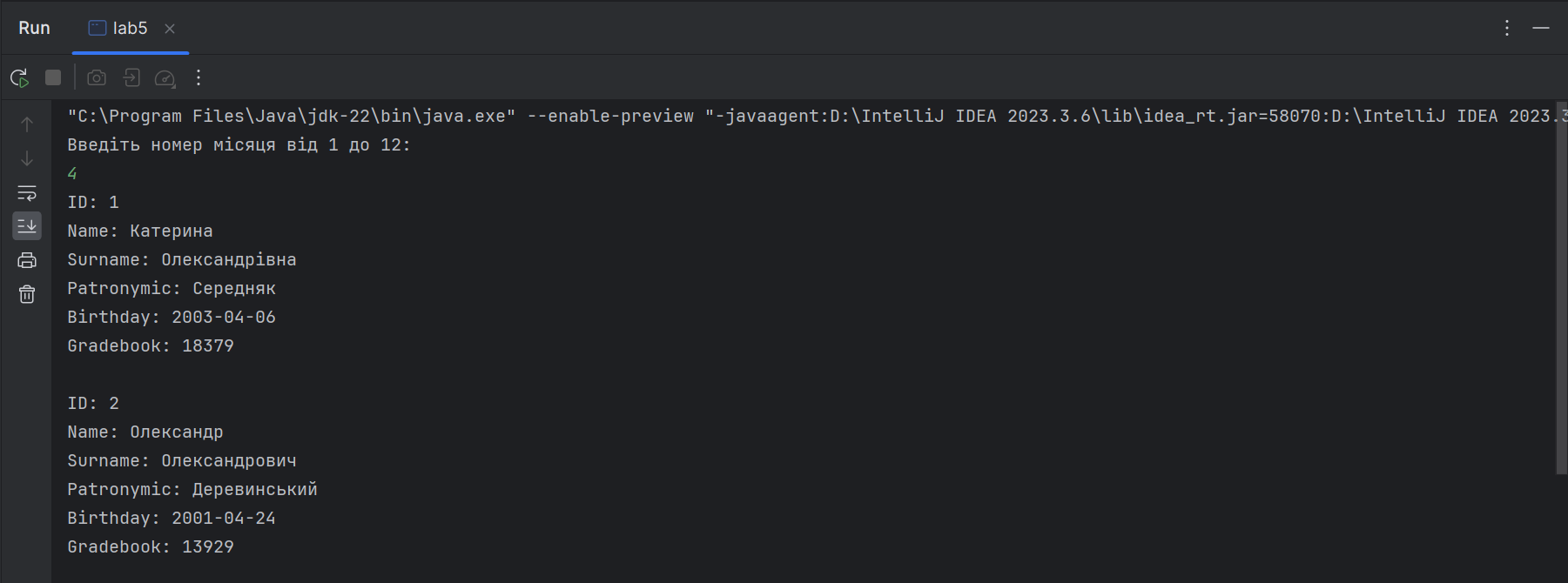
1. executeRequest







**Результат виконання коду:**





**Переваги:**

* Використання прямого SQL запиту спрощує код та робить його легким для розуміння.
* Код менший та менше витрачається на виконання операцій.
* Використання підготовленого запиту допомагає запобігти SQL ін'єкціям, оскільки параметри запиту не встановлюються прямо в рядок запиту.
* Деякі СУБД можуть оптимізувати виконання підготовлених запитів, що може покращити продуктивність.

**Недоліки:**

* Прямий SQL запит може бути уразливим до SQL ін'єкцій, якщо дані користувача не перевіряються на безпеку перед використанням.
* Відсутня або обмежена можливість обробки помилок, пов'язаних з підключенням до бази даних та виконанням запитів.
* Використання підготовленого запиту вимагає більше коду та робить його складнішим для розуміння.
* Це вимагає створення PreparedStatement та налаштування параметрів запиту, що може займати більше часу.