Об’єктно орієнтоване програмування

Лабораторна робота №1

**Типи даних, умови, цикли та масиви**

**Мета:** Використовуючи теоретичне підґрунтя про типи даних, що використовуються у мові Java, правила використання умовних переходів, написання циклів та використання масивів виконати дії що будуть вказано в завданні до лабораторної роботи.

**Хід роботи**

Завдання №1

Напишіть програму, яка прочитає рядок із System.in і виведе повідомлення: "Hello, прочитаний рядок". Якщо у вхідному рядку кілька слів, потрібно надрукувати їх усі.

Код програми:

import java.util.Scanner;

public class Task1 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        String text = scan.nextLine();

        System.out.println("Hello, " + text);

        scan.close();

    }

}

Завдання №2

Напишіть програму, яка отримує на вхід ціле число. Це число є поточний добовий час, який обчислюється як кількість секунд, що пройшли з півночі. Потім програма повинна надрукувати значення екрану електронного годинника для цього часу.

Вхідне значення отримується через System.in. Вихідне значення має бути надруковане у System.out.

Гарантується, що вхідне значення — невід'ємне.

Формат виводу: h:mm:ss (можливі значення: [0:00:00; 23:59:59]).

Код програми:

import java.util.Scanner;

public class Task2 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        int time = scan.nextInt();

*while* (time <= 0) {

            time = scan.nextInt();

        }

        int hours = time / 3600;

        int minutes = (time % 3600) / 60;

        int seconds = time % 60;

        System.out.println("Time is: " + hours + ":" + minutes + ":" + seconds);

        scan.close();

    }

}

Завдання №3

Напишіть програму, яка запитує на вхід число-пароль. Якщо воно дорівнює секретній кількості, програма виводить: "Hello, Agent", інакше друкує: "Access denied".

Секретний пароль зберігається в final static int password.

import java.util.Scanner;

public class Task3 {

    private static final int password = 550022;

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        System.out.print("Password: ");

        int secretPassword = scan.nextInt();

*if* (secretPassword == password) {

            System.out.println("Hello, Agent");

        }

*else* {

            System.out.println("Access denied");

        }

        scan.close();

    }

}

Завдання №4

Напишіть просту програму, яка запитує кількість незнайомців, з якими потрібно зустрітися, рядково читає імена незнайомців, по рядках виводить: "Hello, (ім'я незнайомця)" для кожного незнайомця.

Доповнити програму перевірками на кількість, що є від’ємною, та рівна 0.

Код програми:

import java.util.Scanner;

public class Task4 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        System.out.print("Amount of strangers: ");

        int amountStrangers = scan.nextInt();

*while* (amountStrangers <= 0) {

            System.out.print("Amount cant be 0 or negative. Try again: ");

            amountStrangers = scan.nextInt();

        }

        scan.nextLine();

*for* (int i = 0; i < amountStrangers; i++) {

            System.out.println("\nWhat's a stranger name?");

            String name = scan.nextLine();

            System.out.println("\nHello, " + name);

        }

        scan.close();

    }

}

Завдання №5

Наведено таку умову: равлик піднімається по дереву на a футів на день. Потім щоночі равлик сповзає вниз на b футів. Висота дерева — h футів.

Напишіть програму з підрахунком кількості днів, які знадобляться равлику, щоб дістатися до вершини дерева.

Програма читає a, b, h рядково. Вхідні значення гарантовано є додатніми цілими числами.

Якщо равлик не може дістатися до вершини дерева, має виводитися повідомлення: Impossible

Код програми:

import java.util.Scanner;

public class Task5 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        System.out.print("Type a, b, h: ");

        int a = scan.nextInt();

        int b = scan.nextInt();

        int h = scan.nextInt();

*if* (a <= b) {

            System.out.println("Impossible");

            System.exit(0);

        }

        int days = 0;

        int difference = 0;

*while* (true) {

            difference += a;

            days++;

*if* (difference >= h)

*break*;

            difference -= b;

        }

        System.out.println(days);

        scan.close();

    }

}

Завдання №6

Дано така умова: компанія друзів відвідує ресторан. Вони вирішили розділити рахунок порівну та додати 10 відсотків від загальної суми рахунку як чайові. Далі друзі покривають рівними частинами загальну суму платежу.

Напишіть програму, яка зчитує загальну суму рахунку та кількість друзів, а потім виводить розмір частини оплати.

Доповнити програму перевірками на від’ємну суму рахунку та нульову кількість друзів.

Код програми:

import java.util.Scanner;

public class Task6 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        System.out.print("Money to pay: ");

        int toPay = scan.nextInt();

        System.out.print("Amout of people: ");

        int amout = scan.nextInt();

*if* (toPay <= 0 || amout <= 0) {

            System.exit(0);

        }

        double tip = (toPay / amout) + (toPay / amout)\*0.1;

        System.out.println("Tip is " + tip);

        scan.close();

    }

}

Завдання №7

Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення і знаходить максимальне з них. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її елементом. Послідовність гарантовано містить хоча б одне значення.

Код програми:

import java.util.Scanner;

public class Task7 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        int max = -1;

        System.out.print("Type a nums: ");

*while* (true) {

            int num = scan.nextInt();

*if* (num == 0) {

*break*;

            }

*if* (num > max) {

                max = num;

            }

        }

        System.out.println("Max num is " + max);

        scan.close();

    }

}

Завдання №8

Напишіть програму, яка зчитує послідовність цілих значень зі стандартного введення та знаходить середнє значення. Ви повинні зчитувати значення послідовності, доки наступне не стане 0. Нульове значення означає кінець вхідної послідовності та не є її частиною. Послідовність гарантовано містить хоча б одне значення.

Код програми:

import java.util.Scanner;

public class Task8 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        int sum = 0;

        int count = 0;

        System.out.print("Type a nums: ");

*while* (true) {

            int num = scan.nextInt();

*if* (num == 0) {

*break*;

            }

            sum += num;

            count++;

        }

        double average = (double) sum / count;

        System.out.println("Result: " + average);

        scan.close();

    }

}

Завдання №9

Напишіть програму для вирішення квадратних рівнянь. При виконанні врахуйте залежність значення дискримінанту до кількості результатів.

Код програми:

public class Task9 {

    public static void main(String[] args) {

        Scanner scan = *new* Scanner(System.in);

        System.out.print("a, b, c: ");

        double a = scan.nextDouble();

        double b = scan.nextDouble();

        double c = scan.nextDouble();

        double discriminant = b\*b - 4\*a\*c;

*if* (discriminant > 0) {

            double root1 = (-b + Math.sqrt(discriminant)) / (2 \* a);

            double root2 = (-b - Math.sqrt(discriminant)) / (2 \* a);

            System.out.println("Results: " + root1 + " " + root2);

        } *else* *if* (discriminant == 0) {

            double root = -b / (2 \* a);

            System.out.println("Result: " + root);

        } *else* {

            System.out.println("No roots");

        }

        scan.close();

    }

}

Завдання №10

Напишіть програму, що містить метод max(…). Правильна реалізація має отримати масив значень типу int та повернути його максимальне значення. Вхідний масив гарантовано не буде порожнім або null. Метод max не повинен змінювати масив.

Код програми:

public class Task10 {

    public static void main(String[] args) {

        int[] nums = {1,2,3,7,1,9,7,2};

        System.out.println(max(nums));

    }

    public static int max(int[] array) {

        int max = 0;

*for* (int i = 0; i < array.length; i++) {

*if* (array[i] > max)

                max = array[i];

        }

*return* max;

    }

}

Завдання №11

Напишіть програму, що містить метод sum. Правильна реалізація має отримати масив значень int та повернути суму парних чисел. Якщо цей масив дорівнює нулю або порожній, метод повертає 0. Метод sum не повинен змінювати масив.

Код програми:

public class Task11 {

    public static void main(String[] args) {

        int[] nums = {1,2,3,6,7,8,2,10};

        System.out.println("Sum is " + sum(nums));

    }

    public static int sum(int[] array) {

        int sum = 0;

*for* (int i = 0; i < array.length; i++) {

*if* (array[i] % 2 == 0) {

                sum += array[i];

            }

*else* {

*continue*;

            }

        }

*return* sum;

    }

}

Завдання №12

Напишіть програму, що містить метод getSumCheckArray. Правильна реалізація повинна отримати масив значень int і повернути масив логічних значень, де кожен елемент є результатом перевірки: чи є вихідний елемент сумою двох попередніх елементів у даному масиві.

Цей масив гарантовано не дорівнює null.

Метод повертає масив логічних значень, де кожен елемент є результатом для відповідного елемента в даному масиві.

Перші два елементи логічного масиву завжди false.

Гарантується, що довжина масиву становить 2 або більше.

Код програми:

import java.util.Arrays;

public class Task12 {

    public static void main(String[] args) {

        int[] array = {1, -1, 0, 4, 6, 10, 15, 25};

        boolean[] boolArray = getSumCheckArray(array);

        System.out.println("Result: " + Arrays.toString(boolArray));

    }

    public static boolean[] getSumCheckArray(int[] array) {

        boolean[] boolArray = *new* boolean[array.length];

        boolArray[0] = false;

        boolArray[1] = false;

*for* (int i = 2; i < array.length; i++) {

*if* (array[i-1] + array[i-2] == array[i])

                boolArray[i] = true;

*else*

                boolArray[i] = false;

        }

*return* boolArray;

    }

}

Завдання №13

Напишіть програму, що містить метод removeLocalMaxima(…). Правильна реалізація має отримати масив значень типу int і повернути копію даного масиву з усіма видаленими локальними максимумами. Вихідний масив не можна змінювати.

Локальний максимум — це елемент, який більший за будь-який з його сусідніх елементів. Вам необхідно видалити елементи, які є локальними максимумами у вихідному масиві.

Код програми:

import java.util.Arrays;

public class Task13 {

    public static void main(String[] args) {

        int[] nums = {18, 1, 3, 6, 7, -5};

        int[] result = removeLocalMaxima(nums);

        System.out.println("Result: " + Arrays.toString(result));

    }

    public static int[] removeLocalMaxima(int[] array) {

        boolean[] isLocalMaxima = *new* boolean[array.length];

        int count = 0;

*for* (int i = 1; i < (array.length) - 1; i++) {

*if* (array[i] > array[i+1] && array[i] > array[i-1]) {

                isLocalMaxima[i] = true;

                count++;

            }

        }

*if* (array[0] > array[1]) {

            isLocalMaxima[0] = true;

            count++;

        }

*if* (array[array.length-1] > array[array.length-2]) {

            isLocalMaxima[isLocalMaxima.length-1] = true;

            count++;

        }

        int[] result = *new* int[array.length - count];

        int j = 0;

*for* (int i = 0; i < array.length; i++) {

*if* (!isLocalMaxima[i])

                result[j++] = array[i];

        }

*return* result;

    }

}

Завдання №14

Напишіть програму що містить два методи, що реалізують наступний функціонал:

void cycleSwap(int[] array) - Зсуває всі елементи в даному масиві вправо на 1 позицію. В цьому випадку останній елемент масиву стає першим. Наприклад, 1 3 2 7 4 стає 4 1 3 2 7.

void cycleSwap(int[] array, int shift) - Зсуває всі елементи в заданому масиві вправо на shift позицій. Гарантується, що значення зсуву невід'ємне і не більше за довжину масиву. Наприклад, 1 3 2 7 4 зі зсувом 3 стає 2 7 4 1 3.

Код програми:

public class Task14 {

    public static void main(String[] args) {

        int[] array = {1,3,2,7,4};

        cycleSwapByOne(array);

        System.out.println();

        int shift = 3;

        cycleSwap(array, shift);

    }

    public static void cycleSwapByOne(int[] array) {

        int lastElem = array[array.length - 1];

*for* (int i = array.length - 1; i > 0; i--) {

            array[i] = array[i - 1];

        }

        array[0] = lastElem;

        printArray(array);

    }

    public static void cycleSwap(int[] array, int shift1) {

        int shift = shift1-1;

*for* (int j = 0; j < shift; j++) {

            int lastElem = array[array.length - 1];

*for* (int i = array.length - 1; i > 0; i--)

                array[i] = array[i - 1];

            array[0] = lastElem;

        }

        printArray(array);

    }

    public static void printArray(int[] array) {

*for* (int i = 0; i < array.length; i++) {

            System.out.print(array[i] + " ");

        }

    }

}

Посилання на github: [link](https://github.com/7878seveneight/OOP-KB-222-Petr-Ratozii/tree/main/lab01)

**Висновок:**

Використовуючи теоретичне підґрунтя про типи даних, що використовуються у мові Java, правила використання умовних переходів, написання циклів та використання масивів виконав дії що були вказані в завданні до лабораторної роботи.