

Міністерство освіти і науки України

Національний університет «Львівська політехніка»



Звіт

З лабораторної роботи №1

З дисципліни «Кросплатформенні засоби програмування»

На тему: «Основи розробки програм мовою Java»

Виконав:
ст.гр. КІ-36
Литовко С.Г
Прийняв:
Іванов Ю.С.

Львів-2022

Мета роботи: Оволодіти основами розробки програм мовою Java.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Програми написані мовою Java виконуються в середовищі виконання програм Java, яке у термінології Java називається JRE (Java Runtime Environment). Структурно це середовище складається двох елементів: стандартних бібліотек класів Java та віртуальної машини Java, яка у термінології Java називається JVM (Java Virtual Machine). JVM в свою чергу складається з блоку перевірки безпечності коду програми, інтерпретатора або JIT (Just-in-time) компілятора та блоку керування пам'яттю.

Виконання програм написаних мовою Java починається з виклику JRE та передачі у нього скомпільованого у байт-код файлу з розширенням **.class*, що містить `main` метод. Якщо файлів **.class* є багато, то вони можуть бути заархівованими у файл з розширенням **.jar*. При цьому у файлі-архіві **.jar* мають міститися файли класів і ресурсів програми, а клас, який містить `main` метод має бути вказаний в декларативному заголовку Main-Class (Main-Class manifest header).

Для запуску середовища JRE використовуються два інструменти: `java` та `javaw`, - у які через параметри передається необхідна для виконання програми інформація. Різниця між інструментами полягає в тому, що перший виводить консольне вікно на екран, а другий – ні. Синтаксис команд запуску цих інструментів, які виконуються у командному рядку операційної системи, наступний:

```
java [опції] <class> [агументи ... ]  
java [опції] -jar <file.jar> [агументи... ]
```

```
javaw [опції] <class> [агументи... ]  
javaw [опції] -jar <file.jar> [агументи... ]
```

Параметри:

[опції]

Опції командного рядка ОС, які передаються в JRE.

<class>

Клас, який містить `main` метод.

<file.jar>

Ім'я **.jar* файлу, який необхідно запустити на виконання. Використовується лише з опцією `-jar`. Вказаний **.jar* файл має містити файли класів і ресурсів програми. Клас, який містить `main` метод має бути вказаний в декларативному заголовку Main-Class (Main-Class manifest header).

[агументи...]

Агументи командного рядка, які мають бути передані `main` методу.

Переданий у JRE байт-код програми поступає у блок перевірки безпечності коду програми, який здійснює три головні перевірки:

- перевірка чи переходи відбуваються в допустимі місця програми;
- перевірка чи дані є ініціалізовані і посилання є типізовані;
- жорстка перевірка можливості доступу до приватних секцій класів і пакетів.

Перші дві перевірки переважно відбуваються при завантаженні програми в JRE, а остання – в процесі виконання програми, коли одні класи звертаються до полів інших класів.

Після успішного завершення перевірки байт-коду JVM емулює (виконує) набір інструкцій, написаних для JVM, інтерпретуючи їх або використовуючи JIT компілятор. Сучасні JVM використовують JIT компілятор, який одночасно компілює частини байт-коду, що мають аналогічну функціональність. Завдяки цьому зменшується кількість часу, що необхідна для компіляції порівняно з використанням інтерпретатора.

Для створення власних програм наявність JRE є недостатньою, оскільки JRE здатне лише виконувати байт-код, але не генерувати його в процесі компіляції, бо не містить компілятора. Тому для створення власних програм мовою Java необхідно встановити JDK (Java Development Kit), яке підтримує ту версію Java, якою ви будете писати програми. На відміну від JRE JDK містить компілятор, який називається *javac*, набір утиліт, які працюють в режимі командного рядка та набір бібліотек для платформ Java SE і Java EE. JDK так само як і JRE є безкоштовним, та разом з усією необхідною документацією і вихідним кодом бібліотек Java може бути вільно завантаженим з сайту фірми-виробника Oracle.

Встановивши JDK можна приступати до розробки програм мовою Java. Типова програма на мові Java складається з файлів з кодом програми на мові Java, які мають розширення **.java*. Для їх компіляції в байт-код використовується компілятор *javac*. Наприклад, для компіляції програми на мові Java, що міститься у файлі *example.java* необхідно ввести в командному рядку ОС:

javac example.java

в результаті успішного виконання цієї команди одержимо *example.class* – файл з байт-кодом, який готовий до виконання у JRE.

Нажаль, JDK не є інтегрованим середовищем розробки та не має графічного інтерфейсу, що ускладнює процес написання та відлагодження програм мовою Java. Для полегшення процесу розробки програм мовою Java компанія Oracle і сторонні виробники пропонують інтегровані середовища розробки з графічним інтерфейсом та відлагоджувачами, які в своїй роботі використовують JDK. Тому для їх встановлення спочатку необхідно встановити JDK. Такими середовищами є:

NetBeans IDE - безкоштовне інтегроване середовище розробки для всіх платформ Java - Java ME, Java SE і Java EE. Пропагувалося Sun Microsystems, як базовий засіб для розробки ПЗ на мові Java і інших мовах (C, C++, PHP, Fortran тощо).

Eclipse IDE – одне з найпопулярніших безкоштовних інтегрованих середовищ розробки для Java SE, Java EE і Java ME. Пропагується IBM, як базовий засіб для розробки ПЗ на мові Java і інших мовах (C, C++, Ruby, Fortran тощо).

IntelliJ IDEA - інтегроване середовище розробки для платформ Java SE, Java EE і Java ME. Розробник - компанія JetBrains. Розповсюджується в двох версіях: безкоштовної (Community Edition) та комерційної пропрієтарної (Ultimate Edition).

JDeveloper - інтегроване середовище розробки для платформ Java SE, Java EE і Java ME. Розробник - компанія Oracle.

BlueJ - інтегроване середовище розробки програм на мові Java, створене в основному для використання в навчанні, але також підходить для розробки невеликих програм.

ЗАВДАННЯ

1. Написати, скомпілювати, налагодити та запустити програму на мові Java, яка виводитиме на екран прізвище, ім'я та групу автора програми.
2. Скласти звіт про виконану роботу з приведенням тексту програми та результату її виконання.
3. Дати відповідь на контрольні запитання.

Код програми:

```
public class hellojava
{
    public static void main(String[] args)
    {
        String message = new String();
        message = "Lytovko Sofia KI-36";
        System.out.println(message);
    }
}
```

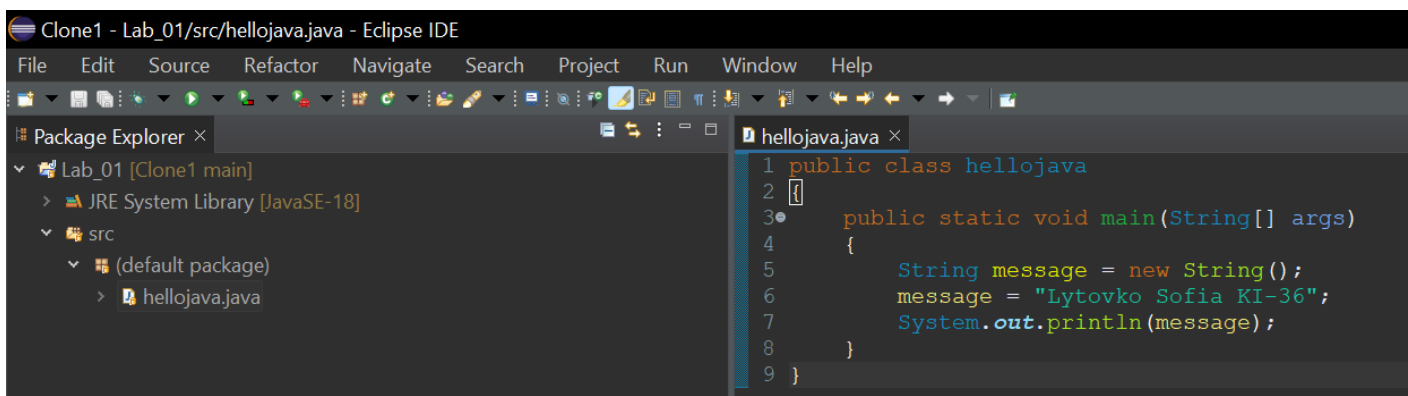
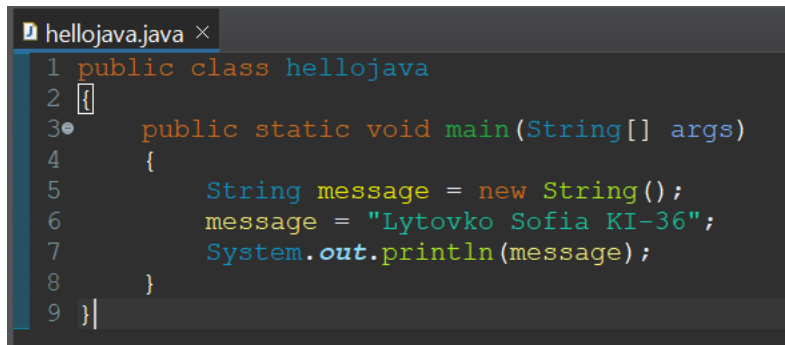


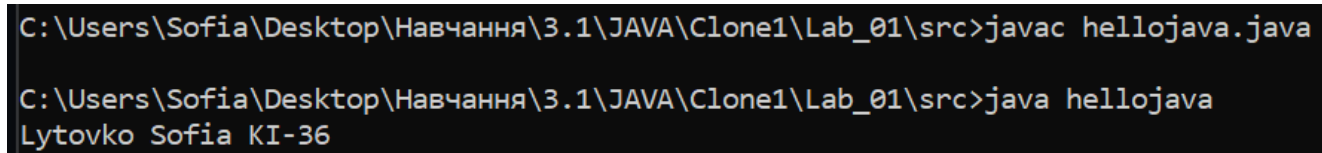
Рис.1. Середовище Eclipse IDE з відкритим проектом Java.

A screenshot of the Eclipse IDE showing a Java file named 'hellojava.java'. The code is as follows:

```
1 public class hellojava
2 {
3     public static void main(String[] args)
4     {
5         String message = new String();
6         message = "Lytovko Sofia KI-36";
7         System.out.println(message);
8     }
9 }
```

Рис.2. Код програми в середовищі Eclipse IDE.

Результат програми:

A screenshot of a Windows command prompt showing the compilation and execution of the Java program. The commands and their outputs are:

```
C:\Users\Sofia\Desktop\Навчання\3.1\JAVA\Clone1\Lab_01\src>javac hellojava.java

C:\Users\Sofia\Desktop\Навчання\3.1\JAVA\Clone1\Lab_01\src>java hellojava
Lytovko Sofia KI-36
```

Висновок: На цій лабораторній роботі я оволоділа основами розробки програм мовою Java.