

ЗАТВЕРДЖУЮ

Завідувач кафедри управління проєктами, інформаційних технологій та телекомунікацій, д.т.н., професор

Євген Мартин

«__» _____ 20__ р.

МЕТОДИЧНА РОЗРОБКА

для проведення практичного заняття з студентами (курсантами) 1-го курсу спеціальності 122 «Комп'ютерні науки» з дисципліни «Основи програмування»

Практичне заняття 5.14: «Застосування потоків введення/виведення та рядків в програмуванні»

Мета: здобути навички щодо застосування класів Scanner, String та їх екземплярів в процесі створення найпростіших програм.

Час: 2 годин

Місце заняття: комп'ютерна лабораторія

Матеріальне забезпечення:

- ПК з відповідним ПЗ та доступом до мережі Internet;
- Методичні рекомендації щодо виконання практичної роботи.

Після практичного заняття студенти (курсанти) повинні:

вміти: створювати елементарні програми застосовуючи потоки введення та рядки.

знати: принцип роботи та особливості застосування потоків введення/виведення, а також рядків.

Література:

1. Віртуальне навчальне середовище ЛДУ БЖД «Віртуальний університет» [Електронний ресурс]. – Доступний з : <http://virt.ldubgd.edu.ua/course/view.php?id=1778>
2. Програмування на Java (рос.) [Електронний ресурс]. – <http://kostin.ws/java>

План заняття:

1. Написання найпростіших програм використовуючи класи та екземпляри класів Scanner, String
2. Видача індивідуальних практичних завдань

1. Написання найпростіших програм використовуючи класи та екземпляри класів Scanner, String

Для реалізації перших застосунків використовуючи потоки введення/виведення та рядки розглянемо декілька розповсюджених прикладів.

Приклад 1: Створити програму виводу на екран повідомлення про парність (непарність) цілого числа введеного з клавіатури. Передбачити можливість повідомлення користувача про введення не цілого числа:

```
import java.util.Scanner;

public class Main{

    public static void main(String[] args) {

        int n;
        System.out.print("Введіть ціле число: ");
        Scanner sc = new Scanner(System.in);
        if(sc.hasNextInt()) {
            n = sc.nextInt();
            if(n%2==0) {
                System.out.println("Введене число - парне");
            } else {
                System.out.println("Введене число - не парне");
            }
        } else {
            System.out.println("Ви ввели не ціле число");
        }
    }
}
```

Приклад 2: Створити програму розрахунку та виведення на екран суми двох цілих чисел введених з клавіатури. Передбачити можливість повідомлення користувача про некоректне введення чисел:

```
import java.util.Scanner;

public class Main2 {

    public static void main(String[] args) {

        int a, b, c;
        System.out.println("Для виконання програми введіть два цілих числа");
        Scanner sc = new Scanner(System.in);
        System.out.print("Введіть перше число: ");
        if (sc.hasNextInt()) {
            a = sc.nextInt();
            System.out.print("Введіть друге число: ");
            if (sc.hasNextInt()) {
                b = sc.nextInt();
                c = a + b;
                System.out.print("Сума введених чисел = " + c);
            } else {
                System.out.print("Ви ввели не ціле число");
            }
        } else {
            System.out.print("Ви ввели не ціле число");
        }
    }
}
```

```

        System.out.print("Ви ввели не ціле число");
    }
}

```

Приклад 3: Створити програму визначення та виводу на екран меншого по модулю з трьох введених цілих чисел. Передбачити можливість повідомлення користувача про некоректне введення чисел:

```

import java.util.Scanner;

public class Main3 {

    public static void main(String[] args) {

        int a, b, c, moda, modb, modc, min;
        Scanner sc = new Scanner(System.in);
        System.out.print("Введіть перше ціле число: ");
        if (sc.hasNextInt()) {
            a = sc.nextInt();
            System.out.print("Введіть друге ціле число: ");
            if (sc.hasNextInt()) {
                b = sc.nextInt();
                System.out.print("Введіть третє ціле число: ");
                if (sc.hasNextInt()) {
                    c = sc.nextInt();
                    moda = Math.abs(a);
                    modb = Math.abs(b);
                    modc = Math.abs(c);
                    if (moda <= modb && moda <= modc) {
                        min = a;
                    } else if (modb <= moda && modb <= modc) {
                        min = b;
                    } else {
                        min = c;
                    }
                    System.out.println(min);
                } else {
                    System.out.println("Ви ввели не ціле число");
                }
            } else {
                System.out.println("Ви ввели не ціле число");
            }
        } else {
            System.out.println("Ви ввели не ціле число");
        }
    }
}

```

Приклад 4: Створити програму перевірки належності введеного слова з п'яти букв до паліндрому («ротатор», «комок» тощо). Передбачити можливість повідомлення користувача про некоректне введення слова (більше або менше п'яти букв):

```

import java.util.Scanner;

public class Main4 {

    public static void main(String[] args) {

        String s1,s2;
        char a0,a1,a2,a3,a4;
        Scanner sc = new Scanner(System.in);
        System.out.print("Введіть слово з п'яти букв: ");
        if(sc.hasNext()) {
            s1 = sc.next();
            if(s1.length()==5) {
                a0 = s1.charAt(0);
                a1 = s1.charAt(1);
                a2 = s1.charAt(2);
                a3 = s1.charAt(3);
                a4 = s1.charAt(4);
                s2 = a4+" "+a3+" "+a2+" "+a1+" "+a0;
                System.out.println(s2);
                if(s1.equals(s2)) {
                    System.out.println("Введене слово "+s1+" є
поліндромом");
                } else {
                    System.out.println("Введене слово "+s1+" не є
поліндромом");
                }
            } else {
                System.out.print("Ви ввели слово не з п'яти букв
"+s1);
            }
        } else {
            System.out.print("Ви нічого не ввели");
        }
    }
}

```

2. Видача індивідуальних практичних завдань

Вкінці практичного заняття студенти отримують індивідуальне практичне завдання, результат виконання якого необхідно завантажити в віртуальне навчальне середовище. Для виконання завдання слід нагадати теоретичний матеріал лекції. Індивідуальне практичне завдання полягає у створенні програм із застосуванням потоків введення/виведення інформації та методів класу String.

Завдання 1. З використанням методу concat() класу String скласти речення з п'яти окремих слів введених користувачем.

Завдання 2. Створити програму визначення та виведення на екран першої літери п'яти довільно введених користувачем слів із використанням методу substring(pos1, pos2).

Завдання 3. Створити програму виводу на екран повідомлення про більше дробове число з трьох введених користувачем. Передбачити можливість повідомлення користувача про введення не дробового числа.

Завдання 4. Створити програму перевірки ідентичності двох введених з клавіатури імен без урахування регістру.

Завдання 5. Створити програму визначення більшого за кількістю символів рядка з двох введених користувачем.

Файли з кодами написаних програм (розширення .java) потрібно завантажувати у віртуальне середовище (категорія контрольна частина). За бажанням, кожне завдання може завантажуватись у вигляді окремого файлу, або усі завдання в одному.

Методичну розробку розробив:

Заступник начальника кафедри УПІТтаТ
підполковник служби цивільного захисту

Олександр Придатко

Методична розробка обговорена на засіданні кафедри УПІТтаТ

Протокол № _____ від "___" _____ 20__ р.