

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Факультет радіоелектроніки, комп'ютерних систем та інфокомунікацій

Кафедра комп'ютерних систем, мереж і кібербезпеки

**Лабораторна робота**

з Кросплатформенні технології  
(назва дисципліни)

на тему: «Робота з інструментальними засобами пакета JDK: розробка  
простих  
застосунків на мові Java»

Виконав: студент 4 курсу групи №5456  
напряму підготовки (спеціальності)  
123 – комп'ютерна інженерія

\_\_\_\_\_  
(шифр і назва напряму підготовки (спеціальності))

Поліщук А.О.

\_\_\_\_\_  
(прізвище й ініціали студента)

Прийняв: асистент каф.503

Годованюк П.А.

\_\_\_\_\_  
(посада, науковий ступінь, прізвище й ініціали)

Національна шкала: \_\_\_\_\_

Кількість балів: \_\_\_\_\_

Оцінка: ECTS \_\_\_\_\_

## 1. Постановка мети і завдань роботи

### 1.1 Постановка мети

Отримання практичних навичок роботи з інструментальними засобами

пакета JDK (компілятор, динамічний інтерпретатор, генератор документації) для розробки програмного забезпечення на мові Java.

### 1.2 Завдання роботи

1. Вивчити теоретичний матеріал до лабораторної роботи.
2. Реалізувати на мові Java програмний код додатку відповідно до No варіанту.
3. Скомпілювати і запустити цю програму під керуванням JVM.
4. Отримати байт-код програми за допомогою утиліти javap.exe.
5. Отримати документацію на розроблене додаток за допомогою утиліти javadoc.exe у вигляді HTML-файлів.

## 2. Лістинг коду

### 2.1. Initials.java

```
/*
 * @author Polishuk Artem Alexandrovich
 * @adress a.polishuk@student.csn.khai.edu
 * @version 1.0
 * @since 2020-09-23
 */

public class Initials
{
    /**
     * This method demonstrates short info about student.
     * @param args Unused.
     * @return Nothing.
     */
    public static void main(String[] args)
```

```

{
    /*This is the code which display some information about student
(author, group, FIO)*/
    System.out.println("Author: Polishuk Artem Alexandrovich");
    System.out.println("Group: 545-b\n");
    System.out.println("*****      *****      *****");
    System.out.println("****  ***      ***  ***      ***  ****");
    System.out.println("****      ***  ***      ***  ***      ***  ****");
    System.out.println("****      ***  ***      ***  ***      ***  ****");
    System.out.println("****      ***  ***      ***  ***      ***  ****");
    System.out.println("****  ***      ***      ***  ***      ****");
    System.out.println("*****      *****");
    *****");
    System.out.println("****      *****");
    *****");
    System.out.println("****      ***      ***  ***      ****");
    System.out.println("****      ***      ***  ***      ****");
    System.out.println("****      ***      ***  ***      ****");
    System.out.println("****      ***      ***  ***      ****");
}
}

```

## 2.2. Bite-code.txt

Compiled from "Initials.java"

```

public class Initials {
    public Initials();

```

Code:

0: aload\_0

1: invokespecial #1 // Method

java/lang/Object."<init>":()V

4: return

```

    public static void main(java.lang.String[]);

```

Code:

0: getstatic #7 // Field

java/lang/System.out:Ljava/io/PrintStream;

3: ldc #13 // String Author: Polishuk Artem

Alexandrovich

5: invokevirtual #15 // Method

java/io/PrintStream.println:(Ljava/lang/String;)V

```

      8: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     11: ldc          #21          // String Group: 545-b\n
     13: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
     16: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     19: ldc          #23          // String *****
*****
     21: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
     24: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     27: ldc          #25          // String ***   ***   ***   ***
***   ***
     29: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
     32: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     35: ldc          #27          // String ***   ***   ***   ***   ***
***
     37: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
     40: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     43: ldc          #29          // String ***   ***   ***   ***   ***
***
     45: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
     48: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     51: ldc          #31          // String ***   ***   ***   ***   ***
***
     53: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
     56: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
     59: ldc          #33          // String ***   ***   ***   ***   ***
***
     61: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V

```

```

        64: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
        67: ldc          #35          // String *****
*****

        69: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
        72: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
        75: ldc          #37          // String ***
*****

        77: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
        80: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
        83: ldc          #39          // String ***
***
***

        85: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
        88: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
        91: ldc          #39          // String ***
***
***

        93: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
        96: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
        99: ldc          #39          // String ***
***
***

       101: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
       104: getstatic    #7          // Field
java/lang/System.out:Ljava/io/PrintStream;
       107: ldc          #39          // String ***
***
***

       109: invokevirtual #15          // Method
java/io/PrintStream.println:(Ljava/lang/String;)V
       112: return
    }

```

### 3. Скріншот Javadoc-документації до застосунку.

The screenshot shows the Javadoc documentation for the `Initials` class. The interface includes a navigation bar with links like PACKAGE, CLASS, TREE, DEPRECATED, INDEX, and HELP. Below the navigation bar, the class name `Initials` is displayed, along with its inheritance path: `java.lang.Object` and `Initials`. The class is defined as `public class Initials` and extends `java.lang.Object`. The documentation is organized into sections: **Constructor Summary**, **Method Summary**, and **Constructor Details**. The **Constructor Summary** section shows a table with one constructor: `Initials()`. The **Method Summary** section shows a table with one method: `main(java.lang.String[] args)`. The **Constructor Details** section shows the constructor `Initials`.

PACKAGE: **CLASS** TREE DEPRECATED INDEX HELP

SUMMARY: NESTED | FIELD | CONSTR | METHOD DETAIL: FIELD | CONSTR | METHOD SEARCH

#### Class Initials

java.lang.Object  
Initials

public class **Initials**  
extends java.lang.Object

#### Constructor Summary

**Constructors**

Constructor	Description
<code>Initials()</code>	

#### Method Summary

**All Methods** **Static Methods** **Concrete Methods**

Modifier and Type	Method	Description
static void	<code>main(java.lang.String[] args)</code>	This method demonstrates short info about student.

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

#### Constructor Details

**Initials**

### 4. Посилання на репозиторій

<https://github.com/PolishukArtem/Cross>

### Выводы:

В результате выполнения лабораторной работы были получены практические навыки работы с инструментальными средствами пакета JDK для разработки программного обеспечения на языке Java. Во время выполнения работы было создано простое приложение на языке Java, создан .bat файл для запуска приложения, получения байт-кода и генерации документации. Также был создан репозиторий на GitHub для лабораторных работ по данному курсу.