

**Мета:** навчитися визначати символні та рядкові типи даних і виконувати операції над ними.

### Хід роботи

- 1 У середовищі IntelliJ створити проєкт та під'єднати його до репозиторію Git.
- 2 Відповідно до свого номеру у списку студентської групи обрати завдання з таблиці 4.
- 3 Написати відповідний вихідний код програми мовою Java.
- 4 За необхідності в ході виконання та обов'язково по завершенні роботи завантажити результати на репозиторій Git.
- 5 Створити та надати викладачеві звіт з виконання лабораторної роботи.

#### Завдання:

1) З таблиці 5 прочитати фрагменти коду з'ясувати як вони працюють. Якщо в результаті перегляду не зрозуміло що вони роблять, вставити відповідні шматки коду в просту програму на Java та з'ясувати що вони виводять на екран. До звіту включити результати обробки таблиці з позначкою які фрагменти ви змогли прочитати та розтлумачити без компіляції.

Фрагмент коду: `System.out.println(Character.isUpperCase(100));`

Цей код перевіряє, чи є символ, що має код 100 у таблиці Unicode, великою літерою. Але він поверне false, оскільки символ 100 в Unicode відповідає літері d, яка є малою.

					<b>ФКЗЕ. 121ТП00.03ЛР</b>		
Змн.	Арк.	№ докум.	Підпис	Дата	Вивчення типу даних String та операції з ними.		
Розроб.	Лашко Р.А.						
Перевір.	Новодравнова						
Реценз.							
Н. Контр.	Саприкіна І.Г.						
Затверд.	Саприкіна І.Г.				Група ПЗ-21-1/9		
					Літ.	Арк.	Аркуші
						1	7

```

J main.java > ...
1 public class main {
    Run | Debug
2     public static void main(String[] args) {
3         System.out.println(Character.isUpperCase(codePoint:100));
4     }
5 }
6
7

```

OUTPUT    DEBUG CONSOLE    **TERMINAL**    PORTS

```

PS C:\Users\katar\Desktop\УЧ\ТП\ЛР3> & 'C:\Program Files\Microsoft\jdk-11.0.16
.101-hotspot\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend
=y,address=localhost:50113' '-cp' 'C:\Users\katar\AppData\Roaming\Code\User\wor
kspaceStorage\db4c3179c4f85e8627cc0a41d10a655c\redhat.java\jdt_ws\ЛР3_e2c96bf5\
bin' 'main'
false

```

Рисунок 1 – Результат виконання пункту 1

- 2) Написати просту програму яка друкує на екран символи емоджі з рисунку 2,3. Кількість символів 3, символи обрати на власний смак.

```

J main.java > main > main(String[])
1 public class main {
    Run | Debug
2     public static void main(String[] args) {
3         System.out.println(x:"\uD83D\uDE00"); // Усміхнене обличчя
4         System.out.println(x:"\uD83D\uDE02"); // Обличчя зі сльозами від с
5         System.out.println(x:"\uD83D\uDE09"); // Підморгує обличчя
6     }
7 }
8
9

```

OUTPUT    DEBUG CONSOLE    **TERMINAL**    PORTS

```

PS C:\Users\katar\Desktop\УЧ\ТП\ЛР3> ^C
PS C:\Users\katar\Desktop\УЧ\ТП\ЛР3>
PS C:\Users\katar\Desktop\УЧ\ТП\ЛР3> c:; cd 'c:\Users\katar\Desktop\УЧ\ТП\ЛР3'
; & 'C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe' '-agentli
b:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:51261' '-cp' 'C
:\Users\katar\AppData\Roaming\Code\User\workspaceStorage\db4c3179c4f85e8627cc0a
41d10a655c\redhat.java\jdt_ws\ЛР3_e2c96bf5\bin' 'main'
?
?
?

```

Рисунок 2 – Результат виконання пункту 2

3) Написати програму яка з'ясовує чи є подане речення окличним чи питальним. Речення подається у вигляді рядка його зміст придумати самостійно.

```

J main.java > main > main(String[])
1  import java.util.Scanner;
2
3  public class main {
4      Run | Debug
      public static void main(String[] args) {
5          Scanner scanner = new Scanner(System.in);
6
7          // Запит користувача на введення речення
8          System.out.println(x:"Введіть речення: ");
9          String sentence = scanner.nextLine();
10
11         // Перевірка типу речення
12         if (sentence.endsWith(suffix:"!")) {
13             System.out.println(x:"Це окличне речення.");
14         } else if (sentence.endsWith(suffix:"?")) {
15             System.out.println(x:"Це питальне речення.");
16         } else {
17             System.out.println(x:"Це звичайне речення.");
18         }
19         scanner.close();
20     }
21 }
22
OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS C:\Users\katar\Desktop\УЧ\ТП\ЛР3>
PS C:\Users\katar\Desktop\УЧ\ТП\ЛР3> c:: cd 'c:\Users\katar\Desktop\УЧ\ТП\ЛР3'
; & 'C:\Program Files\Microsoft\jdk-11.0.16.101-hotspot\bin\java.exe' '-agentli
b:jdpw=transport=dt_socket,server=n,suspend=y,address=localhost:51510' '-cp' 'C
:\Users\katar\AppData\Roaming\Code\User\workspaceStorage\db4c3179c4f85e8627cc0a
41d10a655c\redhat.java\jdt_ws\ЛР3_e2c96bf5\bin' 'main'
Введіть речення:
Привіт!
Це окличне речення.

```

Рисунок 3 – Результат виконання пункту 3

### Контрольні питання:

1 Які причини призвели до виникнення додаткових кодових таблиць.

Причинами виникнення додаткових кодових таблиць стали потреби компаній у власних стандартах, розширення комп'ютерної техніки в країнах з

					ФКЗЕ. 121ТП00.03ЛР	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		3

ієрогліфічним письмом та обмеженість однобайтових таблиць, що призвела до впровадження Unicode.

## 2 Що таке площина, яку кількість символів можна в ній відобразити.

Площина в Unicode — це група символів, що може вміщати до 65,536 символів.

## 3 Яким чином можна перевірити можливості виходу коду символу за межі Unicode.

Перевірити вихід символу за межі Unicode можна, перевіривши, чи код символу перевищує максимальне значення 0x10FFFF (в десятковій системі 1,114,111).

## 4 Чим клас String відрізняється від класу StringBuffer.

Клас String створює незмінні рядки (immutable), тобто їх не можна змінити після створення, тоді як StringBuffer створює змінні рядки (mutable), які можна змінювати без створення нових об'єктів, що робить його ефективнішим для частих змін рядка.

## 5 Яким чином в Java заблокована можливість створення похідних класів від String та StringBuffer.

У Java класи String та StringBuffer оголошені як final, що блокує можливість їх успадкування і створення похідних класів.

## 6 Чому для порівняння рядків не можна використовувати логічні оператори == і !=?

Оператори == і != порівнюють лише посилання на об'єкти, а не їх значення. Для порівняння змісту рядків потрібно використовувати метод .equals(), який порівнює значення символів у рядках, а не їх посилання в пам'яті.

					ФКЗЕ. 121ТП00.03ЛР	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		4

## 7 Яким чином налаштування локалі впливають на роботу з рядками.

Налаштування локалі впливають на обробку рядків у Java, зокрема на сортування, форматування, регістр і методи порівняння. Наприклад, для різних локалей однакові символи можуть мати різний порядок сортування або правила регістру.

**Висновок:** навчився визначати символічні та рядкові типи даних і виконувати операції над ними.

					ФКЗЕ. 121ТП00.03ЛР	Арк.
						5
Змін.	Арк.	№ докум.	Підпис	Дата		