

## Лабораторна робота №1

Мета: ознайомитися із основами мови програмування Java.

### Хід роботи

#### Завдання № 1.

Перетворення заданого цілого числа у двійкову систему.

*Рекомендовані посилання:*

- <https://docs.oracle.com/javase%2Ftutorial%2F/java/nutsandbolts/datatypes.html>
- <https://www.baeldung.com/java-primitives>
- [https://www.w3schools.com/java/java\\_operators.asp](https://www.w3schools.com/java/java_operators.asp)
- [https://www.w3schools.com/java/java\\_for\\_loop.asp](https://www.w3schools.com/java/java_for_loop.asp)

#### Завдання № 2.

Написати простий парсер виразів. Програма повинна вирішувати прості вирази типу “2 + 4 = ?”, “2 + 4 = ”, “2 + 4”. Повинен підтримувати основні операції (додавання, віднімання, множення, ділення). Необхідно обробити виключні ситуації.

*Рекомендовані посилання:*

- [https://www.w3schools.com/java/java\\_strings.asp](https://www.w3schools.com/java/java_strings.asp)
- [https://www.w3schools.com/java/java\\_switch.asp](https://www.w3schools.com/java/java_switch.asp)

#### Завдання № 3. FizzBuzz.

Вивести в консоль числа від 1 до 100. Замість чисел кратних 3 виводити Fizz, замість чисел кратних 5 вивести Buzz. Замість чисел що кратні і 3, і 5 вивести FizzBuzz.

#### Завдання № 4.

Вивести на екран кількість кожного символу у рядку. Виконати завдання використовуючи масиви та створити клас, що представляє пару значень (Символ, Кількість)

Рекомендовані посилання:

- [https://www.w3schools.com/java/java\\_arrays.asp](https://www.w3schools.com/java/java_arrays.asp)
- [https://www.w3schools.com/java/java\\_oop.asp](https://www.w3schools.com/java/java_oop.asp)
- [https://www.w3schools.com/java/java\\_classes.asp](https://www.w3schools.com/java/java_classes.asp)
- [https://www.w3schools.com/java/java\\_class\\_attributes.asp](https://www.w3schools.com/java/java_class_attributes.asp)
- [https://www.w3schools.com/java/java\\_class\\_methods.asp](https://www.w3schools.com/java/java_class_methods.asp)
- [https://www.w3schools.com/java/java\\_constructors.asp](https://www.w3schools.com/java/java_constructors.asp)
- [https://www.w3schools.com/java/java\\_modifiers.asp](https://www.w3schools.com/java/java_modifiers.asp)
- [https://www.w3schools.com/java/java\\_encapsulation.asp](https://www.w3schools.com/java/java_encapsulation.asp)
- <https://www.geeksforgeeks.org/object-oriented-programming-oops-concept-in-java/>
- <https://docs.oracle.com/javase%2Ftutorial%2F/java/concepts/index.html>  
+ всі посилання що є у змісті ліворуч
- <https://docs.oracle.com/javase%2Ftutorial%2F/java/javaOO/index.html> +  
всі посилання що є у змісті ліворуч до Nested Classes

#### Завдання № 5.

Знайти кількість входжень заданого підрядка (substring) у двовимірну матрицю типу String\*String.

#### Завдання № 6.

Створити конвертер валют для: гривня, американський долар, канадський долар, євро. Введення даних у програму відбувається у вигляді “100 UAH into USD”, або в іншому зручному форматі.

## Завдання № 7.

Відтворіть перекази між банківськими рахунками. Реалізуйте клас Банк та Рахунок (BankAccount). Один користувач може мати декілька рахунків в одному чи кількох банках. Рахунок прив'язаний до заданої під час створення банку валюти. При переказі між рахунками з різною валютою необхідно конвертувати переказ до валюти цільового рахунку. Комісії:

- Переказ між власними рахунками в межах одного банку: 0%
- Переказ між власними рахунками у різних банках: 2%
- Переказ між рахунками різних користувачів в одному банку: 3%
- Переказ між рахунками різних користувачів у різних банках: 6%

Вимоги до завдань:

1. Кожне завдання повинне бути виконане у відповідному класі.

Наприклад, `public class Task1;`

2. Варто розділяти дані, логіку та виведення;

3. Локальні змінні повинні бути `final` тоді, коли це можливо;

4. Вхідні дані можна як вводити з клавіатури під час виконання програми, так і використати задані вручну у методі `main`.

## Контрольні запитання

1. Примітиви `java`
2. Цикли `java`
3. Як виглядає сигнатура методу
4. Тип методу `void`
5. Яким відношенням повинне відповідати `equals`
6. Для чого потрібний `equals` і `hashCode`
7. Методи класу `Object`

8. Клас String та його особливості
9. Інкапсуляція, успадкування, поліморфізм, абстракція
10. Різниця між об'єктом класу і екземпляром класу
11. Конструктор
12. Надклас для всіх класів
13. Передача примітивів та об'єктів у методи
14. Контракт між equals та hashCode
15. String pool
16. Порядок ініціалізації полів класу
17. Модифікатори доступу і яка різниця між ними
18. Абстрактні класи
19. Overload vs override