**Лабораторна робота №2**

**Тема:** Розробка програм з використанням умовних операторів, операторів циклу мови Java.

**Мета:**

* Сформувати уявлення про Java програми.
* Набути навиків у створенні програм з використанням умовних операторів, операторів циклу та масивів.
* Розвинути навики у створенні програм в середовищі Eclipse.

**Теоретичні відомості**

Умовний оператор у мові Java записується так:

if (логічний вираз) оператор1 else оператор2

і діє наступним чином. Спочатку обчислюється логічний вираз. Якщо результат true, то діє оператор1 і на цьому дія умовного оператора завершується, оператор2 не діє, далі буде виконуватися наступний за if оператор. Якщо результат false, то діє оператор2, при цьому оператор1 взагалі не виконується.

Умовний оператор може бути скороченим:

if (логічний вираз) оператор1

і у випадку false не виконується нічого.

Основний оператор циклу — оператор while — виглядає так:

while (логічний вираз) оператор

Спочатку обчислюється логічний вираз; якщо його значення true, то виконується оператор, що утворює цикл. Потім знову обчислюється логічний вираз і діє оператор, і так до тих пір, поки не отримається значення false. Якщо логічний вираз з самого початку рівний false, то оператор не буде виконуватися ні разу. Попередня перевірка забезпечує безпеку виконання циклу, дозволяє уникнути переповнення, ділення на нуль і інші неприємності. Тому оператор while являється основним, а в деяких мовах і єдиним оператором циклу.

Другий оператор циклу — оператор do-while — має вигляд:

do оператор while (логічний вираз)

Тут спочатку виконується оператор, а потім відбувається обчислення логічного виразу логічний вираз. Цикл виконується, поки логічний вираз залишається рівним true.

Третій оператор циклу – оператор for – виглядає так:

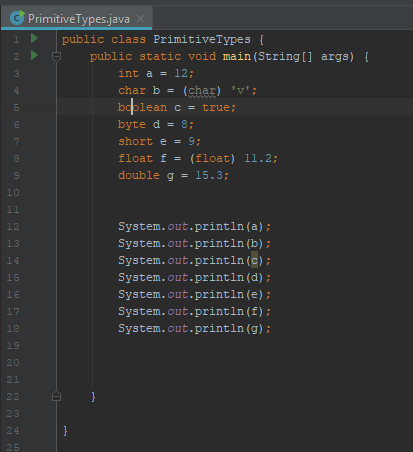
for (списокВир1; логВир; списокВир2) оператор

Перед виконанням циклу обчислюється список виразів списокВир1. Це нуль або декілька виразів, перерахованих через кому. Вони обчислюються зліва направо, і в наступному виразі уже можна використовувати результат попереднього виразу. Як правило, тут задаються початкові значення змінних циклу.

Потім обчислюється логічний вираз логВир. Якщо він істинний, true, то діє оператор, потім обчислюються зліва направо вирази із списку виразів списокВир2. Далі знову перевіряється логВир. Якщо він істинний, то виконується оператор і списокВир2 і т. д. Як тільки логВир стане рівним false, виконання циклу закінчується.

**Хід роботи**

1. Створено новий клас з назвою PrimitiveTypes та добавлено точку входу в програму – метод main().



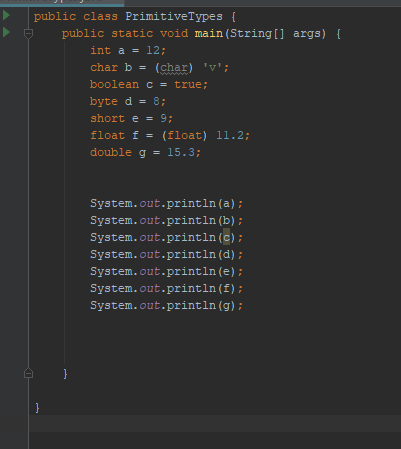
1. Добавлено оголошення змінної “a” типу int, присвоєно початкове значення та виведено на екран:

**public** **static** **void** main(String[] args) {

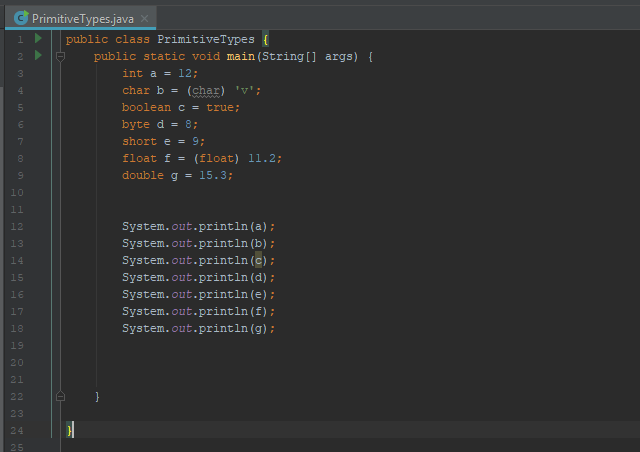
**int** a = 12;

System.***out***.println(a);

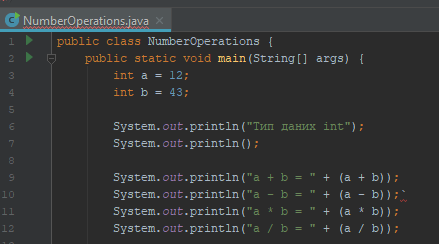
}



1. Змінено текст програми, оголосивши змінні для всіх примітивних типів (boolean, char, byte, short, long, double, float). Збережено текст програми та вказано її разом із результатами виконання у звіті.



1. Створено новий клас з назвою NumberOperations та добавлено точку входу в програму – метод main().



1. Виконано операції додавання, віднімання, множення та ділення для типу int.

**public** **static** **void** main(String[] args) {

**int** a = 12;

**int** b = 43;

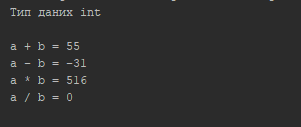
System.***out***.println("a+b=" + (a + b));

System.***out***.println("a-b=" + (a - b));

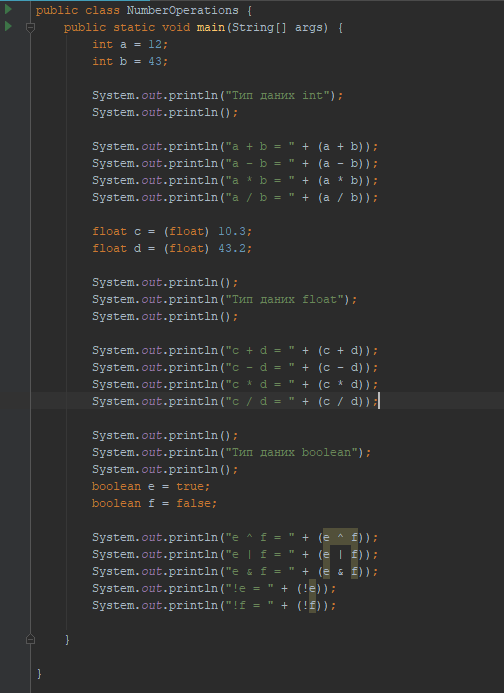
System.***out***.println("a\*b=" + (a \* b));

System.***out***.println("a/b=" + (a / b));

}



1. Змінено текст програми, добавивши виконання та вивід арифметичних операції для всіх числових типів. Змінено початкові значення змінних. Збережено текст програми та вказано її разом із результатами виконання у звіті.



1. Створено новий клас із назвою BooleanOperations та добавлено точку входу. В даному класі використано логічні оператори для роботи із типом boolean.
2. Змінна типу boolean може приймати тільки true (істина) та false (не істина). Значення можна присвоювати при ініціалізації, в процесі виконання значенням або результатом логічного виразу, наприклад:

**public** **static** **void** main(String[] args) {

**boolean** a = **true**;

**boolean** b = **false**;

System.***out***.println("a=" + a + ", b = " + b); // виведемо початкове

a = b; // значення змінних

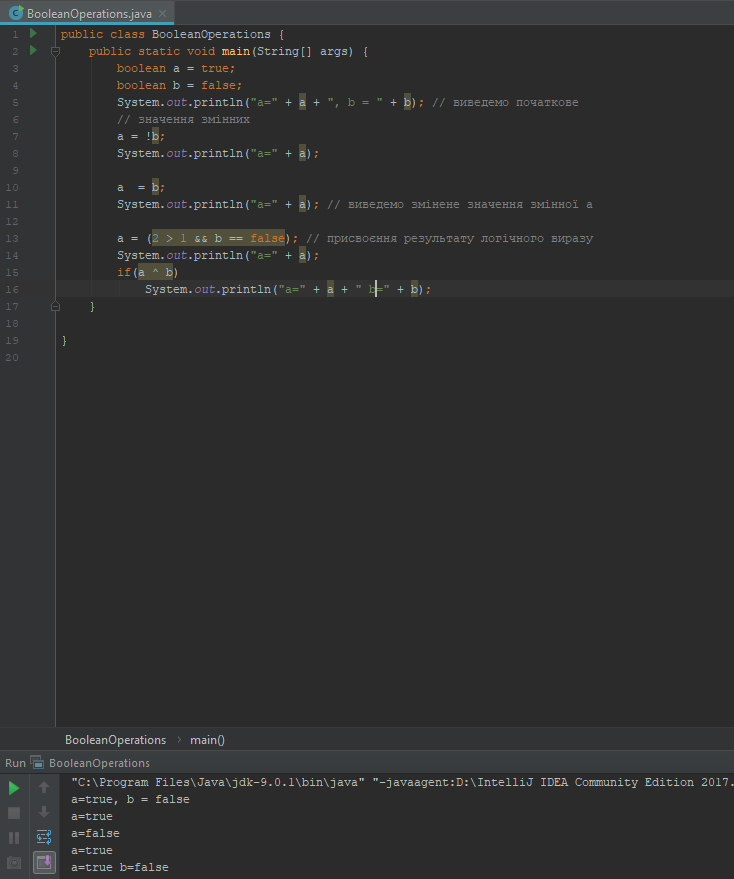
System.***out***.println("a=" + a); // виведемо змінене значення змінної а

a = (2 > 1 && b == **false**); // присвоєння результату логічного виразу

System.***out***.println("a=" + a);

}

1. Виконано дану програму та вказано у звіті результати виконання. Добавлено в програму додаткові логічні вирази, які би повертали true та false. Вказано текст програми та результати її виконання в звіті.



1. Виконано програму та дано відповіді на наступні запитання:
2. Яке значення приймає змінна а в рядку 7?

Змінна а в 7 рядку приймає значення true.

1. Що можна поміняти, щоб змінна а в рядку 7 приймала протилежне значення.

Замість 2 >1 написати 2< 1, або b ==true.

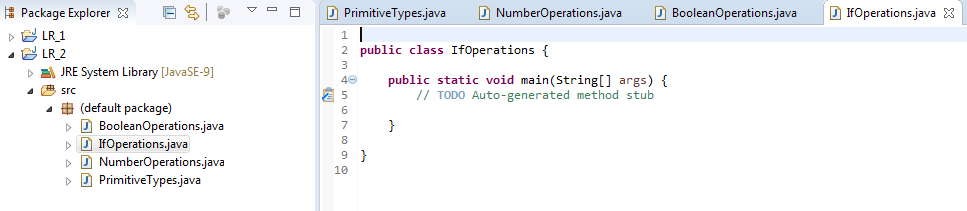
1. Який буде результат виконання, якщо початковим значення змінної a буде false, а змінної b – true?

a= false, b = true

a= true

a= false

1. Створено новий клас з назвою IfOperations. Добавлено метод main().



1. Наступна програма виконує порівняння двох чисел і виводить більше число на екран:

**public** **static** **void** main(String[] args) {

**int** a = 10;

**int** b = 12;

**if** (a > b) {

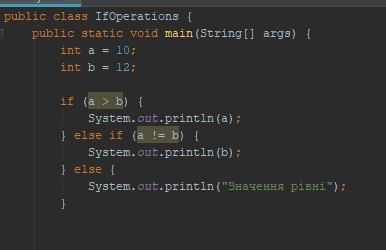
System.***out***.println(a);

} **else** {

System.***out***.println(b);

}

}



1. Змінено програму таким чином, щоб виводився текст «Значення рівні», якщо значення змінних a та b є рівними, інакше виводиться максимальне значення.

**int** a = 10;

**int** b = 12;

**if** (a > b) {

System.***out***.println(a);

}

**else** **if** (a != b) {

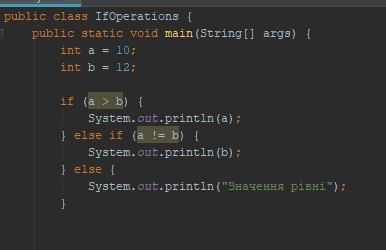
System.***out***.println(b);

}

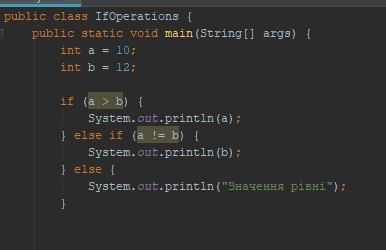
**else** {

System.***out***.println("Значення рівні");

}



1. Переміщено блоки в операторі if-else таким чином, щоб спочатку перевірялося чи значення рівні, а потім знаходилося максимальне значення.



1. Наступна програма виводить максимальне число, використовуючи скорочений варіант оператора if-then-else:

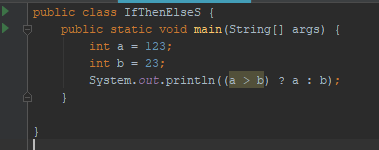
**public** **static** **void** main(String[] args) {

**int** a = 123;

**int** b = 23;

System.***out***.println((a > b) ? a : b);

}



1. Виконано наступні завдання. Занотовано програмний код та результати кожної програми в звіті:
2. Дано змінні а = 10, b = 12, c = 34. Використовуючи оператор if-else, виведіть на екран найбільше число, найменше число, суму змінних.
3. Дано змінні a = 12, b = 42, c. Обрахуйте значення змінної с як суми Вашого номера порядкового номера по журналі та номера місяця. Знайдіть суму змінних a та b, a та c, b та с. Використовуючи оператор if-else, виведіть на екран найбільшу суму, середнє арифметичне найбільшої та найменшої суми.
4. Програма приймає 2 аргументи. Виведіть на екран аргумент, що майте більшу довжину. Виведіть на екран інформацію про те чи аргументи є рівними між собою чи ні.

public class IfElsea {

public static void main(String[]args)

{

a();

b();

c();

}

public static int sum (int a, int b, int c){

return a+b+c;

}

static void a()

{

System.out.println("------------------------------------------------------------------------------");

System.out.println("метод a");

int a = 10;

int b = 12;

int c = 34;

int sum = sum(a, b, c);

System.out.println("найбільше число - " +(((a > b))&&(a > c) ? a : (b > a)&& (b > c)? b : c ));

System.out.println("найменше число - " +(((a < b))&&(a < c) ? a : (b < a)&& (b < c)? b : c ));

System.out.println("сума - " + sum);

}

static void b()

{

System.out.println("------------------------------------------------------------------------------");

System.out.println("метод b");

System.out.println("метод b");

int a =12;

int b =42;

int c = 29;

int ab = a + b;

int ac = a + c;

int bc = b + c;

int max = (((ab > ac))&&(ab > bc) ? ab : (ac > ab)&& (ac > bc)? ac : bc );

int min = (((ab < ac))&&(ab < bc) ? ab : (ac < ab)&& (ac < bc)? ac : bc );

System.out.println("сума a + b = " + ab);

System.out.println("сума a + c = " + ac);

System.out.println("сума b + c = " + bc);

System.out.println("найбільша сума = " +max);

System.out.println("найменша сума = " +min);

System.out.println("середнє арифметичне найбільшої і найменшої суми = " +(max + min)/2 );

}

static void c()

{

System.out.println("------------------------------------------------------------------------------");

System.out.println("метод c");

String argument1 = "Я довго думав над цим";

String argument2 = "Я довго думав над цим";//і через 20 хв до мене дойшло

int arg1 = argument1.length();

int arg2 = argument2.length();

String ar1 = "Аргумент 1";

String ar2 = "Аргумент 2";

System.out.println("аргумент з більшою довжиною - " + ((arg1 > arg2) ? ar1 : (arg1 < arg2) ? ar2 : "такого немає вони рівні"));

}

}

1. Дано змінні a=21, b=12. Програма приймає один аргумент. Програма повинна:

* Якщо введений аргумент рівний “max”, то вивести максимальне значення.
* Якщо введений аргумент рівний “min”, то вивести мінімальне значення.
* Якщо аргумент не рівний “max” і не рівний “min”, а різниця чисел a та b більша 0, то вивести про це повідомлення.

**public** **class** g\_16 {

**public** **static** **void** main(String args[]) {

**int** a=21, b=12,max,min;

**if** (a>b) {

max=a;

min=b;

}

**else** {

max=b;

min=a;

}

**if** (args.length==max) {

System.***out***.println("max=" +max);

}

**else** **if** (args.length==min) {

System.***out***.println("min=" +min);

}

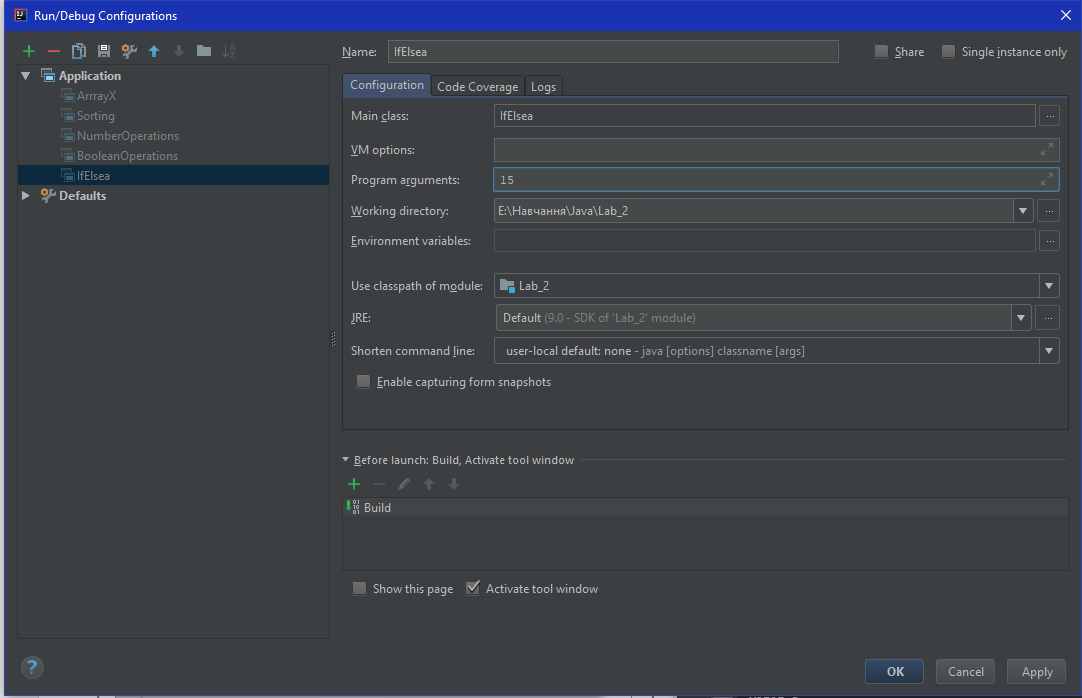
**else** **if**(a-b>0) {

System.***out***.println("Значення аргементу не рівний ні max-значенню, ні min-значенню");

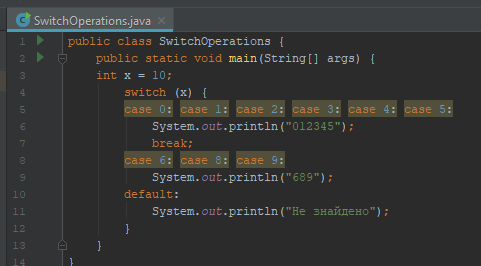
}

}

}



1. Створено новий клас з назвою SwitchOperations. Добавлено метод main().



1. Написано програму, яка використовує оператор switch та виводить відповідний текст відповідно до умови:
2. Якщо 0 <= x <=5 – вивести текст «012345».
3. Якщо x = 6 або 8 або 9 – вивести текст «689».
4. Інакше вивести текст «Не знайдено».

**int** x = 10;

**switch** (x) {

**case** 0: **case** 1: **case** 2: **case** 3: **case** 4: **case** 5:

System.***out***.println("012345");

**break**;

**case** 6: **case** 8: **case** 9:

System.***out***.println("689");

**break**;

**default**:

System.***out***.println("Не знайдено");

}



1. Виконано програму декілька разів змінюючи значення параметра x від 0 до 10. Занотовано в звіт результати виконання для різних значень x.

x=9;



x=8;



x=7;



x=6;



x=5;



x=4;



x=3;



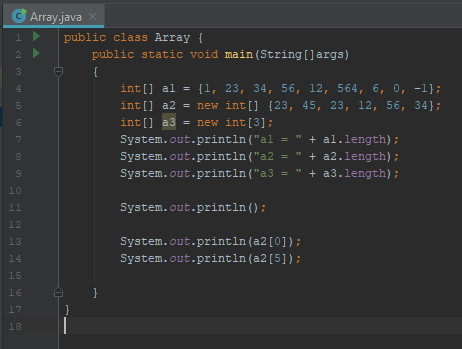
x=2;



x=1.



Створено новий клас з назвою Array. Добавлено метод main(). Клас буде використовуватися для роботи із масивами.



1. Оголошено 3 різних масиви цілих чисел, та заповнено їх довільними значеннями. Одновимірний масив це по суті список однотипних змінних. Щоб оголосити масив потрібно спочатку оголосити змінну масиву потрібного типу. Загальний формат оголошення масиву наступний: type var\_name[]:

**int**[] a1 = {1, 23, 34, 56, 12, 564, 6, 0, -1};

**int**[] a2 = **new** **int**[] {23, 45, 23, 12, 56, 34};

**int**[] a3 = **new** **int**[3];

a3[0] = 11;

a3[1] = 111;

a3[2] = 1111;

1. Довжину масиву можна визначити за допомогою поля length, наприклад a1.length. Змінено програму таким чином, щоб виводилася довжина кожного масиву.

**public** **class** ArrayOperations {

**public** **static** **void** main(String[] args) {

**int**[] a1 = {1, 23, 34, 56, 12, 564, 6, 0, -1};

**int**[] a2 = **new** **int**[] {23, 45, 23, 12, 56, 34};

**int**[] a3 = **new** **int**[3];

a3[0] = 11;

a3[1] = 111;

a3[2] = 1111;

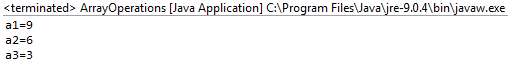
System.***out***.println("a1="+a1.length);

System.***out***.println("a2="+a2.length);

System.***out***.println("a3="+a3.length);

}

}



1. Доступ до елементів масиву починається з 0. Індекс останнього елемента менший довжини масиву на одиницю.

**int**[] a2 = **new** **int**[] {23, 45, 23, 12, 56, 34};

System.***out***.println(a2[0]); // перший елемент масиву - виводить 23

System.***out***.println(a2[5]); // виводить елемент масиву – виводить 34



1. Створено новий клас з довільною назвою та добавлено точку входу в програму – метод main().
2. Наступним завданням є навчитись працювати з одновимірними масивами. У щойно створеному класі наберано наступний програмний код:

**public** **static** **void** main(String[] args) {

**int** month\_day[];

month\_day = **new** **int**[12];

month\_day[0] = 31;

month\_day[1] = 28;

month\_day[2] = 31;

month\_day[3] = 30;

month\_day[4] = 31;

month\_day[5] = 30;

month\_day[6] = 31;

month\_day[7] = 31;

month\_day[8] = 30;

month\_day[9] = 31;

month\_day[10] = 30;

month\_day[11] = 31;

System.*out*.println("У місяці лютому " + month\_day[1] + " днів");

}



1. Модифіковано програму таким чином, щоб ініціалізація масиву відбувалась безпосередньо підчас його оголошення. Написаний програмний код та прінтскріни результатів виконання збережено у звіт.

**public** **class** OneMasuv {

**public** **static** **void** main(String[] args) {

**int**[] month\_day = {31,28,31,30,31,30,31,31,30,31,30,31};

System.***out***.println("У місяці лютому " + month\_day[1] + " днів");

}

}



1. Створено новий клас з довільною назвою та добавлено точку входу в програму – метод main().
2. Наберано наступний програмний код:

**public** **static** **void** main(String[] args) {

**int** z;

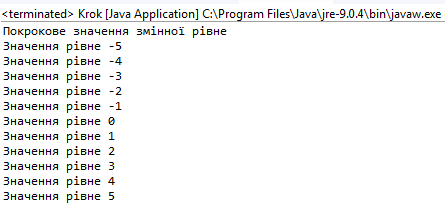
System.*out*.println("Покрокове значення змінної рівне");

**for** (z = -5; z <= 5; z++) {

System.*out*.println("Значення рівне " + z);

}

}



1. Модифіковано програму таким чином, щоб крім значення змінної виводились значення кроку.

**public** **class** ArrayAvto{

**public** **static** **void** main(String[] args) {

**int** z,a=1;

System.***out***.println("Покрокове значення змінної рівне");

**for** (z = -5; z <= 5; z++) {

System.***out***.println("Крок" +a);

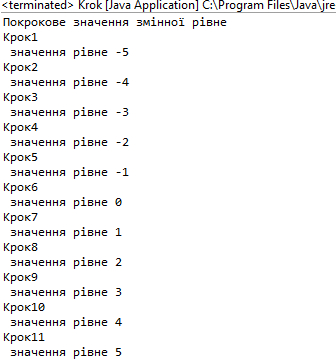
a++;

System.***out***.println(" значення рівне " + z);

}

}

}



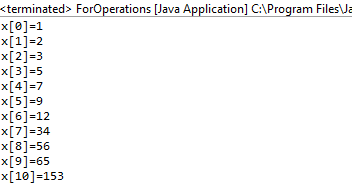
1. Створено новий клас з назвою ForOperations. Добавлено метод main().
2. Написано наступну програму, яка виводить на екран всі елементи масиву x:

**int**[] x= {1, 2, 3, 5, 7, 9, 12, 34, 56, 65, 153};

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

System.***out***.println("x[" + i + "]=" + x[i]);

}



1. Виконано наступні завдання. Занотовано програмний код та результати кожної програми в звіті.
2. Виведіть кожен другий елемент масиву.
3. Виведіть тільки парні елементи масиву.
4. Введіть число як аргумент програми.

Використовуючи Integer.parseInt(), переведіть його до цілого числа. Виведіть окремо елементи та їх кількість, що рівні, менші та більші введеного числа. Виведіть також введене число.

public class ForOperations {

public static void main(String[] args)

{

int[] x = {1, 2, 3, 5, 7, 9, 12, 34, 56, 65, 153};

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

System.out.println("x[" + i + "]=" + x[i]);

}

System.out.println();

System.out.println("Кожний 2 елемент");

System.out.println();

for (int i = 0; i < x.length; i += 2) {

System.out.println("x[" + i + "]=" + x[i]);

}

System.out.println();

System.out.println("Парні елементи");

System.out.println();

for (int i = 0; i < x.length; i++)

{

if(x[i]%2 == 0) {

System.out.println("x[" + i + "]=" + x[i]);

}

}

String abc = "12";

int cba = Integer.parseInt(abc);

int count = 0;

System.out.println();

System.out.println("Введене число - "+ cba);

System.out.println();

System.out.println("Рівні");

System.out.println();

for (int i = 0; i < x.length; i++)

{

if(x[i] == cba) {

count++;

System.out.println("x[" + i + "]=" + x[i]);

}

}

System.out.println("Кількість = "+ count);

System.out.println();

System.out.println("Менші");

System.out.println();

for (int i = 0; i < x.length; i++)

{

if(x[i] < cba) {

count++;

System.out.println("x[" + i + "]=" + x[i]);

}

}

System.out.println("Кількість = "+ count);

System.out.println();

System.out.println("Більші");

System.out.println();

for (int i = 0; i < x.length; i++)

{

if(x[i] > cba) {

count++;

System.out.println("x[" + i + "]=" + x[i]);

}

}

System.out.println("Кількість = "+ count);

}

}

k++;

System.***out***.println("x[" + i + "]=" + x[i]);

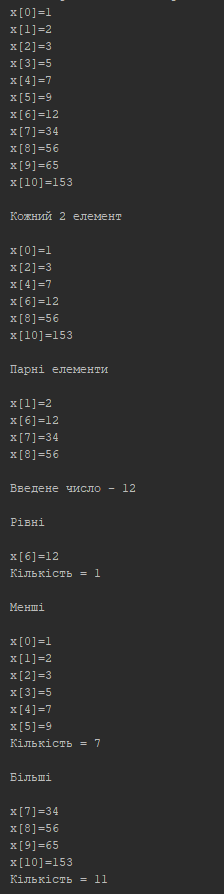
}

}

System.***out***.println("k-st=" + k);

}

}



1. Java допускає групування двох і більше операторів у блоки коду, які називаються кодовими блоками (code blocks). Наступний програмний код демонструє використання блоку коду у тілі циклу:

**int** a,b;

b=10;

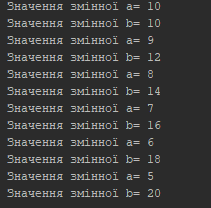
**for** (a = 10; a >= 5; a--) {

System.*out*.println("Значення змінної а= " + a);

System.*out*.println("Значення змінної b= " + b);

b=b+2;

}



1. Розглянено інший приклад, в якому знаходимо суму елементів одновимірного масиву.

**static** **int** *a*[] = **new** **int**[10];

**static** **void** print\_() {

System.*out*.println("Масив \n");

**for** (**int** i = 0; i < 10; i++) {

System.*out*.println("a[" + i + "]" + *a*[i]);

}

}

**static** **void** sum() {

**int** sum = 0;

**for** (**int** i = 0; i < 10; i++)

sum += *a*[i];

System.*out*.println("");

System.*out*.println("Сума елементів одновимірного масиву " + sum);

}

**public** **static** **void** main(String args[]) {

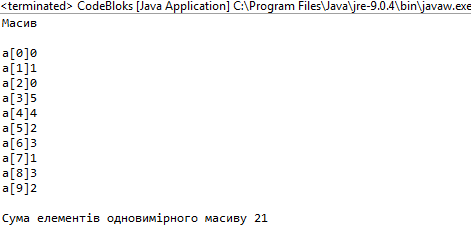
**for** (**int** i = 0; i < 10; i++)

*a*[i] = (**int**) (Math.*random*() \* 6);

*print\_*();

*sum*();

}



1. Самостійно написано програму, яка б рахувала середнє арифметичне групи з дисципліни: «Основи крос-платформенного програмування» за минулий семестр.

**public class AvgClass {**

**static int array[] = new int[28];**

**static void print\_v() {**

**System.out.println("Масив \n");**

**for (int i = 0; i < 28; i++) {**

**System.out.println("array[" + i + "] = " + array[i]);**

**}**

**}**

**static void avg() {**

**int avgg;**

**int sum = 0;**

**int count = 0;**

**for (int i = 0; i < 28; i++)**

**{**

**sum += array[i];**

**count ++;**

**}**

**avgg = sum/count;**

**System.out.println("");**

**System.out.println("Середннє арифметичне основи крос-платформенного програмування» за минулий семестр = " + avgg);**

**}**

**public static void main(String args[]) {**

**for (int i = 0; i < 28; i++)**

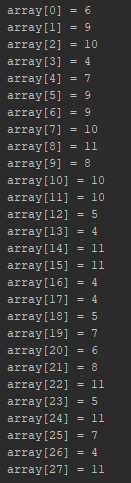
**array[i] = (int) (Math.random()\* 8 + 4);**

**print\_v();**

**avg();**

**}**

**}**



1. Модифіковано програму таким чином, щоб середнє арифметичне рахувалось з 5 обраних Вами предметів.
2. Створено новий клас з назвою WhileOperations. Добавлено метод main(). Написано наступну програму:

**int** a = 10;

**int** k = 5;

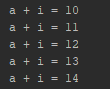
**int** i = 0;

**while** (i < k) {

System.out.println(“a + i =” + (a + i));

i++;

}



1. Переписано програму написану в пункті 29 таким чином, щоб вона використовувала цикл while.

**public** **class** Krok2 {

**public** **static** **void** main(String[] args) {

**int** z=-5,a=1;

System.***out***.println("Покрокове значення змінної рівне");

**while** (z <= 5) {

System.***out***.println("Крок-" +a+" значення рівне " + z);

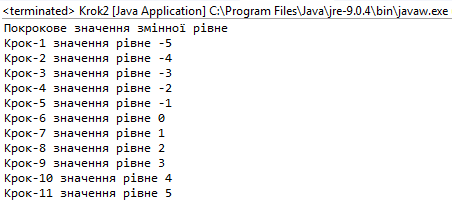
a++;

z++;

}

}

}



1. Створено новий клас з назвою ForEachOperations. Добавлено метод main().
2. Foreach використовується для того, щоб проходити по кожному елементу масиву чи колекції. Написано наступну програму, що виводить кожен елемент масиву рядків та кожен елемент масиву цілих чисел, що менші 10 та є парними:

String[] strings = {"College", "of", "Electronic", "Equipments", "Computer", "Science"};

**for** (String each : strings) {

System.***out***.println(each);

}

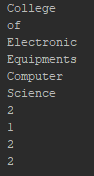
**int**[] numbers = {23, 45, 12, 2, 1,2, 45, 2, 3,6, 34, 7};

**for** (**int** each : numbers) {

**if** (each >= 10 || each % 2 != 0) **continue**;

System.***out***.println(each);

}



1. Змінено програму таким чином, щоб відбувався вихід з циклу по елементам масиву numbers, коли досягнуто елемент рівний 3.

**int**[] numbers = {23, 45, 12, 2, 1,2, 45, 2, 3,6, 34, 7};

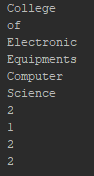
**for** (**int** each : numbers) {

**if** (each == 3) **break**;

**if** (each >= 10) **continue**;

System.***out***.println(each);

}



1. Виконано наступні завдання:
2. Вивести масив цілих чисел через кому в один рядок.

**int**[] numbers = {23, 45, 12, 2, 1, 2, 45, 2, 3, 6, 34, 7};

**for** (**int** each : numbers) {

System.***out***.print(each+", ");

}



1. Вивести масив цілих чисел у зворотньому порядку.

**int**[] numbers = {23, 45, 12, 2, 1, 2, 45, 2, 3, 6, 34, 7};

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

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

}



1. Вивести тільки першу та останню літери кожного елемента з масиву рядків.
2. Створено новий клас з довільною назвою та добавлено точку входу в програму – метод main().
3. У Java двовимірні масиви це фактично масиви масивів. Наступний приклад програми виводить на екран матрицю елементів, починаючи з 0 і закінчуючи 8. Наведена матриця складається з трьох стовпчиків і трьох рядків.

**public** **static** **void** main(String[] args) {

**int** masuv[][] = **new** **int**[3][3];

**int** i, j, z = 0;

**for** (i = 0; i < 3; i++)

**for** (j = 0; j < 3; j++) {

masuv[i][j] = z;

z++;

}

**for** (i = 0; i < 3; i++) {

**for** (j = 0; j < 3; j++)

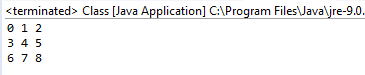
System.*out*.print(masuv[i][j] + " ");

System.*out*.println();

}

}

}



1. Метою наступного завдання є навчитись використовувати форматний вивід для змінних типу float та double. Створено новий клас з довільною назвою та добавлено точку входу в програму – метод main().У щойно створеному класі наберано наступний програмний код:

**public** **static** **void** main(String[] args) {

**float** f = 1 / 1000000f; // вказуємо f у дільнику,

**double** d = 2 / 1000000f; // щоб результат не перетворювався у ціле

System.*out*.println(f); // вивід без форматування

System.*out*.println(d);

System.*out*.println("==========");

System.*out*.format("%10.9f", f); // 10 знаків всього, 9 після коми

System.*out*.print(" ");

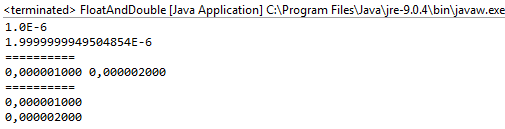
System.*out*.format("%10.9f", d);

System.*out*.println("\n==========");

System.*out*.format("%10.9f%n", f); // %n дозволяє додати перевід рядка

System.*out*.format("%10.9f%n", d); // після значення

}



1. Самостійно написано програму, яка б рахувала середнє арифметичне матриці 5х5, елементи якої обчислюються за формулою: (номер по списку \* поточне значення рядка + номер групи\* поточне значення стовпця) /2. Виконано форматне виведення.
2. Підраховано кількість елементів більших 10 у третьому стовпчику.
3. Виведено на екран елементи, які знаходяться над головної діагоналлю.
4. Виведено елементи матриці, які знаходяться під головною діагоналлю.

public class Avg5x5 {

static int array[][] = new int[5][5];

static void print\_v() {

System.out.println("Масив \n");

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (array[i][j] >= 100)

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

else System.out.print(array[i][j] + " \t");

}

System.out.println();

}

}

static void count() {

int count = 0;

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (j == 2 && array[i][j] > 10) count++;

}

}

System.out.println();

System.out.println("Кількість елементів більше 10 в 3 стовпчику = "+ count);

System.out.println();

}

static void amo() {

System.out.println();

System.out.println("Над головною");

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (i < j){

System.out.print(array[i][j]);

}

System.out.print("\t");

}

System.out.println();

}

}

static void um() {

System.out.println();

System.out.println("Під головною");

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (i > j) {

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

}

}

System.out.println();

}

}

static void avg() {

int avgg;

int sum = 0;

int count = 0;

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

sum += array[i][j];

count++;

}

}

avgg = sum / count;

System.out.println("");

System.out.println("Середннє арифметичне масиву = " + avgg);

}

public static void main(String []args) {

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

array[i][j] = (26 \* i + 1 \* j)/2;

}

}

print\_v();

count();

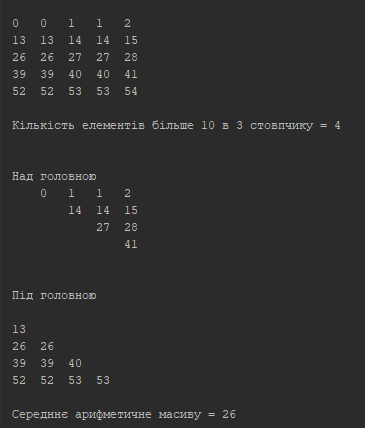
amo();

um();

avg();

}

}



1. \*Дано одновимірний масив Х, який складається з 20 елементів(додатних, від’ємних та нульових). Утворити два нові масиви, елементами, яких будуть елементи масиву Х, які більші або менші, рівні нуля.

public class ArrrayX {

static int X[] = new int[20];

static int O[] = new int[20];

static int K[] = new int[20];

static void print\_v() {

System.out.println("Масив \n");

for (int i = 0; i < 20; i++) {

System.out.println("X[" + i + "] = " + X[i]);

}

}

static void nulll()

{

int j = 0,count = 0;

for (int i = 0; i < 20; i++) {

if (X[i] == 0)

{

O[j] = X[i];

count++;

j++;

}

}

System.out.println();

for ( j = 0; j < count;j++ )

{

System.out.println("O[" + j + "] = " + O[j]);

}

}

static void maxAndMin()

{

int b = 0,countt = 0;

for (int i = 0; i < 20; i++) {

if (X[i] != 0)

{

K[b] = X[i];

countt++;

b++;

}

}

System.out.println();

System.out.println(countt);

for ( b = 0; b < countt; b++ )

{

System.out.println("K[" + b + "] = " + K[b]);

}

}

public static void main(String args[]) {

for (int i = 0; i < 20; i++)

X[i] = (int) (Math.random() \* 20 - 10);

print\_v();

nulll();

maxAndMin();

}

}

\*\*\* Індивідуальні Завдання:

1. Обчисліть f= 10! трьома різними способами (з використанням операторів циклу while/for/do-while)

public class ArrrayX {

static int X[] = new int[20];

static int O[] = new int[20];

static int K[] = new int[20];

static void print\_v() {

System.out.println("Масив \n");

for (int i = 0; i < 20; i++) {

System.out.println("X[" + i + "] = " + X[i]);

}

}

static void nulll()

{

int j = 0,count = 0;

for (int i = 0; i < 20; i++) {

if (X[i] == 0)

{

O[j] = X[i];

count++;

j++;

}

}

System.out.println();

for ( j = 0; j < count;j++ )

{

System.out.println("O[" + j + "] = " + O[j]);

}

}

static void maxAndMin()

{

int b = 0,countt = 0;

for (int i = 0; i < 20; i++) {

if (X[i] != 0)

{

K[b] = X[i];

countt++;

b++;

}

}

System.out.println();

System.out.println(countt);

for ( b = 0; b < countt; b++ )

{

System.out.println("K[" + b + "] = " + K[b]);

}

}

public static void main(String args[]) {

for (int i = 0; i < 20; i++)

X[i] = (int) (Math.random() \* 20 - 10);

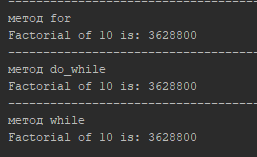
print\_v();

nulll();

maxAndMin();

}

}



2. Напишіть програму, що сортує масив цілих чисел за збільшенням та зменшенням.

public class Sorting {

static int mas[] = new int[10];

static void print\_v() {

System.out.println("Масив \n");

for (int i = 0; i < 10; i++) {

System.out.println("mas[" + i + "] = " + mas[i]);

}

}

static void sortingOnGrowth() {

System.out.println();

System.out.println("Сортування за спаданням");

int n = 10;

int tmp;

int test = 1;

while(test > 0) {

test = 0;

for (int i = 0; i < n - 1; i++) {

if (mas[i] < mas[i + 1]) {

tmp = mas[i];

mas[i] = mas[i + 1];

mas[i + 1] = tmp;

test++;

}

}

}

for(int i = 0; i < n; i++)

System.out.println("mas[" + i + "] = " + mas[i]);

System.out.println();

}

static void sortingByRecurrance\_() {

System.out.println();

System.out.println("Сортування за зростанням");

int n = 10;

int tmp;

int test = 1;

while(test > 0) {

test = 0;

for (int i = 0; i < n - 1; i++) {

if (mas[i] > mas[i + 1]) {

tmp = mas[i];

mas[i] = mas[i + 1];

mas[i + 1] = tmp;

test++;

}

}

}

for(int i = 0; i < n; i++)

System.out.println("mas[" + i + "] = " + mas[i]);

}

public static void main(String[]args)

{

for (int i = 0; i < 10; i++)

mas[i] = (int) (Math.random() \* 100);

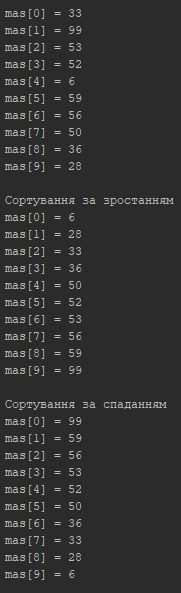
print\_v();

sortingByRecurrance\_();

sortingOnGrowth();

}

}



3. Знайдіть найменший елемент в двовимірному масиві та номер рядка і стовпця, в якому він розташовані.

public class SumaArray {

static int array [] = new int [10];

public static void main(String[]args)

{

int Suma = 0;

for (int i = 0;i < 10;i++)

{

array[i] = (int) (Math.random()\* 20);

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

}

for (int i = 0;i < 10;i++)

{

if (array[i]%2 != 0)

Suma += array[i];

}

System.out.println();

System.out.println("Suma = "+ Suma);

}

}

4. Знайдіть суму всіх непарних чисел масиву цілих чисел.

public class SearchArray {

static int array[][] = new int[5][5];

static void print\_v() {

System.out.println("Масив \n");

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (array[i][j] >= 100)

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

else System.out.print(array[i][j] + " \t");

}

System.out.println();

}

}

static void serachElements() {

System.out.println();

int i = 0,j = 0;

int min = array[i][j] ;

int row = i;

int column = j;

for ( i = 0; i < 5; i++) {

for (j = 0; j < 5; j++)

{

if (array[i][j] < min) {

min = array[i][j];

row = i;

column = j;

}

}

}

System.out.println("Рядок - "+ row);

System.out.println("Стовпчик - "+ column);

System.out.println("Мінімальне - "+ min);

}

public static void main(String[] args) {

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

array[i][j] = (int) (Math.random() \* 50);

}

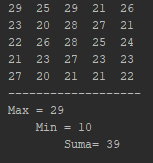
}

print\_v();

serachElements();

}

}



5. Знайдіть суму найбільшого та найменшого елементів двовимірного масиву.

public class SearchSumMaxMin {

static int array[][] = new int[5][5];

public static void main(String[] args) {

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

array[i][j] = (int) (Math.random() \* 10 + 20);

if (array[i][j] >= 100)

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

else System.out.print(array[i][j] + " \t");

}

System.out.println();

}

serachElements();

}

static void serachElements() {

int Suma;

int i = 0, j = 0;

int max = array[i][j];

int min = 10;

for ( i = 0; i < 5; i++) {

for (j = 0; j < 5; j++)

{

if (array[i][j] < min)

min = array[i][j];

if (array[i][j] > max)

max = array[i][j];

}

}

Suma = max + min;

System.out.println("-------------------");

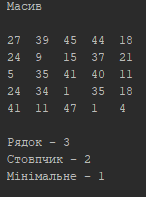
System.out.println("Max = " + max);

System.out.println(" Min = " + min);

System.out.println(" Suma= " + Suma);

}

}



6. \*Знайдіть індекс елемента, що має найменше відхилення від найбільшого елемента двовимірного масиву.

public class DeviationArgs {

static int array[][] = new int [5][5];

static void deviation() {

int max = 0;

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (array[i][j] > max) {

max = array[i][j];

}

}

}

int vid = max ;

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

if (max - array[i][j] < vid && max - array[i][j] != 0)

vid = max - array[i][j];

}

}

System.out.println();

System.out.println("Максимальне - " + max);

System.out.println("Відхилення - " + vid);

}

public static void main(String[]args)

{

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

array[i][j] = (int) (Math.random() \* 50);

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

}

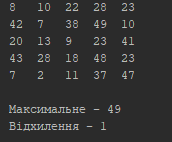
System.out.println();

}

deviation();

}

}



7. \*Знайдіть сумарне відхилення по рядках кожного елементу від найбільшого його елементу.

public class T8\_SumDeviations {

static int array[][] = new int [5][5];

static int max[] = new int [5];

static int dev[] = new int [5];

static void deviation() {

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

max[i] = array[i][0];

}

int z = 0;

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

for (int j = 0; j < 5; j++) {

if (array[i][j] > max[z]) {

max[z] = array[i][j];

}

}

z++;

}

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

System.out.println("max[" + i + "] = " + max[i]);

}

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

dev[i] = max[i];

}

z = 0;

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

for (int j = 0; j < 5; j++) {

if (max[z] - array[i][j] < dev[z] && max[z] - array[i][j] != 0) {

dev[z] = max[z] - array[i][j];

}

}

z++;

}

System.out.println();

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

System.out.println("Відхилення в " + (i + 1) + " - му рядку = " + dev[i]);

}

int sum = 0;

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

sum += dev[i];

}

System.out.println();

System.out.println("Сумарне відхилення = " + sum);

}

public static void main(String[] args)

{

for (int i = 0; i < 5; i++) {

for (int j = 0; j < 5; j++) {

array[i][j] = (int) (Math.random() \* 50);

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

}

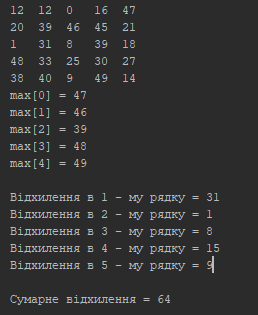
System.out.println();

}

deviation();

}

}



8. \*Елементи одновимірного масиву циклічно змістити на дві позиції вліво та на п’ять позицій вправо

public class For\_Then {

public static void main (String [] args){

int a[]= {0,1,2,3};

moveLeft(a,2);

for (int i = 0;i < a.length; i++)

System.out.println("a = "+ a[i]);

System.out.println();

moveRight(a,5);

for (int i = 0;i < a.length; i++)

System.out.println("a = "+ a[i])

}

public static void moveRight(int[] array, int positions) {

int size = array.length;

for (int i = 0; i < positions; i++) {

int temp = array[size - 1];

for (int j = size - 1; j > 0; j--) {

array[j] = array[j-1];

}

array[0] = temp;

}

}

public static void moveLeft(int[] array, int positions) {

int size = array.length;

for (int i = size; i > positions; i--) {

int temp = array[size-1];

for (int j = size-1; j > 0; j--) {

array[j] = array[j-1];

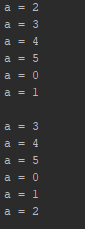
}

array[0] = temp;

}

}

}



**Контрольні запитання:**

1. Описати формат і дію умовних операторів у мові Java.
2. Описати формат і дію операторів циклу у мові Java.
3. Описати способи форматного виведення змінних типу float та double у мові Java.
4. Описати способи оголошення масивів у мові Java.