**Лабораторна робота №1**

**Тема**: 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**

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

**Виконання:** створюється об’єкт класу Scanner для читання вводу користувача, користувач вводить рядок, який потім використовується для виводу повідомлення "Hello, прочитаний рядок".

**Текст програми**:

import java.util.Scanner;

public class task01 {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        String input = scanner.nextLine();

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

    }

**Результат виконання:**



**Завдання 2**

**Умова:** 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]).

**Виконання:** задана кількість секунд проходить такі етапи:

* 3600, після чого береться залишок від ділення на 24
* Береться залишок від 3600, після чого ділиться на 60
* Береться залишок від 60

Всі результати заносяться в змінні, і відповідають кількості годин, хвилин та секунд.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task02 {  public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int seconds = scanner.nextInt();          int h = (seconds / 3600) % 24;          int m = (seconds % 3600) / 60;          int s = seconds % 60;          System.out.printf("%d:%02d:%02d%n", h, m, s);      }  } |

**Результат виконання:**



**Завдання 3**

**Умова:** 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..

**Виконання:** за межами методу main в класі task\_3 створюється константа, що доступна всередині класу, після чого в користувача запитується пароль та перевіряється, чи співпадає він із значенням константи, якщо так, то програма виводить: "Hello, Agent", інакше друкує: "Access denied".

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task03 {      final static int password = 12345;      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int input = scanner.nextInt();          if (input == password) {              System.out.println("Hello, Agent");          } else {              System.out.println("Access denied");          }      }  } |

**Результат виконання:**





**Завдання 4**

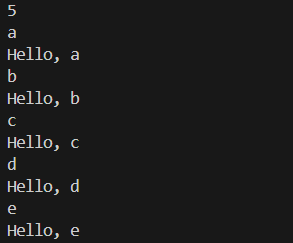
**Умова:** Напишіть просту програму, яка запитує кількість незнайомців, з якими потрібно зустрітися, рядково читає імена незнайомців, по рядках виводить: "Hello, (ім'я незнайомця)" для кожного незнайомця. Доповнити програму перевірками на кількість, що є від’ємною, та рівна 0.

**Виконання:** програма запитує кількість незнайомців, перевіряє щоб це число було не від’ємним та не рівним нулю, після чого відповідає на кожне ім’я з привітанням

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task04 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int count = scanner.nextInt();          scanner.nextLine();          if (count < 0) {              System.out.println("Error: negative number");          } else if (count == 0) {              System.out.println("No one to greet.");          } else {              for (int i = 0; i < count; i++) {                  String name = scanner.nextLine();                  System.out.println("Hello, " + name);              }          }      }  } |

**Результат виконання:**



**Завдання 5**

**Умова:** 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.

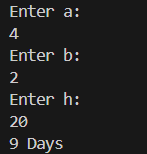
**Виконання:** програма отримує дані, перевіряє чи можливий взагалі підйом, перевіряючи загальну висоту підйому за добу. Потім перевіряється умова, якщо равлик підніметься за один день, а потім використовується формула для розрахунку кількості днів.

Формула працює так: від загальної висоти віднімається дистанція сповзання (це враховує момент, коли равлик досягне точки, де наступний підйом буде останнім), після чого ця відстань ділиться на висоту підйому за цілу добу. Результат завжди округлюється вгору.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task05 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          System.out.println("Enter a:");          int a = scanner.nextInt();          System.out.println("Enter b:");          int b = scanner.nextInt();          System.out.println("Enter h:");          int h = scanner.nextInt();          if (a <= b && a < h) {              System.out.println("Impossible");          }           else if (a >= h) {              System.out.println(1);           }           else {              int days = (int) Math.ceil( (double)(h - b) / (a - b));              System.out.println(days + " Days");          }      }  } |

**Результат виконання:**



**Завдання 6**

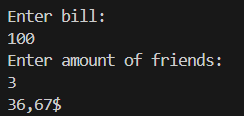
**Умова:** Дано така умова: компанія друзів відвідує ресторан. Вони вирішили розділити рахунок порівну та додати 10 відсотків від загальної суми рахунку як чайові. Далі друзі покривають рівними частинами загальну суму платежу. Напишіть програму, яка зчитує загальну суму рахунку та кількість друзів, а потім виводить розмір частини оплати. Доповнити програму перевірками на від’ємну суму рахунку та нульову кількість друзів.

**Виконання:** зчитується сума рахунку та кількість друзів, після чого обраховується розмір частини оплати з урахуванням 10% чайових.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task06 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          System.out.println("Enter bill:");          double bill = scanner.nextDouble();          System.out.println("Enter amount of friends:");          int friends = scanner.nextInt();          if (bill < 0) {              System.out.println("Error: negative bill");          } else if (friends <= 0) {              System.out.println("Error: number of friends must be positive");          } else {              double total = bill \* 1.1;              System.out.println(String.format("%.02f$", (double)total / friends));          }      }  } |

**Результат виконання:**



**Завдання 7**

**Умова:** 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.

**Виконання:** сканер зчитує дані введеного масиву, і порівнює по черзі його зі змінною(за замовчуванням вибрано мінімальне значення int), якщо зчитане число більше ніж змінна, змінна перезаписується чим числом. Якщо зчитується нуль, цикл переривається й виводиться змінна.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task07 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int max = Integer.MIN\_VALUE;          while (true) {              int value = scanner.nextInt();              if (value == 0) break;              if (value > max) max = value;          }          System.out.println(max);      }  } |

**Результат виконання:**



**Завдання 8**

**Умова:** 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.

**Виконання:** сканер зчитує число, і додає її до змінної sum. Змінна count використовується для підрахунку кількості чисел, і збільшується на одиницю за кожну ітерацію циклу. Після розривання циклу sum ділиться на count і виводиться результат.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task08 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int sum = 0, count = 0;          while (true) {              int value = scanner.nextInt();              if (value == 0) break;              sum += value;              count++;          }          System.out.println(sum / count);      }  } |

**Результат виконання:**



**Завдання 9**

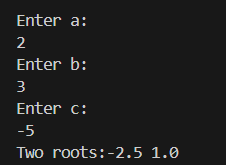
**Умова:** Напишіть програму для вирішення квадратних рівнянь. При виконанні врахуйте залежність значення дискримінанту до кількості результатів.

**Виконання:** після обрахунку дискримінанту перевірками визначається кількість коренів та формули для їхнього обрахунку.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task09 {      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          System.out.println("Enter a:");          double a = scanner.nextDouble();          System.out.println("Enter b:");          double b = scanner.nextDouble();          System.out.println("Enter c:");          double c = scanner.nextDouble();          double d = b \* b - 4 \* a \* c;          if (d < 0) {              System.out.println("no roots");          } else if (d == 0) {              System.out.println("One root:" + -b / (2 \* a));          } else {              double r1 = (-b - Math.sqrt(d)) / (2 \* a);              double r2 = (-b + Math.sqrt(d)) / (2 \* a);              System.out.println("Two roots:" + r1 + " " + r2);          }      }  } |

**Результат виконання:**



**Завдання 10**

**Умова:** 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.

**Виконання:** для цього завдання було створено 2 методи, перший для створення массиву, а другий для пошуку максимального значення у послідовності.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task10 {      public static int max(int[] array) {          int max = array[0];          for (int val : array) {              if (val > max) {                  max = val;              }          }          return max;      }  public static int[] input\_data(){      Scanner input = new Scanner(System.in);      System.out.print("Enter your numbers: ");      String[] parts = input.nextLine().split(" ");      int[] array = new int[parts.length];      for (int i = 0; i < parts.length; i++) {          array[i] = Integer.parseInt(parts[i]);      }      return array;  }      public static void main(String[] args) {          int array[] = input\_data();          System.out.println("Max: " + max(array));      }  } |

**Результат виконання:**



**Завдання 11**

**Умова:** 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.

**Виконання:** відповідно до минулого завдання, одна функція зчитує дані та формує масив, друга перебирає значення, перевіряє парність та сумує значення.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  public class task11 {      public static int sum(int[] array) {          if (array == null || array.length == 0) return 0;          int sum = 0;          for (int val : array) {              if (val % 2 == 0) {                  sum += val;              }          }          return sum;      }      public static int[] input\_data(){          Scanner input = new Scanner(System.in);          System.out.print("Enter your numbers: ");            String[] parts = input.nextLine().split(" ");          int[] array = new int[parts.length];          for (int i = 0; i < parts.length; i++) {              array[i] = Integer.parseInt(parts[i]);          }          return array;      }      public static void main(String[] args) {          int[] array = input\_data();          System.out.println("Sum of even: " + sum(array));      }  } |

**Результат виконання:**



**Завдання 12**

**Умова:** 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. Гарантується, що довжина масиву становить 2 або більше. 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.

**Виконання:** введений масив перезбирається в масив типу int, після чого його отримує метод getSumCheckArray та повертає масив типу bool.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  import java.util.Arrays;  public class task12 {      public static boolean[] getSumCheckArray(int[] array) {          boolean[] result = new boolean[array.length];          result[0] = false;          result[1] = false;          for (int i = 2; i < array.length; i++) {              result[i] = (array[i] == array[i - 1] + array[i - 2]);          }          return result;      }      public static int[] input\_data(){          Scanner input = new Scanner(System.in);          System.out.print("Enter your numbers: ");            String[] parts = input.nextLine().split(" ");          int[] array = new int[parts.length];          for (int i = 0; i < parts.length; i++) {              array[i] = Integer.parseInt(parts[i]);          }          return array;      }      public static void main(String[] args) {          int[] input = input\_data();          boolean[] result = getSumCheckArray(input);          System.out.println(Arrays.toString(result));      }  } |

**Результат виконання:**



**Завдання 13**

**Умова:** 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.

* Розмір цього масиву гарантовано більше 1.
* Розмір цього масиву гарантовано більше 1.
* Якщо масив не має локальних максимумів, ви повинні повернути його копію без змін.
* Ви можете використовувати методи java.util.Arrays.\*.

**Виконання:** в методі removeLocalMaxima спочатку рахується кількість локальних максимумів, щоб створити масив потрібного розміру, після цього в нього заносяться значення.

**Текст програми**:

|  |
| --- |
| import java.util.Scanner;  import java.util.Arrays;  public class task13 {      public static int[] removeLocalMaxima(int[] array) {          if (array == null || array.length == 0) {              return new int[0];          }            int count = 0;          for (int i = 0; i < array.length; i++) {              if (i > 0 && array[i] <= array[i - 1] ||                  i < array.length - 1 && array[i] <= array[i + 1] ||                  i == 0 && array.length > 1 && array[i] <= array[i + 1] ||                  i == array.length - 1 && array.length > 1 && array[i] <= array[i - 1]) {                  count++;              }          }          int[] result = new int[count];          int index = 0;          for (int i = 0; i < array.length; i++) {              if (i > 0 && array[i] <= array[i - 1] ||                  i < array.length - 1 && array[i] <= array[i + 1] ||                  i == 0 && array.length > 1 && array[i] <= array[i + 1] ||                  i == array.length - 1 && array.length > 1 && array[i] <= array[i - 1]) {                  result[index++] = array[i];              }          }            return result;      }      public static int[] input\_data(){          Scanner input = new Scanner(System.in);          System.out.print("Enter your numbers: ");            String[] parts = input.nextLine().split(" ");          int[] array = new int[parts.length];          for (int i = 0; i < parts.length; i++) {              array[i] = Integer.parseInt(parts[i]);          }          return array;      }      public static void main(String[] args) {          int[] input = input\_data();          int[] result = removeLocalMaxima(input);          System.out.println(Arrays.toString(result));      }  } |

**Результат виконання:**



**Завдання 14**

**Умова:** 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.

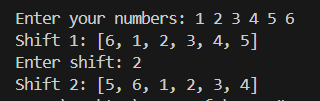
void cycleSwap(int[] array, int shift) - Зсуває всі елементи в заданому масиві вправо на shift позицій. Гарантується, що значення зсуву невід'ємне і не більше за довжину масиву. Наприклад, 1 3 2 7 4 зі зсувом 3 стає 2 7 4 1 3..

**Виконання:** у вказаних методах зміщення відбувається за допомогою повного відзеркалення, і відзеркалення 2 груп: shift елементів, та решти.

**Текст програми**:

|  |
| --- |
| import java.util.Arrays;  import java.util.Scanner;  public class task14 {      public static int[] input\_data() {          Scanner input = new Scanner(System.in);          System.out.print("Enter your numbers: ");          String[] parts = input.nextLine().split(" ");          int[] array = new int[parts.length];          for (int i = 0; i < parts.length; i++) {              array[i] = Integer.parseInt(parts[i]);          }          return array;      }      public static void cycleSwap(int[] array) {          if (array.length <= 1) return;          int last = array[array.length - 1];          for (int i = array.length - 1; i > 0; i--) {              array[i] = array[i - 1];          }          array[0] = last;      }      public static void cycleSwap(int[] array, int shift) {          if (array.length <= 1 || shift <= 0) return;          shift %= array.length;            for (int i = 0, j = array.length - 1; i < j; i++, j--) {              int tmp = array[i];              array[i] = array[j];              array[j] = tmp;          }            for (int i = 0, j = shift - 1; i < j; i++, j--) {              int tmp = array[i];              array[i] = array[j];              array[j] = tmp;          }          for (int i = shift, j = array.length - 1; i < j; i++, j--) {              int tmp = array[i];              array[i] = array[j];              array[j] = tmp;          }      }      public static void main(String[] args) {          Scanner scanner = new Scanner(System.in);          int[] array = input\_data();          int[] array1 = Arrays.copyOf(array, array.length);          cycleSwap(array1);          System.out.println("Shift 1: " + Arrays.toString(array1));          System.out.print("Enter shift: ");          int shift = scanner.nextInt();          int[] array2 = Arrays.copyOf(array, array.length);          cycleSwap(array2, shift);          System.out.println("Shift " + shift + ": " + Arrays.toString(array2));      }  } |

**Результат виконання:**



**Висновок:** під час виконання роботи я використовуючи теоретичне підґрунтя про типи даних, що використовуються у мові Java, правила використання умовних переходів, написання циклів та використання масивів, виконував завдання до лабораторної роботи.