

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з навчальної дисципліни
“ *Алгоритмізація та програмування* ”
для студентів I року навчання денної та заочної форми навчання
спеціальності 122 «Комп’ютерні науки»

ЗАТВЕРДЖЕНО

на засіданні групи забезпечення спеціальності
протокол № 13 від «30» березня 2023 р.

Голова групи  Кузніченко С.Д.

ЗАТВЕРДЖЕНО

на засіданні кафедри інформаційних технологій
протокол № 6 від «16» березня 2023 р.

Завідувач кафедри  Казакова Н.Ф.

Одеса 2023

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ ДЕРЖАВНИЙ ЕКОЛОГІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт з навчальної дисципліни
“ *Алгоритмізація та програмування* ”

для студентів I року навчання денної та заочної форми навчання
спеціальності 122 «Комп’ютерні науки»

ЗАТВЕРДЖЕНО

на засіданні групи забезпечення спеціальності
протокол № ____ від «____» _____ 2023 р.

Одеса – 2023

Методичні вказівки до виконання лабораторних робіт студентів з дисципліни “Алгоритмізація та програмування”, для студентів I року навчання денної та заочної форми навчання за спеціальністю 122 «Комп’ютерні науки», рівень вищої освіти бакалавр /Кузніченко С.Д., к.г.н., доц., Клепатська В.В., асистент – Одеса, ОДЕКУ, 2023. – 71 с.

ЗМІСТ

Передмова	5
Список джерел посилань	6
Правила техніки безпеки та охорона праці	7
Лабораторна робота № 1 «Примітивні типи даних. Арифметичні і логічні операції. Робота з пакетом java.lang. Методи класу Math»	8
Лабораторна робота № 2 «Організація розгалуження з використанням умовного оператора та оператора switch. Умовна операція»	22
Лабораторна робота № 3 «Складання програм циклічної структури. Цикли while, do і for»	30
Лабораторна робота № 4 «Робота з лінійними масивами. Двовимірні масиви» ..	36
Лабораторна робота № 5 «Оголошення і виклик методів. Передача в метод параметрів примітивних типів. Рекурсивні методи»	45
Лабораторна робота № 6 «Файлове введення/виведення. Рядки. Робота з методами класу String і StringBuffer»	54
Додаток. Етапи виконання Java-програми та особливості роботи із середовищем розробки BlueJ	65

Передмова

Методичні вказівки призначені для студентів I року навчання денної та заочної форми навчання. Мета виконання лабораторних робіт – закріплення теоретичного лекційного матеріалу та придбання практичних навичок програмування мовою Java, яка є сучасною та найбільш розвиненою у цей час.

Дисципліна «Алгоритмізація і програмування» є обов'язковою дисципліною, спеціальності 122 «Комп'ютерні науки». Внаслідок вивчення даної дисципліни студенти повинні **знати** найпростіші поняття ООП: класи, об'єкти, властивості (поля) і дії (методи), структури даних і алгоритми, основи роботи з рядками, ведення-виведення у/з файл(у), динамічні структури даних (списки, черги, стеки і бінарні дерева) та алгоритми їх оброблення. Вони повинні **вміти** застосовувати алгоритми подання, зберігання і обробки інформації, скласти лінійні та розгалужені програми мовою Java та здобути практичні навички створення і модифікації невеликих програмних проектів.

Методичні вказівки містять контрольні запитання та завдання. Всі лабораторні роботи підкріплені прикладами розв'язання типових задач на ПЕОМ.

Під час підготовки до лабораторної роботи студент повинен вивчити відповідний теоретичний матеріал за конспектом лекцій і літературою, розібрати приклади розв'язання задач, та відповісти на контрольні питання.

На початку лабораторної роботи викладач проводить співбесіду за результатами якої студент отримує, або не отримує допуск до виконання лабораторної роботи. Якщо студент не отримав допуску, він залишається на заняттях, але не виконує лабораторної роботи на комп'ютері. Замість цього він вивчає теоретичний матеріал за даною темою, щоб відповісти на питання викладача та отримати допуск до виконання роботи.

За кожну лабораторну роботу студент отримує дві оцінки: за виконання та за захист роботи. Згідно з цих пунктів студенту зараховується відповідна кількість балів. Максимальні бали з кожної лабораторної роботи встановлюються згідно силлабусу дисципліни. На першому занятті студенти отримують графік контролюючих заходів: перелік контролюючих заходів, терміни виконання, бали за кожний вид робіт.

Критерії оцінювання наведені в силлабусі навчальної дисципліни «Алгоритмізація і програмування» (http://eprints.library.odku.edu.ua/id/eprint/9593/1/Tereschenko_Algoritmizatsiya_ta_programuvannya_silabus_2021.pdf)

Список джерел посилань

Основна

1. *А.Ф.Верлань, І.О.Чмирь, С.Д. Кузніченко С.Д., Л.Б.Коваленко* Імперативне програмування та об'єктно-орієнтоване моделювання: Java, UML, OCL. Навчальний посібник для студентів вищих навчальних закладів. Одеса: Видавництво Екологія.– 2013., 326 с.
2. *Кузніченко С.Д., Л.Б.Коваленко.* Алгоритмізація та програмування. – Конспект лекцій. Одеса: ОДЕКУ – 2015 р. , 340 с.

Додаткова

1. *ЕККЕЛ' В.* Java Philosophy. Programmer's Library. 2001. 850 p.
2. *Глушаков С.В.* Програмування на Java 2: Вид.2-е. – Харків: Фоліо, 2003. – 536 с. – (Навчальний курс).
3. *А.В. Картузов, Д.В. Ніколенко.* Програмуємо на мові Java – Видавництво: Наука та техніка, 2001. – 192 с.
4. *Любош Бруга.* Java по-швидкому. Практичний експрес-курс. Видавництво: Наука і техніка, 2006. – 384 с

Правила техніки безпеки та охорона праці

Лабораторні роботи з дисципліни проводяться у лабораторіях кафедри інформаційних технологій або кафедри інформатики, які оснащені комп'ютерною технікою з відповідним програмним забезпеченням. Студенти зобов'язані дотримуватися правил техніки безпеки та правил користування обчислювальною технікою в лабораторіях кафедр.

Згідно з «Правилами техніки безпеки в лабораторіях» студентам забороняється:

- з'являтися та знаходитись приміщенні в нетверезому стані;
- ставити поруч з клавіатурою ємності з рідиною;
- перебувати в приміщенні у верхньому одязі та завалювати ним робочі столи та стільці;
- працювати в лабораторії більше 6-ти годин на день (для вагітних жінок більше 4-х годин);
- за власною ініціативою змінювати закріплені за ними робочі місця та знаходитись в приміщенні під час роботи іншої учбової групи;
- самостійно виконувати вмикання електроживлення лабораторії та заміну складових частин ПК, що вийшли із ладу.

У випадку виявлення несправностей обчислювальної техніки студент повинен сповістити про це викладача чи будь-кого з навчально-допоміжного персоналу лабораторії.

Лабораторна робота № 1

«Примітивні типи даних. Арифметичні і логічні операції. Робота з пакетом java.lang. Методи класу Math»

1. Мета роботи

Метою лабораторної роботи є ознайомлення студентів з прийомами програмування найпростіших програм мовою Java з використанням примітивних типів даних та арифметичних і логічних операцій. Отримання навичок програмування алгоритмів лінійної структури та алгоритмічних виразів з використанням методів класу Math пакета java.lang. Вивчення роботи класу Scanner, який містить методи для введення даних примітивних типів з клавіатури.

2. Завдання до лабораторної роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

Напишіть програму розрахунку заданих арифметичних виразів. Для введення даних із клавіатури необхідно використовувати клас Scanner.

2.1 Методичні вказівки

Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Виконання Java-програми. Коментар. Ідентифікатори.
- Прості типи даних: логічний, цілі, дійсні, символи та рядки.
- Оголошення змінних.
- Арифметичні і логічні операції.
- Операції порівняння.
- Правила приведення типів.
- Операції присвоювання.
- Методи класу Math пакета java.lang.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

Перед виконанням лабораторної роботи необхідно уважно ознайомитися з особливостями виконання Java – програми.

Написати програму на Java можна в будь-якому текстовому редакторі, наприклад, Notepad, WordPad в MS Windows і ін. Запустити компілятор, а потім інтерпретатор байт-кодів можна з командного рядка. Однак, зручніше для цих цілей використовувати одну із численних інтегрованих середовищ розробки програм для Java. Найпростіші з них є просто інтегрованими оболонками над

JDK, що викликають із одного вікна текстовий редактор, компілятор і інтерпретатор. Ці інтегровані середовища вимагають попередньої установки JDK. На лабораторних роботах студенти можуть використовувати середовище розробки програм BlueJ, яке спеціально створене для навчання на початковому рівні. У додатку наведений докладний опис етапів виконання Java-програми та особливості роботи із середовищем розробки BlueJ.

В кожному варіанті лабораторної роботи є два невеликих завдання, тому студент за бажанням може створити у власному проєкті два окремих класа, або передбачити виконання усіх завдань в одному класі проєкту.

При рішенні завдань можуть бути корисними наступні математичні формули з планіметрії:

- площа круга $S = \pi R^2$;
- довжина кола $l = 2\pi R$;
- об'єм конуса $V = \frac{\pi R^2 h}{3}$;
- об'єм прямого кругового циліндра $V = \pi R^2 h$;
- площа бічної поверхні циліндра $S = 2\pi R h$;
- об'єм кулі $V = \frac{4}{3} \pi R^3$
- площа поверхні кулі $S = 4\pi R^2$.

Для використання математичних методів класу Math, необхідно імпортувати в програму пакет java.lang: **import java.lang.Math;**

Для введення даних із клавіатури треба використовувати клас Scanner, для цього необхідно імпортувати в програму пакет **import java.util.*;**

Для виведенні розрахункових значень на консоль треба використовувати метод **System.out.printf()** ;

При форматованому виведенні використовуються специфікатори формату, деякі з них наведенні у таблиці¹ нижче:

Специфікатор формату	Тип елементу даних
%b	Логічне значення аргументу (boolean)
%c	Одиночний символ
%d	Ціле число типу int
%e	Число з плаваючою точкою
%f	Число з фіксованою точкою типу float
%g	Обирає більш коротке представлення з двох: %e або %f
%s	Рядок символів
%n	Вставка символу нового рядку
%%	Вставка знака %

Також можливе використання специфікаторів з великими літерами: %S (еквівалентно %s). Форматування з їх допомогою забезпечує переводсимволів в

верхній реєстр.

Приклади застосування форматного виводу:

```
System.out.printf("%b %B", true, false);
```

```
true FALSE
```

```
System.out.printf("%-3C, %3c", 'd', 'e');
```

```
D , e
```

```
System.out.printf("%-6dx",5);
```

```
5 x
```

```
System.out.printf("%1.5f", 123.456f);
```

```
123.45600
```

```
System.out.printf("%6.0f", 123.456f);
```

```
123
```

```
System.out.printf("%1.5E", 123.456f);
```

```
1.23456E+02
```

```
System.out.printf("This %s is about %n%S %c", "book","java",'6');
```

```
This book is about
```

```
JAVA 6
```

В кожному варіанті лабораторної роботи є 5 завдань. Друге завдання містить два математичних вирази. Якщо студент правильно склав алгоритм розв'язання задачі, то повинен отримати два однакових результати розрахунку. В третьому завданні необхідно згенерувати випадкові числа у заданому інтервалі з використанням методу random().

Таблиця 1 – Тригонометричних методів класу Math

Метод	Формула	Опис
<i>Прямі тригонометричні функції</i>		
sin(x)	sin(x)	Повертає синус кута x, переданого в радіанах
cos(x)	cos(x)	Повертає косинус кута x, переданого в радіанах
tan(x)	tg(x)	Повертає тангенс кута x, переданого в радіанах
<i>Зворотні тригонометричні функції</i>		
asin(x)	arcsin(x)	Повертає кут в радіанах, синус якого дорівнює x
acos(x)	arccos(x)	Повертає кут в радіанах, косинус якого дорівнює x
atan(x)	arctg(x)	Повертає кут в радіанах, тангенс якого дорівнює x
atan2(y, x)		Повертає кут в радіанах, тангенс якого дорівнює x/y
<i>Гіперболічні функції</i>		
sinh(x)	$sh(x) = \frac{e^x - e^{-x}}{2}$	Повертає гіперболічний синус кута x, переданого в радіанах
cosh(x)	$ch(x) = \frac{e^x + e^{-x}}{2}$	Повертає гіперболічний косинус кута x, переданого в радіанах
tanh(x)	$th(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	Повертає гіперболічний тангенс кута x, переданого в радіанах.
<i>Додаткові функції</i>		
toRadians(angle)	$\alpha [rad] = \alpha [] \cdot (\pi / 180)$	Перетворює градуси на радіани
toDegrees(angle)	$\alpha [] = \alpha [rad] \cdot (180 / \pi)$	Перетворює радіани на градуси
hypot(x, y)	$\sqrt{x^2 + y^2}$	Повертає довжину гіпотенузи за двома катетами x і y

Таблиця 2 – Експоненціальні методи класу Math

Метод	Формула	Опис
<code>exp(x)</code>	e^x	Повертає експоненту x .
<code>expm1(x)</code>	$e^x - 1$	Повертає експоненту $x-1$. При x близькому до 0, дає більш точні значення, ніж <code>exp(x) - 1</code>
<code>log(x)</code>	$\ln(x)$	Повертає натуральний логарифм
<code>log10(x)</code>	$\log_{10}(x)$	Повертає логарифм за основою 10
<code>log1p(x)</code>	$\ln(1+x)$	Повертає натуральний логарифм $x+1$
<code>sqrt(x)</code>	\sqrt{x}	Повертає квадратний корінь з x
<code>cbrt(x)</code>	$\sqrt[3]{x}$	Повертає кубічний корінь з x
<code>pow(x, y)</code>	x^y	Повертає x в степені y
<code>scalb(x, n)</code>	$x \cdot 2^n$	Повертає значення $x \cdot 2^n$. Параметр n має тип <code>int</code> . Параметр x може бути <code>double</code> , тоді і результат виконання методу буде <code>double</code> , або <code>float</code> , тоді і результат буде мати тип <code>float</code>

Таблиця 3 – Методи округлення класу Math

Метод	Опис
<i>Методи знаходження модуля, знака, мінімального і максимального числа</i>	
<code>abs(x)</code>	Повертає абсолютне значення числа. Аргумент типу <code>int</code> , <code>long</code> , <code>float</code> або <code>double</code> . Результат того ж типу, що аргумент.
<code>signum(x)</code>	Визначає знак числа. Повертає 0, якщо x дорівнює 0, 1 якщо x більший за 0 і -1 – якщо x менший від 0. Аргумент типу <code>float</code> або <code>double</code> . Результат того ж типу, що й аргумент.
<code>copySign(x, s)</code>	Повертає x з тим же знаком, що у s . Аргументи одного типу. Можливі типи: <code>float</code> і <code>double</code> . Результат того ж типу, що й аргументи.
<code>min(x, y)</code>	Повертає мінімальне з двох чисел. Аргументи одного типу. Можливі типи: <code>int</code> , <code>long</code> , <code>float</code> , <code>double</code> . Результат того ж типу, що аргумент.
<code>max(x, y)</code>	Повертає максимальне з двох чисел. Аргументи одного типу. Можливі типи: <code>int</code> , <code>long</code> , <code>float</code> , <code>double</code> . Результат того ж типу, що аргумент.
<i>Методи округлення</i>	
<code>ceil(x)</code>	Повертає найменше ціле число, яке більше за x
<code>floor(x)</code>	Повертає найбільше ціле число, яке менше або дорівнює x
<code>round(x)</code>	Повертає найближче до x ціле. Аргумент типу <code>float</code> або <code>double</code> . Результат типу <code>long</code> , якщо аргумент <code>double</code> , і типу <code>int</code> – якщо <code>float</code> . Те ж, що <code>(int) floor(x+0.5)</code>
<code>rint(x)</code>	Повертає найближче ціле до x .

Метод	Опис
<code>ulp(x)</code>	Відстань до найближчого більшого ніж аргумент значення того ж типу. Можна використовувати для отримання точності результату. Аргумент типу <code>float</code> або <code>double</code> . Результат того ж типу, що аргумент.
<code>nextUp(x)</code>	Повертає наступне значення в додатному напрямку від <code>x</code> . Аргумент типу <code>float</code> або <code>double</code> . Результат того ж типу, що аргумент.
<code>nextAfter(x, d)</code>	Починаючи зі значення <code>x</code> , повертає наступне значення в напрямку <code>d</code> . Якщо <code>x==d</code> , то повертається <code>d</code> . Аргумент <code>d</code> має тип <code>double</code> . Аргумент <code>x</code> типу <code>float</code> або <code>double</code> . Результат того ж типу, що й аргумент <code>x</code> .
<i>Додаткові методи</i>	
<code>random()</code>	Повертає псевдовипадкове число в діапазоні від 0.0 до 1.0. При цьому <code>Math.random() < 1</code>
<code>IEEEremainder(x, y)</code>	Якщо потрібно отримати псевдовипадкове число в діапазоні від <code>x</code> до <code>y</code> , то можна використовувати вираз: $(y - x) * \text{Math.random}() + x$

2.2 Приклади програм

Знайти площу поверхні та об'єм циліндра висотою `h` та радіусом `R`. Визначити, чи є цифра 5 серед цифр, що складають десятинний запис додатного тризначного цілого числа `M`.

```
class Lab_1{
    public static void main (String[] args){
        //_____I_____
        double h=5.8, R=3.2, PI=3.1457;
        double S=2*PI*R*(R+h);
        double V=PI*R*R*h;
        System.out.println("Площа циліндра = "+S+"\n"+"Об'єм циліндра = "+V);
        //_____II_____
        int M=375;
        int k1=M/100; //кількість сотень
        int k2=M/10%10; //кількість десятків
        int k3=M%10; //кількість одиниць
        boolean log=(k1==5) || (k2==5) || (k3==5);
        System.out.println("Серед цифр числа є цифра 5 - "+log);
    }
}
```

Розрахуйте $y = (-1)^n (2n + 1) / 2$, де n – довільне ціле число, що вводиться з клавіатури;

Розрахуйте

$$z_1 = \frac{2 \sin^2(3\pi - 2\alpha) \cos^2(5\pi + 2\alpha)}{1 - 1 - 5}$$

$$z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5}{2}\pi - 8\alpha\right)$$

```
import java.lang.Math;
import java.util.*;
class Rashet{
public static void main (String[] args) {
Scanner scan = new Scanner (System.in);
System.out.println("Введіть n->");
int n=scan.nextInt();
double y=Math.pow(-1,n)*(2*n+1)/2.;

System.out.println("Введіть a->");
double a=scan.nextDouble();
double z1,z2;
//double p=10e5;
z1=2*Math.pow(Math.sin(3*Math.PI-2*a)*Math.cos(5*Math.PI+2*a),2);
z2=1./4.-1./4.*Math.sin(5./2.*Math.PI-8*a);
//System.out.println("Y= "+Math rint(y*p)/p);
//System.out.println("Z1= "+Math rint(z1*p)/p);
//System.out.println("Z2= "+Math rint(z2*p)/p);
System.out.printf("Y= %6.3f",y);
System.out.printf("Z1=%6.2f",z1);
System.out.printf("Z2= %6.2f",z2);
}}
```

2.3 Контрольні питання

1. Які ідентифікатори припустимі в Java? Чим відрізняються прості й складені ідентифікатори?
2. Які види коментарів існують в Java?
3. Назвіть особливості виконання Java – програми. Яким чином можна створити вихідний, об'єктний та завантажувальний модуль?
4. Які примітивні типи даних існують в Java? Приведіть діапазони припустимих значень і розрядність для цілих і дійсних типів даних.
5. Який формат має оголошення змінної? Приведіть приклади оголошень змінних примітивних типів.
6. Поясніть дію операції зчеплення.
7. За яким правилом здійснюється приведення типів в Java?
8. Поясніть дію постфіксної і префіксної форми запису операцій інкрементування та декрементування.
9. Поясніть дію простої та складової операцій присвоювання.

10. Які особливості виконання арифметичних операцій ділення та знаходження остачі від ділення стосовно до змінних цілого типу?
11. За допомогою якого виразу можна імпортувати пакет у програмний модуль? Приведіть приклади виразів, що імпортують пакет java.lang повністю або тільки його клас Math.
12. Яким чином здійснити ввід даних за допомогою методів класу Scanner. Наведіть приклади вводу змінних цілого та дійсного типів.
13. Для чого призначений метод random() класу Math? Як з його допомогою можна одержати випадкове число із діапазону $[-15 \div 35]$?
14. Яка відмінність між методами rint() і round() класу Math? Якого типу буде результат виконання методу round(), якщо аргумент має тип double? float?
15. Опишіть призначення методів ceil() і floor(). Приведіть приклади їхнього використання.
16. Яка різниця між методами print() і println()?

2.4 Варіанти завдань

1.
 - * Цілій змінній h присвоїти третю з кінця цифру в записі додатного цілого числа k.
 - * Визначити, чи може окружність радіуса R пройти впрямокутний отвір зі сторонами a, b.
 - * Розрахуйте $y = n(n + 1)/2$, де n – довільне ціле число, що вводиться з клавіатури;
 - * Розрахуйте

$$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin \alpha$$

$$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin \left(\frac{\pi}{4} + 2\alpha\right)$$
 - * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.
2.
 - * Цілій змінній d присвоїти першу цифру дрібної частини додатного дійсного числа x.
 - * Визначити, чи є рівнобедреним трикутник зі сторонами a, b, c.
 - * Розрахуйте $y = (2n + n)(n + 1) / 2$, де n – довільне ціле число, що вводиться з клавіатури;
 - * Розрахуйте

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2\sin^2 2\alpha}$$

$$z_2 = 2\sin \alpha$$
 - * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

3. * Визначити, чи дорівнює сума двох перших цифр заданого чотиризначного числа сумі двох останніх.
- * Для k -ої секунди доби визначити, скільки повних годин (h) і повних хвилин (m) пройшло до цього моменту.
- * Розрахуйте $y = n(n + 1)(2n + 1) / 6$, де n – довільне ціле число, що вводиться з клавіатури;
- * Розрахуйте
- $$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha}$$
- $$z_2 = \operatorname{tg} 3\alpha$$
- * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.
4. * Визначити, чи є цифра 0 серед перших трьох цифр дрібної частини заданого дійсного числа.
- * Визначите кут (у градусах) між положенням годинної стрілки на початку доби і її положенням в h годин, m хвилин, s секунд ($0 \leq h \leq 11$, $0 \leq (m, s) \leq 59$).
- * Розрахуйте $y = n^2 (n + 1)^2 / 4$, де n – довільне ціле число, що вводиться з клавіатури;
- * Розрахуйте
- $$z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha$$
- $$z_2 = \cos^2 \alpha + \cos^4 \alpha$$
- * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.
5. * Визначите, чи є серед цифр заданого тризначного числа однакові.
- * Визначите повну кількість годин і хвилин, які пройшли від початку доби до того моменту (у першій половині дня), коли годинна стрілка повернулася на кут φ градусів ($0 \leq \varphi < 360$).
- * Розрахуйте $y = n(4n^2 - 1) / 3$, де n – довільне ціле число, що вводиться з клавіатури;
- * Розрахуйте
- $$z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha$$
- $$z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5\alpha}{2} \cdot \cos 4\alpha$$
- * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.
6. * Визначити суму перших трьох цифр дрібної частини додатного дійсного числа x .
- * Дано три довільних числа. Визначити, чи можна побудувати трикутник з такими довжинами сторін.
- * Розрахуйте $y = n^2 (2n^2 - 1) / 5$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \cos^2 \left(\frac{3}{8}\pi - \frac{\alpha}{4} \right) - \cos^2 \left(\frac{11}{8}\pi + \frac{\alpha}{4} \right)$$

$$z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

7. * Дано два додатних цілих тризначних числа. Визначите, чи дорівнюють добутки їхніх цифр.

* Поміняти місцями значення цілих змінних x і y , не використовуючи додаткових змінних.

* Розрахуйте $y = n(n+1)(2n+1)(3n^2-3n-1) / 30$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \cos^4 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1$$

$$z_2 = \sin(y+x) \cdot \sin(y-x)$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

8. * Визначите, чи більше перша цифра додатного цілого тризначного числа за середнє арифметичне всіх його цифр.

* Знайти, скільки цілих діб і годин пройшло за m хвилин.

* Розрахуйте $y = n^3(n-1)(2n+1) / 30$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha - \sin \beta)^2$$

$$z_2 = -4 \sin^2 \frac{\alpha - \beta}{2} \cdot \cos(\alpha + \beta)$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

9. * Надрукувати число, яке виходить при перестановці цифр сотень і одиниць заданого тризначного числа.

* Для довільних дійсних чисел a , b , c визначити, чи має квадратне рівняння хоча б один дійсний корінь.

* Розрахуйте $y = n\sqrt{2n+1} / 3$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{\sin \left(\frac{\pi}{2} + 3\alpha \right)}{1 - \sin(3\alpha - \pi)}$$

$$z_2 = \operatorname{ctg} \left(\frac{5}{4}\pi + \frac{3}{2}\alpha \right)$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

10. * Визначите, чи є перша цифра додатного цілого чотиризначного числа більше за добуток його інших цифр.

*Обчислити довжину кола, площу круга та об'єм сфери одного й того ж радіусу R.

*Розрахуйте $y = \sqrt{8 + n^3 + n^2} / 20$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{1 - 2\sin^2 \alpha}{1 + \sin 2\alpha}$$

$$z_2 = \frac{1 - \operatorname{tg} \alpha}{1 + \operatorname{tg} \alpha}$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

11. * Надрукувати число, яке виходить при перестановці перших двох і останніх двох цифр додатного цілого чотиризначного числа.

* Дано: (x1, y1) і (x2, y2) – координати лівої верхньої та правої нижньої вершин прямокутника. Визначити, чи лежить точка A(x, y) усередині або на одній зі сторін цього прямокутника.

* Розрахуйте $y = n(n^3 + n^2 - 1) / 6$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{\sin 4\alpha}{1 + \cos 4\alpha} \cdot \frac{\cos 2\alpha}{1 + \cos 2\alpha}$$

$$z_2 = \operatorname{ctg} \left(\frac{3}{2}\pi - \alpha \right)$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

12. *Визначите, чи зустрічається в записі додатного цілого чотиризначного числа хоча б одна парна цифра.

* По довжинам двох катетів обчислити площу прямокутного трикутника.

*Розрахуйте $y = (1 - 2n + n^4) / 3$, де n – довільне ціле число, що вводиться з клавіатури;

* Розрахуйте

$$z_1 = \frac{\sin \alpha + \cos(2\beta - \alpha)}{\cos \alpha - \sin(2\beta - \alpha)}$$

$$z_2 = \frac{1 + \sin 2\beta}{\cos 2\beta}$$

* Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

13. * Визначите, чи утворюють цифри даного чотиризначного числа Нстрого зростаючу послідовність.
- * Знайти об'єм конуса з радіусом R і з висотою h .
- * Розрахуйте $y = n(n^{2n+1} + 2) / 6$, де n – довільне ціле число, що вводиться з клавіатури;
- * Розрахуйте
- $$z_1 = \frac{(m-1)\sqrt{m} - (n-1)\sqrt{n}}{\sqrt{m^3n} + n \cdot m + m^2 - m}$$
- $$z_2 = \frac{\sqrt{m} - \sqrt{n}}{m}$$
- * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.
14. * Визначите, чи зустрічається в записі додатного цілого тризначного числа хоча б одна непарна цифра.
- * Знайти об'єм та площу поверхні кулі з радіусом R .
- * Розрахуйте $y = (nn - 2n + 3) / 3$, де n – довільне ціле число, що вводиться з клавіатури;
- * Розрахуйте
- $$z_1 = \frac{\sqrt{2b + 2\sqrt{b^2 - 4}}}{\sqrt{b^2 - 4} + b + 2}$$
- $$z_2 = \frac{1}{\sqrt{b + 2}}$$
- * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.
15. * Цілій змінній N привласнити суму цифр цілого тризначного числа.
- * Визначите, чи більша площа бокової поверхні циліндра за площу круга, що є основою циліндра. Радіус круга R , висота циліндра h .
- * Розрахуйте $y = (-1)^{2n+1} (2n + 1) / 2$, де n – довільне ціле число, що вводиться з клавіатури;
- * Розрахуйте
- $$z_1 = \frac{x^2 + 2x - 3 + (x + 1)\sqrt{x^2 - 9}}{x^2 - 2x - 3 + (x - 1)\sqrt{x^2 - 9}}$$
- $$z_1 = \sqrt{\frac{x + 3}{x - 3}}$$
- * Виконати **завдання 5** (сторінка 20) згідно варіанту за **таблицею 1**.

Завдання 5. Привласнити випадково (за допомогою методу `random()`) значення числовим цілим змінним **a1, a2,..., a5** у зазначеному в квадратних дужках діапазоні. Вивести результат на екран по одному значенню, використовуючи методи `print()` або `println()`.

Таблиця 1 – Варіанти завдання до лабораторної роботи №1

Варіант 1	Варіант 2	Варіант 3	Варіант 4	Варіант 5
[15÷25] a1 a2 a3 a4 a5	[100÷200] a1 a2 a3 a4 a5	[1÷25] a1 a2 a3 a4 a5	[-10÷10] a1 a2 a3 a4 a5	[-5÷25] a1 a2 a3 a4 a5
Варіант 6	Варіант 7	Варіант 8	Варіант 9	Варіант 10
[-150÷250] a1 a2 a3 a4 a5	[-35÷20] a1 a2 a3 a4 a5	[-75÷15] a1 a2 a3 a4 a5	[-5÷65] a1 a2 a3 a4 a5	[10÷100] a1 a2 a3 a4 a5
Варіант 11	Варіант 12	Варіант 13	Варіант 14	Варіант 15
[-55÷60] a1 a2 a3 a4 a5	[12÷22] a1 a2 a3 a4 a5	[7÷77] a1 a2 a3 a4 a5	[-12÷-10] a1 a2 a3 a4 a5	[-15÷-5] a1 a2 a3 a4 a5

3. Прилади, устаткування та інструменти

Для виконання лабораторної роботи використовується ПЕОМ з установленим пакетом Sun Microsystems JDK 1.5 і вище та інтегрованим середовищем розробки BlueJ. Для написання програми на Java може бути використаний будь-який текстовий редактор, наприклад, Notepad, WordPad в MSWindows і ін.

4. Правила техніки безпеки та охорони праці

Правила техніки безпеки при виконанні лабораторної роботи регламентуються «Правилами техніки безпеки при роботі в комп'ютерній лабораторії».

5. Порядок проведення лабораторної роботи

Для виконання роботи кожен студент повинен:

1. Відповісти на контрольні питання та пройти усне опитування за теоретичним матеріалом лабораторної роботи;
2. Пройти інструктаж за правилами охорони праці;
3. Запустити на комп'ютері інтегроване середовище розробки BlueJ;
4. Ознайомитись з правилами роботи в середовищі BlueJ та запустити на виконання просту програму, розглянуту на лекційному занятті;
5. Отримати варіант завдання у викладача;
6. Скласти алгоритм розв'язання задачі;
7. Записати код програми на комп'ютері;
8. Від компілювати програму та виправити всі помилки;
9. Запустити програму на виконання;
10. Отримати результати роботи програми і показати їх викладачу;
11. Підготувати і захистити звіт до лабораторної роботи.

6. Оформлення і захист звіту

Підготовлений до захисту звіт до лабораторної роботи повинен містити:

- титульний лист, де вказані номер і назва лабораторної роботи, відомості про виконавця;
- номер варіанта роботи та текст завдання;
- відповіді на контрольні запитання до лабораторної роботи;
- текст програми алгоритмічною мовою Java;
- лістинг результатів виконання програми.

Лабораторна робота № 2

«Організація розгалуження з використанням умовного оператора та оператора switch. Умовна операція»

1. Мета роботи

Отримання навичок програмування алгоритмів розгалуженої структури за допомогою умовного оператора if та оператора варіанту switch.

2. Завдання до лабораторної роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

2.1 Методичні вказівки

Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Умовна операція.
- Пріоритет операцій. Вираження.
- Організація розгалужень у програмі. Умовний оператор if.
- Оператор вибору switch.
- Мітки. Організація переривань за допомогою оператора break.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

Алгоритми розгалуженої структури (розгалуження) – це така схема, у якій передбачене розгалуження зазначеної послідовності дій на два напрямки залежно від результату перевірки заданої умови. До операторів розгалуження в мові Java відносяться if, if else, ?: і switch.

В кожному варіанті лабораторної роботи є три завдання. Перше завдання треба виконати використовуючи **умовну операцію**. Друге завдання передбачає використання **вкладених умовних операторів**, а третє – **оператора вибору switch**.

При рішенні завдань можуть бути корисними наступні математичні формули:

Прямі $a_1 \cdot x + b_1 \cdot y + c_1 = 0$ і $a_2 \cdot x + b_2 \cdot y + c_2 = 0$ збігаються, якщо виконується наступна умова $\frac{a_1}{a_2} = \frac{b_1}{b_2} = \frac{c_1}{c_2}$

паралельні, якщо $\frac{a_1}{a_2} = \frac{b_1}{b_2} \neq \frac{c_1}{c_2}$

Якщо прямі перетинаються, то координати точки перетинання розраховуються за наступними формулами:

$$x = \frac{(c_1 b_2 - c_2 b_1)}{(b_1 a_2 - b_2 a_1)}; \quad y = \frac{(c_2 a_1 - c_1 a_2)}{(b_1 a_2 - b_2 a_1)}.$$

- Відстань між двома точками, які задані координатами $a(x_1, y_1)$ і $b(x_2, y_2)$ розраховується за формулою: $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$

– Площа трикутника за формулою Герона

$$S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)}, \text{ де } p = \frac{a+b+c}{2}$$

$$a^2 = b^2 + c^2 - 2 \cdot b \cdot c \cdot \cos A$$

- Теорема косинусів $b^2 = a^2 + c^2 - 2 \cdot a \cdot c \cdot \cos B$

$$c^2 = a^2 + b^2 - 2 \cdot a \cdot b \cdot \cos C$$

- Площа довільного трикутника з відомими основою і висотою:

$$S = \frac{1}{2} \cdot a \cdot h \cdot \sin C$$

- Площа рівностороннього трикутника: $S = \frac{a^2 \sqrt{3}}{4}$

- Висота рівностороннього трикутника: $h = \frac{a \sqrt{3}}{2}$

- Площа трикутника, вписаного в коло: $S = \frac{abc}{4R}$

- Площа трикутника, описаного навколо кола $S = \frac{(a+b+c) \cdot r}{2}$

2.2 Приклад програми

*Обчислити значення функції з використанням умовної операції:

$$F(x) = \begin{cases} \cos(x), & \text{якщо } x > 5 \\ \sin(x), & \text{якщо } x \leq 5 \end{cases}$$

* Знайти суму мінімального та максимального з чисел a, b, c .

* Скласти програму, яка згідно введеному номеру місяця (1-12) виводить номер кварталу до якого він належить. Наприклад, 2 – I квартал.

```
import java.lang.Math;
import java.util.*;
class Lab_2{
    public static void main (String[] args){
        //***** I *****
        Scanner scan = new Scanner (System.in);
        System.out.println("Введіть x->");
        double x=scan.nextDouble();
        double f=(x>5)?Math.cos(x):Math.sin(x);
        System.out.printf("F(x)= %5.3f", f);
        System.out.println();
    }
}
```

```

//***** II *****
System.out.println("Введіть a->");
double a=scan.nextDouble();
System.out.println("Введіть b->");
double b=scan.nextDouble();
System.out.println("Введіть c->");
double c=scan.nextDouble();
double min=a, max=a;
if(b>max) max=b;
else min=b;
if(c>max)max=c;
if(c<min) min=c;
System.out.printf("Сума= %5.2f", (min+max));
System.out.println();
//***** III *****
System.out.println("Введіть n->");
int n=scan.nextInt();
switch (n) {
    case 1:case 2:case 3:System.out.println("I квартал");break;
    case 4:case 5:case 6:System.out.println("II квартал"); break;
    case 7:case 8:case 9:System.out.println("III квартал");break;
    case 10:case 11:case 12:System.out.println("IV квартал");break;

    default:System.out.println("Невірний номер");
}
}
}

```

2.3 Контрольні питання

1. Який алгоритм називається алгоритмом розгалуженої структури?
2. Опишіть алгоритм роботи умовного оператора. Яка форма умовного оператора є повною? Скороченою?
3. Дайте визначення блоку операторів. У яких випадках він використовується?
4. Які умовні оператори називаються вкладеними? Назвіть правило роботи з вкладеними умовними операторами?
5. Приведіть синтаксис запису умовної операції.
6. Опишіть алгоритм роботи оператора вибору switch.
7. Для чого призначений оператор break?
8. Поясніть правила запису і використання міток в Java. Який блок називається позначеним?
9. Яка різниця між методами printf() і print()?
10. Знайдіть і виправте помилку в наступному фрагменті програми.

```

double a = 6;int b = 2,c;
switch (a/b) {
    case 1: case 2: c = 1; break;
    case 3: case 4: c = 2; break;
    default: c = 3;
}

```


2.4 Варіанти завдань

1.*Обчислити значення функції з використанням умовної операції:

$$F(x) = \begin{cases} \frac{a}{|a - |b - x||} \\ 1, \text{ якщо знаменник} = 0 \end{cases}$$

*На площині задана точка a з координатами (x, y) . Указати, де вона розташована: на якій осі або в якій координатній чверті.

*Нехай елементами прямокутного рівнобедреного трикутника є: 1) катет a ; 2) гіпотенуза b ; 3) висота, яка опущена з вершини прямого кута на гіпотенузу h ; 4) площа S . Скласти програму, яка за заданим номером i значенням відповідного елемента обчислювала б значення всіх інших елементів трикутника.

2.*Дано точка $M(x, y)$. Привласнити $z = 1$, якщо точка належить окружності з радіусом R і центром у точці (a, b) або $z = 0$ у протилежному випадку. Завдання вирішити з використанням умовної операції.

*Дано числа a_1, b_1, c_1 і a_2, b_2, c_2 , які задають прямі $a_1 \cdot x + b_1 \cdot y + c_1 = 0$ і $a_2 \cdot x + b_2 \cdot y + c_2 = 0$, надрукувати координати точки перетинання прямих, або надрукувати, що прямі збігаються або паралельні.

*Написати програму, яка за номером дня тижня (натуральному числу від 1 до 7) видає назву дня тижня. Указати робочі та вихідні дні.

3.*Скласти програму рішення рівняння $y = \frac{1}{x^2 - 4}$ з використанням умовної операції.

*Дано три числа a, b, c . Визначити, яке з них дорівнює d . Якщо жодне не дорівнює d , то знайти $\max\{d-a, d-b, d-c\}$.

*Нехай елементами кола є радіус (перший елемент), діаметр (другий елемент) і довжина кола (третій елемент). Скласти програму, що за номером елемента запитувала б його відповідне значення та обчислювала б площу круга.

4.*Скласти програму обчислення модуля $y = |5x - 4|$ з використанням умовної операції.

*Дано три точки $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Визначити, чи будуть вони розташовані на одній прямій. Якщо ні, то обчислити периметр трикутника ABC .

*Знайти добуток цифр заданого n – значного числа. Допускаються числа до п'яти знаків включно.

5. *Скласти програму обчислення кореня $y = \sqrt{4x - 6}$ з використанням умовної операції.

*Визначити і надрукувати, який вигляд має трикутник з довільно заданими довжинами сторін a, b, c (довільний, рівнобедрений або взагалі неіснує).

* Написати програму, яка б за уведеним номером одиниці виміру (1 — дециметр, 2 — кілометр, 3 — метр, 4 — міліметр, 5 — сантиметр) і довжині відрізка L видавала б відповідне значення довжини відрізка в метрах.

6. *Скласти програму обчислення функції $y = \ln(3x - 6)$ з використанням умовної операції.

*Дано число x . Надрукувати в порядку зростання числа: $\sin x$, $\cos x$, $\ln x$. Якщо при якому-небудь x деякі з виражень не існують, то вивести відповідні повідомлення і порівнювати значення тільки ті, значення яких існують.

*Дано три числа a , b , c , які задовольняють аксіомі трикутника і число p . Виконати наступні дії: якщо $p=1$ — знайти периметр трикутника; якщо $p=2$ — знайти площу трикутника; якщо $p=3$ — знайти кут a . Інакше надрукувати слово "трикутник".

7. *Задано коло з радіусом R і прямокутний отвір зі сторонами a, b . Привласнити $f = 1$, якщо коло пройде через отвір і $f = 0$ у протилежному випадку. Завдання вирішити з використанням умовної операції.

*Відомо, що із чотирьох чисел a_1, a_2, a_3 і a_4 одне відмінне від трьох інших, рівних між собою; привласнити номер цього числа змінної n .

*Є серія вимірів елементів трикутника. Групи елементів пронумеровані. У серії в довільному порядку можуть зустрічатися такі групи елементів трикутника: 1. Основа та висота; 2. Дві сторони і кут між ними (кут заданий у радіанах); 3. Три сторони.

Розробити програму, яка запитує номер групи елементів, вводять відповідні елементи та обчислює площу трикутника. Обчислення припинити, якщо введений номер групи 0.

8. *Задано прямокутник зі сторонами a, b і круглий отвір радіуса R . Привласнити $f = 1$, якщо прямокутник пройде в отвір або $f = 0$ у протилежному випадку. Завдання вирішити з використанням умовної операції.

*Дано дійсні числа a, b, c ($a > 0$). Повністю дослідити біквадратне рівняння $ax^4 + bx^2 + c = 0$, тобто якщо дійсних коренів немає, то повинне бути видане повідомлення про це, інакше знайти дійсні корені, повідомивши, скільки з них є різними.

*Для цілого числа k від 1 до 99 надрукувати фразу "Мені k років", з розрахунком на те, що при деяких значеннях k слово "років" треба замінити на слово "рік" або "року". Наприклад, 11 років, 22 року, 51 рік.

9. * Обчислити значення функції з використанням умовної операції:

$$F(x) = \begin{cases} \frac{\cos(a)}{2a - |3b - x|} \\ 1, \text{ якщо знаменник} = 0 \end{cases}$$

*За номером деякого року Year (Year>0), визначити номер сторіччя. При цьому вважати, що сторіччя починається в першому році, а не в нульовому.

*Для цілого k у діапазоні від 1 до 100 надрукувати фразу "У лісі ми знайшли k грибів". Обов'язково узгодити закінчення слова "гриб" з числом k.

10. *Скласти програму обчислення кореня $y = \sqrt{3 - 6x}$ з використанням умовної операції.

*Дано відрізки [a, b] і [c, d] і точка A з координатою x. Визначити, чи належить дана точка одному із цих відрізків, обом або лежить поза них.

*Нехай елементами рівностороннього трикутника є: 1) сторона a; 2) площа S; 3) висота h; 4) радіус вписаного кола r; 5) радіус описаного кола R. Скласти програму, що по заданому номеру і значенню відповідного елемента обчислювала б значення всіх інших елементів трикутника.

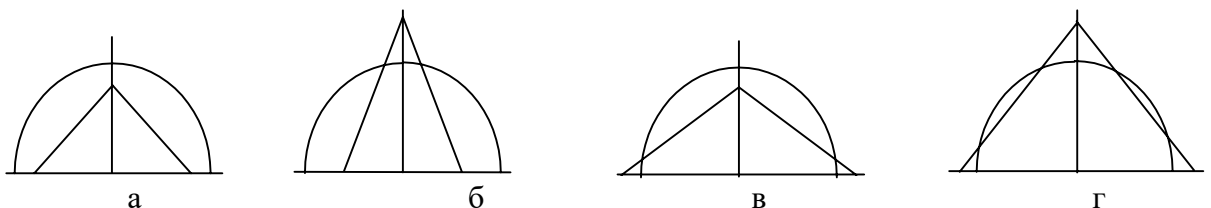
11. *Скласти програму рішення рівняння $y = \frac{1}{4 - 2x^2}$ з використанням умовної операції.

*Дано дійсні числа x1, x2, x3, y1, y2, y3. Чи належить початок координат трикутнику з вершинами (x1, y1), (x2, y2), (x3, y3)?

*Дано два дійсних додатних числа x і y. Арифметичні дії над числами пронумеровані (1 – додавання, 2 – віднімання, 3 – множення, 4 – ділення, 5 – піднесення до степені). Скласти програму, що за введеним номером виконує ту чи іншу дію над числами.

12. Скласти програму обчислення функції $y = \ln(5x+1)$ з використанням умовної операції.

*Дано рівнобедрений трикутник з основою a і висотою h і півколо з радіусом R. Визначите, чи знаходиться повністю трикутник у середині півкола (рис.а), чи виходить за межі півкола тільки вершина (рис.б), або тільки основа (рис.в), або вершина і основа (рис.г). Якщо трикутник повністю знаходиться в середині півкола, то знайти, який відсоток площі півкола він займає

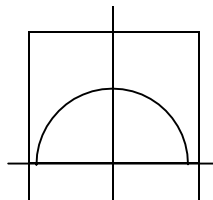


*Написати програму, яка б по введеному номеру одиниці виміру (1 – кілограм, 2 – міліграм, 3 – грам, 4 – тонна, 5 – центнер) і масі M видавала б відповідне значення маси в кілограмах.

13. *Скласти програму обчислення модуля $y=|5-4x|$ з використанням умовної операції.

*Дано квадрат зі стороною a і півколо радіуса R . Визначите, чи знаходиться півколо повністю у квадраті (рис. б) і, якщо це так, то знайдіть, яку частину площі квадрата (в %) займає півколо, інакше знайдіть кут перетинання квадрата і півколо (рис. а).

а



б

*Написати програму, яка б за введеним номером пори року (1 – зима, 2 – весна, 3 – літо, 4 – осінь) видавала відповідні цієї пори року місяці і кількість днів у кожному з місяців.

14. *Дано три числа a, b, c . Привласнити $f = \frac{a+b+c}{a \cdot b \cdot c}$ або $f = 0$, якщо знаменник дорівнює нулю. Завдання вирішити з використанням умовної операції.

*Дано координати трьох вершин деякого прямокутника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , сторони якого паралельні осям координат OX і OY . Знайти координати четвертої вершини.

*По номеру місяця року визначите пору року, до якого він відноситься. (наприклад, травень – весна).

15. *Дано числа a і b . Обчислити якщо це можливо $f = \sqrt{a \cdot b}$ або $f = 0$ у противному випадку. Завдання вирішити з використанням умовної операції.

* Дано чотири точки $A_1 (x_1, y_1)$, $A_2 (x_2, y_2)$, $A_3 (x_3, y_3)$, $A_4 (x_4, y_4)$. Визначити, чи будуть вони вершинами паралелограма.

*Написати програму, що по номеру дня тижня (натуральному числу від 1 до 7) видає розклад занять у вашій групі в цей день.

3. Прилади, устаткування та інструменти

Для виконання лабораторної роботи використовується ПЕОМ з установленим пакетом Sun Microsystems JDK 1.5 і вище та інтегрованим середовищем розробки BlueJ. Для написання програми на Java може бути використаний будь-який текстовий редактор, наприклад, Notepad, WordPad в MS Windows і ін.

4. Правила техніки безпеки та охорони праці

Правила техніки безпеки при виконанні лабораторної роботи регламентуються «Правилами техніки безпеки при роботі в комп'ютерній лабораторії».

5. Порядок проведення лабораторної роботи

Для виконання роботи кожен студент повинен:

1. Відповісти на контрольні питання та пройти усне опитування за теоретичним матеріалом лабораторної роботи;
2. Пройти інструктаж за правилами охорони праці;
3. Запустити на комп'ютері інтегроване середовище розробки BlueJ;
4. Ознайомитись з правилами роботи в середовищі BlueJ та запустити на виконання просту програму, розглянуту на лекційному занятті;
5. Отримати варіант завдання у викладача;
6. Скласти алгоритм розв'язання задачі;
7. Записати код програми на комп'ютері;
8. Відкомпілювати програму та виправити всі помилки;
9. Запустити програму на виконання;
10. Отримати результати роботи програми і показати їх викладачу;
11. Підготувати і захистити звіт до лабораторної роботи.

6. Оформлення і захист звіту

Підготовлений до захисту звіт до лабораторної роботи повинен містити:

1. титульний лист, де вказані номер і назва лабораторної роботи, відомості про виконавця;
2. номер варіанта роботи та текст завдання;
3. відповіді на контрольні запитання до лабораторної роботи;
4. текст програми алгоритмічною мовою Java;
5. лістинг результатів виконання програми.

Лабораторна робота № 3

«Складання програм циклічної структури. Цикли while, do і for»

1. Мета роботи

Отримання навичок програмування ітераційних алгоритмів та алгоритмів циклічної структури за допомогою циклів while, do і for.

2. Завдання до лабораторної роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

2.1 Методичні вказівки

Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Цикли. Ітераційні алгоритми.
- Циклічні конструкції while і do- while.
- Цикл з параметром for.
- Оператор кома. Оператор continue і return.
- Мітки. Організація переривань за допомогою оператора break.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

При рішенні ряду завдань виникає необхідність багаторазового повторення однотипних дій при різних значеннях параметрів, що визначають ці дії. Алгоритми, що реалізують такі обчислення, називаються циклічними, а повторювані ділянки обчислень – циклами. Використання циклів дозволяє виконати більші об'єми обчислень за допомогою компактних програм. Розрізняють цикли із заданим і невідомим числом повторень. До останнього відносяться ітераційні цикли з необхідною точністю. Бувають цикли з перевіркою умови перед початком виконання тіла циклу, по закінченню виконання тіла або усередині тіла циклу.

2.2 Приклад програми

* Знайти $y = \ln(2 + \ln(4 + \dots (20 + \ln(22)) \dots))$;

* Знайти суму членів ряду $\sum_{n=1}^{\infty} \frac{n+1}{2n!}$ с заданою точністю $EPS < 10^{-4}$

```

import java.lang.Math;
class lab_4{
public static void main (String[] args){
//***** I *****
double y=Math.log(22);
for(int i=20; i>=2; i-=2)
y=Math.log(i+y);
System.out.printf("Y= %6.3f",y);
System.out.println();
//***** II *****
double sum=0;
double EPS=1E-4;
double n=1, f=1;
while(Math.abs((n+1)/(2*f))>=EPS) {
sum+=(n+1)/(2*f); n++; f*=n;
}
System.out.printf("Сума= %5.2f",sum);
System.out.println();
}}

```

2.3 Контрольні питання

1. Який алгоритм називається ітераційним? Дайте визначення ітераційним циклом.
2. Опишіть алгоритми виконання циклів while і do. У чому їхня відмінність?
3. Приведіть синтаксис запису оператора циклу for і опишіть алгоритм його роботи.
4. Як організувати нескінченний цикл за допомогою оператора for?While?
5. Чим відрізняється дія операторів break и continue?
6. Яке призначення має оператор кома?

2.4 Варіанти завдань

1. *Обчислити найбільший спільний дільник натуральних чисел А і В.
 *Обчислити $y = \sin 1 + \sin 1.1 + \sin 1.2 + \dots + \sin 1.N$ (де N – ціле число ≤ 10).
 *Використовуючи формулу прямокутників, приблизно обчислити визначений інтеграл в межах 0 – π від функції $\ln(2 + \sin x)$. Відповідно цій формулі інтеграл в межах від А до В для функції f(x) дорівнює

$$h * (f(x_0) + f(x_1) + \dots + f(x_{N-1})),$$
 де $h=(B-A)/N$, $x_i=A+i*h - h/2$. В нашому випадку вважати N=100.
2. *Знайти перший від’ємний член послідовності $\cos(\operatorname{ctg} N)$, коли N приймає значення 1, 2, 3,
 *Знайти, на якому кроці в ряду $\sin X, \sin(\sin X), \sin(\sin(\sin X)), \dots$ зустрінеться перше число, що за модулем менше за $5 \cdot 10^{-2}$. Надрукувати номер

крока і число.

*Обчислити суму квадратів всіх цілих чисел, які потрапляють в інтервал $(\ln X, e^X)$, явно, що X повинен бути більше 1.

3. * Обчислити $p = \left(1 - \frac{1}{2^2}\right) \cdot \left(1 - \frac{1}{3^2}\right) \cdot \dots \cdot \left(1 - \frac{1}{n^2}\right)$, при $n \geq 10$.

* Обчислити $y = \cos X + \cos X^2 + \dots + \cos X^N$.

* Обчислити $1! + 2! + 3! + \dots + n!$

4. *Обчислити $y = \cos(1 + \cos(2 + \dots \cos((n-1) + \cos(n) \dots))$, при $n \geq 40$.

*Логічній змінній присвоїти значення true, якщо серед чисел $\sin X^N$ є хоча б одне від'ємне число. N повинно змінюватися від 1 до 30.

* Обчислити $S = \sum_{k=1}^{12} \frac{\sum_{n=1}^k \sin kn}{k!}$

5. *Підрахувати кількість цифр в десятковому запису цілого невід'ємного числа K .

*Обчислити $\sqrt{3 + \sqrt{6 + \dots + \sqrt{96 + \sqrt{99}}}}$

* Обчислити $p = \prod_{i,j=1}^{10} \frac{1}{i + j^2}$

6. *Обчислити за схемою Горнера значення багаточлена виду:

$$y = x^{10} + 2x^9 + 3x^8 + \dots + 10x + 11.$$

* Обчислити $y = (2n)!! = 2 \cdot 4 \cdot \dots \cdot (2n)$, n ввести > 0 .

*Логічній змінній присвоїти true, якщо введене ціле число N являється простим. Інакше логічна змінна повинна отримати значення false.

7. *Обчислити за схемою Горнера значення багаточлена виду:

$$y = 11x^{10} + 10x^9 + 9x^8 + \dots + 2x + 1.$$

*Обчислити $y = (2n-1)!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot (2n-1)$, n ввести ($n > 0$).

*Визначити кількість п'ятизначних натуральних чисел, сума цифр яких дорівнює заданому числу N (видно, що $1 \leq N \leq 45$). При обчисленнях неможна використовувати операції ділення ($/$, $\%$).

8. *Не використовуючи стандартні функції (за виключенням abs), Обчислити з точністю $\text{Eps} > 0$: $y = e^x = 1 + x/1! + x^2/2! + \dots + x^n/n! + \dots$. Вважати, що необхідна точність досягнена, якщо чергова складова за модулем менше Eps , — всі останні складові можна не враховувати.

*Визначити чи являється задане число досконалим, тобто дорівнює сумі всіх

(додатних) дільників, крім самого цього числа (наприклад, число 6 досконале: $6=1+2+3$).

*Знайти двохзначне число, яке дорівнює строєному добутку його цифр.

9. *Обчислити з похибкою не більше 10^{-4} суму ряду

$$S=1+x^2/2!+x^4/4!+x^6/6!+\dots+x^{2n}/(2n)!+\dots$$

*Знайти тризначне число, що дорівнює добутку його цифр, помноженому на вісім (8). Не використовувати операції /, %.

*Використовуючи формулу прямокутників, приблизно обчислити визначений інтеграл на інтервалі $[-\pi/2, \pi/2]$ від функції $\sqrt{|x| + \cos^2(x)}$.

Відповідно цій формулі інтеграл в межах від А до В для функції $f(x)$ дорівнює $h \cdot (f(x_0) + f(x_1) + \dots + f(x_{N-1}))$,

де $h=(B-A)/N$, $x_i=A+i \cdot h - h/2$. В нашому випадку $N=80$.

10. *Не використовуючи стандартні функції (за виключенням Abs), обчислити з точністю $Eps>0$: $y = \cos(x) = 1-x^2/2!+x^4/4!-x^6/6!+\dots +(-1)^n \cdot x^{2n}/(2n)!$

Вважати, що необхідна точність досягнена, якщо чергова складова по модулю менше Eps , — всі останні складові можна не враховувати.

*В яких двохзначних числах здвоєна сума цифр дорівнює їх добутку?

*Знайти суму квадратів парних чисел від 2 до 20 (2, 4, 6, 8, 10, 12, 14, 16, 18, 20).

11. *Сформувати послідовність, елементи якої обчислюються за формулою: $a_n=n/(n^2+1)$, $n=1,2,\dots,20$, та знайти суму цієї послідовності.

*Знайти, на якому кроці в ряду $\sqrt{x}, \sqrt{\sqrt{x}}, \sqrt{\sqrt{\sqrt{x}}}, \dots$, зустрінеється перше число, що менше за $1+10^{-4}$. Надрукувати номер кроку і число.

*Числа Фібоначі f_n визначаються формулами $f_0=f_1=1$; $f_n=f_{n-1}+f_{n-2}$ при $n=2,3,\dots$. Визначити десяте число Фібоначі.

12. *Обчислити добуток m членів арифметичної прогресії, якщо відомі значення першого члена a_1 і різниця арифметичної прогресії h .

*Використовуючи формулу прямокутників, приблизно обчислити визначений інтеграл в межах $[-\pi/2, \pi/2]$ від функції $\frac{\sin x}{1+|x|}$. Згідно цій формулі інтеграл в межах від А до В для функції $f(x)$ дорівнює

$$h \cdot (f(x_0) + f(x_1) + \dots + f(x_{N-1})),$$

де $h=(B-A)/N$, $x_i=A+i \cdot h - h/2$. У нашому випадку $N=100$.

*Знайти, на якому кроці в ряду $\ln X, \ln(\ln X), \ln(\ln(\ln X)), \dots$ зустрінеється перше від'ємне число. Надрукувати номер кроку і число.

13. *Знайти всі п'ятизначні числа, сума цифр яких більша за N і менша за $2 \cdot N$; ($1 < N < 25$).

*Знайти, з похибкою не більше $\varepsilon=10^{-3}$ добуток $p = \left(1 + \frac{1}{2}\right) \cdot \left(1 + \frac{1}{2 \cdot 3}\right) \cdot \dots \cdot \left(1 + \frac{1}{n!}\right)$.

Вважати, що добуток розрахований з достатньою точністю, якщо $\frac{1}{n!} < \varepsilon$.

*Обчислити $1! + 2 \cdot 2! + 3 \cdot 3! + \dots + n \cdot n!$

14. *Знайти всі додатні чотирьохзначні числа, в яких сума двох перших цифр дорівнює сумі двох останніх цифр. Не використовувати операції /, %.

*Використавши формулу трапеції, приблизно обчислити визначений інтеграл в межах $[\pi/2, 3\pi/2]$ від функції $\frac{\sin 2x}{x}$. Згідно цієї формули інтеграл в межах від А до В для функції $f(x)$ дорівнює $h \cdot (f(x_0) + 2 \cdot f(x_1) + 2 \cdot f(x_2) + \dots + 2 \cdot f(x_{N-3}) + 2 \cdot f(x_{N-2}) + f(x_{N-1}))/2$, де $h=(B-A)/N$, $x_i=A+i \cdot h$, $0 < i < N$. Взяти $N=100$.

*Обчислити $\ln(5 + \ln(10 + (15 + \ln(20 + \dots + \ln(100) \dots)))$

15. *Обчислити за схемою Горнера значення багаточлена виду:

$$y=2 \cdot x^{10} + 4 \cdot x^9 + 6 \cdot x^8 + \dots + 20 \cdot x + 22.$$

*Використавши формулу трапеції, приблизно обчислити визначений інтеграл в межах $[\pi, 2\pi]$ від функції $\frac{\cos x}{1+2x}$. Згідно цієї формули інтеграл в межах від А до В для функції $f(x)$ дорівнює $h \cdot (f(x_0) + 2 \cdot f(x_1) + 2 \cdot f(x_2) + \dots + 2 \cdot f(x_{N-3}) + 2 \cdot f(x_{N-2}) + f(x_{N-1}))/2$, де $h=(B-A)/N$, $x_i=A+i \cdot h$, $0 < i < N$. Взяти $N=80$.

*Числа Фібоначі f_n визначаються формулами $f_0=f_1=1$; $f_n=f_{n-1}+f_{n-2}$ при $n=2,3,\dots$. Знайти f – перше число Фібоначі, більше m ($m>1$).

3. Прилади, устаткування та інструменти

Для виконання лабораторної роботи використовується ПЕОМ з установленим пакетом Sun Microsystems JDK 1.5 і вище та інтегрованим середовищем розробки BlueJ. Для написання програми на Java може бути використаний будь-який текстовий редактор, наприклад, Notepad, WordPad в MS Windows і ін.

4. Правила техніки безпеки та охорони праці

Правила техніки безпеки при виконанні лабораторної роботи регламентуються «Правилами техніки безпеки при роботі в комп'ютерній лабораторії».

5. Порядок проведення лабораторної роботи

Для виконання роботи кожен студент повинен:

1. Відповісти на контрольні питання та пройти усне опитування за теоретичним матеріалом лабораторної роботи;
3. Пройти інструктаж за правилами охорони праці;
4. Запустити на комп'ютері інтегроване середовище розробки BlueJ;
5. Отримати варіант завдання у викладача;
6. Скласти алгоритм розв'язання задачі;
7. Записати код програми на комп'ютері;
8. Відкомпілювати програму та виправити всі помилки;
9. Запустити програму на виконання;
10. Отримати результати роботи програми і показати їх викладачу;
11. Підготувати і захистити звіт до лабораторної роботи.

6. Оформлення і захист звіту

Підготовлений до захисту звіт до лабораторної роботи повинен містити:

1. титульний лист, де вказані номер і назва лабораторної роботи, відомості про виконавця;
2. номер варіанта роботи та текст завдання;
3. відповіді на контрольні запитання до лабораторної роботи;
4. текст програми алгоритмічною мовою Java;
5. лістинг результатів виконання програм.

Лабораторна робота № 4

«Робота з лінійними масивами. Двовимірні масиви»

1. Мета роботи

Метою лабораторної роботи є отримання практичних навичок програмування алгоритмів обробки лінійних та двовимірних масивів. Вивчення алгоритмів упорядкування масивів методом вставки, вибору та лінійного сортування.

2. Завдання до лабораторної роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму розрахунку заданих величин.

У лабораторній роботі для ініціалізації елементів масиву слід використовувати генератор випадкових чисел. Вибирати $n > 20$. Виводити значення в рядок по 10 елементів.

2.1 Методичні вказівки

■ Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Масиви. Опис масивів.
- Операції над масивами.
- Методи сортувань елементів масивів.
- Багатомірні масиви.
- Приклади рішення завдань із використанням багатомірних масивів.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

■ Розглянемо деякі прості алгоритми сортувань лінійних масивів:

Алгоритм сортування лінійного цілого масиву за зростанням методом вибору

```
int i, j, nom, min;
for( i=0; i < n-1; i++) {nom=i; min=a[i];
    for( j=i+1; j < n; j++) if ( a[j] < min ) {nom=j;min=a[j];}
    a[nom] = a[i]; a[i] = min;}
```

Алгоритм сортування цілого масиву за зростанням методом лінійного сортування

```
int i, j, x;
for( i=0; i < n-1; i++) { for( j = i+1; j < n; j++) {
    if ( a[i] > a[j] ) { x=a[i]; a[j]=a[i]; a[i]=x;
    }}}}
```

Алгоритм сортування лінійного цілого масиву за зростанням методом вставки

```
int i, j, x;
for (i=1; i < n; i++){x =
    [i];
```

```

for ( j=i-1; j>=0 && a[j] > x; j--)
a[j+1] = a[j]; a[j+1] = x;}

```

■ Розглянемо фрагменти програм, де використовуються алгоритми сортувань двовимірних масивів за рядками і за стовбцями:

- сортування методом вибору рядків матриці за зростанням максимальних елементів у рядках

```

/* Оголошення двовимірного масиву*/double a[][]=new double[n][m];
. . .
/* Знаходження максимальних елементів за рядками*/double max[]=new
double[a.length];
. . .
// Сортування матриці методом вибору
double c[]=new double[m]; // допоміжний масив, для проміжного зберігання
int nom; double min;
for( int i=0; i < a.length-1; i++) {nom=i; min=max[i];
for(int j=i+1; j < a.length; j++)if ( max[j] < min ) {
nom=j; min=max[j]; }
max[nom] = max[i]; max[i] = min; c=a[nom]; a[nom]=a[i]; a[i]=c;}
. . .

```

- сортування методом вставки рядків матриці за убутанням максимальних елементів у рядках

```

. . .
double x; int j;
for ( int i=1; i < n; i++) {x =
max[i]; c=a[i];
for ( j=i-1; j>=0 && max[j] < x; j--){
max[j+1] = max[j]; a[j+1]=a[j];
}
max[j+1] = x; a[j+1]=c;
}
. . .

```

- сортування методом лінійного сортування рядків матриці за зростанням максимальних елементів у рядках

```

. . .
int i, j; double x;
for( i=0; i < n-1; i++) { for( j = i+1; j < n; j++) {
if ( max[i] > max[j] ) { x=max[i]; max[i]=max[j]; max[j]=x;
c=a[i]; a[i]=a[j]; a[j]=c;
}}}
. . .

```

- сортування методом вибору стовпців матриці за зростанням максимальних елементів у стовпцях

```

/* Оголошення двовимірного масиву*/double a[][]=new double[n][m];
. . .
/* Знаходження максимальних елементів за стовпцями*/double max[]=new double[m];
for(int i=0; i<m; i++){max[i]=a[0][i]; for(int j=0; j<n; j++){
if ( a[j][i]>max[i]) max[i]=a[j][i];}}

/*Сортування масиву методом вибору*/double c[]=new double[n];
int nom; double min;
for( int i=0; i < m-1; i++) {nom=i; min=max[i];
for(int j=i+1; j < m; j++)if ( max[j] < min ) { nom=j; min=max[j];

```

```

    }
    max[nom] = max[i]; max[i] = min; for(int k=0; k<n; k++)
    { c[k]=a[k][nom]; a[k][nom]=a[k][i]; a[k][i]=c[k]; }
    . . .

```

- сортування методом вставки стовпців матриці за убутанням максимальних елементів у стовпцях

```

. . .
double x; int j;
for ( int i=1; i < m; i++) { x =
    max[i];
    for ( int k=0; k < n; k++)
        c[k]=a[k][i];
    for ( j=i-1; j>=0 && max[j] < x; j--)
        {max[j+1] = max[j];
        for ( int k=0; k < n; k++)
            a[k][j+1]=a[k][j];}
    max[j+1] = x;
    for ( int k=0; k < n; k++)
        a[k][j+1]=c[k];}
. . .

```

- сортування методом лінійного сортування стовпців матриці за зростанням максимальних елементів стовпцях

```

. . .
int i, j; double
x;
for( i=0; i < m-1; i++) { for( j =
    i+1; j < m; j++) {
    if ( max[i] > max[j] ) { x=max[i];
    max[i]=max[j]; max[j]=x;
    for( int k=0; k < m-1; k++)
        {c[k]=a[k][i]; a[k][i]=a[k][j]; a[k][j]=c[k];}
    }}}
. . .

```

Приклади програм

Для лінійного масиву, що містить n дійсних елементів, розрахувати суму за модулем всіх його елементів. "Стиснути" масив, видаливши з нього всі від'ємні елементи. Упорядкувати елементи нового масиву за зростанням методом вставки.

```

import java.util.*; class lab_5{
public static void main (String[] args) {
Scanner scan = new Scanner (System.in);
System.out.println("Введіть n->");
int n=scan.nextInt();
double[] a=new double[n];
for(int i=0;i<a.length;i++) a[i]=Math.random()*20-10;

System.out.println("Исходный массив");
for(int i=0;i<a.length;i++){ System.out.printf("%7.2f",a[i]); if(i%10==9)
    System.out.print("\n");
}

double sum=0;
for(int i=0;i<a.length;i++)
sum+=Math.abs(a[i]);
System.out.print("\n");
System.out.printf("Сумма = %7.2f",sum);
System.out.println();
}

```

```

int col=0;
double[] b=new double[n]; for(int i=0;i<a.length;i++)
if(a[i]>=0) {b[col]=a[i]; col++;}System.out.print("\n");

System.out.println("Сжатый массив");
for(int i=0;i<col;i++){
System.out.printf("%7.2f",b[i]);
if(i%10==9) System.out.print("\n");}int i,j;
double x;
for(i=1; i<col; i++){x=b[i];
for(j=i-1; j>=0&&b[j]>x;j--)
b[j+1]=b[j]; b[j+1]=x;}
System.out.print("\n");

System.out.println("Упорядоченный массив");for(i=0;i<col;i++){
System.out.printf("%7.2f",b[i]);
if(i%10==9) System.out.print("\n");
}
System.out.print("\n");
}}

```

Дано двовимірний цілий масив. Упорядкувати рядки матриці за убутанням кількості додатних елементів у рядку. Сортування методом вибору. Знайти суму парних елементів у кожному стовпці матриці.

```

import java.util.*;class Lab1{
public static void main(String[] args){
Scanner scan = new Scanner (System.in);
System.out.println("Кількість рядків:");
int n=scan.nextInt();
System.out.println("Кількість стовпців:");
int m=scan.nextInt();
int a[][]=new int[n][m];

int count[]=new int[a.length];int sum[]=new int[m];
int c[]=new int[m];

//Заповнення елементів матриці за допомогою генератору випадкових чисел
for(int i=0;i<a.length;i++)
for(int j=0;j<a[i].length;j++)
a[i][j]=(int) (Math.random()*20-10);
System.out.println("Вихідний двовимірний масив ");
for(int i=0;i<a.length;i++){
for(int j=0;j<a[i].length;j++) System.out.printf("%8d",a[i][j]);
System.out.println();}

//Знаходження кількості додатних елементів за рядками
for(int i=0;i<a.length;i++){
count[i]=0;
for(int j=0;j<a[i].length;j++)
if (a[i][j]>0) count[i]++;}

//Сортування масиву методом вибору
int nom, min;
for( int i=0; i < a.length-1; i++) {nom=i; min= count[i];
for(int j=i+1; j < a.length; j++)if ( count[j] > min ) {
nom=j; min= count[j];
}
count[nom] = count[i]; count[i] = min;c=a[nom]; a[nom]=a[i]; a[i]=c;
}

//Знаходження суми парних елементів у кожному стовпці
for(int i=0;i<m;i++){sum[i]=0;
for(int j=0;j<n;j++) if(a[j][i]%2==0) sum[i]+=a[j][i];
}
}

```

```
// Виведення упорядкованого масиву System.out.printf("%s%40s%n", "Відсортований двовимірний масив", "Кількість дод. елементів");
for(int i=0;i<a.length;i++){ for(int j=0;j<a[i].length;j++){
System.out.printf("%8d",a[i][j]);}
System.out.printf("%16d%n",count[i]);} System.out.println("Сума парних елементів за стовпцями ");for(int j=0;j<m;j++){
System.out.printf("%8d",sum[j]);
}}
```

2.3 Контрольні питання

1. Дайте визначення масиву.
2. Який формат має оголошення масиву?
3. Як здійснюється виділення пам'яті під масив? Приведіть приклади.
4. Як здійснюється доступ до елементів масиву? Значення яких типів можуть бути індексами масиву? Як визначити довжину масиву?
5. Які операції можуть бути застосовані до посилань на масиви?
6. Як у програмі організувати введення та виведення масивів?
7. Опишіть процедури упорядкування елементів масиву при використанні різних методів сортувань?
8. Як оголошуються в Java багатомірні масиви? Скільки одномірних масивів створюється при оголошенні `int x[][]=new int[3][5];`?
9. Як здійснюється доступ до елементів двовимірного масиву?
10. Чому буде дорівнюватися змінна `len` після наступного оголошення та ініціалізації масиву `x`

```
int x[][] = {{-1,2,3}, null, {0,3}, {8,0,4}};
int len=x[3].length/ x[2].length;
```
11. Чому буде дорівнюватися змінна `len` після наступного оголошення й ініціалізації масиву `x`

```
int x[][]=new int[3][5];
x[0]=new int[7];
x[1]=new int[2];
x[2]=null;
int len= x.length+x[1].length;
```
12. Як перетворити одномірний масив у двовимірний і навпаки?

2.4 Варіанти завдань

1. *Дано масив дійсних чисел, що містить n елементів. Елементи масиву не можуть дорівнювати нулю. Розрахувати корінь ступеня n з абсолютного значення добутку всіх елементів масиву (середнє геометричне всіх елементів). "Стиснути" масив, видаливши з нього кожний другий елемент. Упорядкувати елементи нового масиву за зростанням **методом лінійного сортування**

*Дано двовимірний дійсний масив. Упорядкувати рядки матриці за зростанням максимальних елементів у рядках. Сортування методом вибору. Знайти добуток елементів, які за модулем більше 2, у кожному стовпці матриці.

2. *Дано масив дійсних чисел, що містить n елементів. Обчислити суму всіх елементів масиву так, щоб парні елементи входили до неї зі зворотними

знаками. Сформуйте два масиви: у перший включіть елементи вихідного масиву з парними номерами, а в другий – з непарними. Масив з парними номерами відсортуйте за зростанням **методом вибору**.

*Дано двовимірний дійсний масив. Упорядкувати рядки матриці за зростанням суми додатних елементів у рядках. Сортування методом вставки. Знайти мінімум у кожному стовпці матриці.

3. *Дано масив додатних цілих чисел, що містить n елементів. Обчислити суму добутків першого елемента на останній, другого на передостанній і т.д. Утворити новий масив, елементами якого будуть елементи вихідного, що закінчуються на цифру k . Упорядкувати новий масив за зростанням **методом вставки**.

*Дано двовимірний цілий масив. Упорядкувати рядки матриці за зростанням початкових елементів у рядках. Сортування методом лінійного сортування. Знайти максимальний за модулем елемент у кожному рядку матриці. Знайти для всієї матриці кількість елементів, які кратні 7.

4. *Дано масив дійсних чисел, що містить n елементів. Обчислити суму квадратів всіх елементів масиву. Знайти в масиві мінімальний і максимальний елемент. Упорядкувати елементи масиву, що розташовані між мінімальним і максимальним за зростанням **методом лінійного сортування**.

*Дано двовимірний цілий масив. Упорядкувати рядки матриці за зростанням кількості непарних елементів у рядках. Сортування методом вибору. Знайти максимальний для всієї матриці елемент і номер рядка й стовпця, у якому він знаходиться.

5. *Дано масив дійсних чисел, що містить n елементів. Знайти середнє арифметичне всіх елементів масиву. В одномірному масиві всі від'ємні елементи слід перемістити в початок масиву, а інші – у кінець зі збереженням порядку розташування. Від'ємні елементи упорядкувати за зростанням **методом вибору**.

*Дано двовимірний дійсний масив. Упорядкувати рядки матриці за зростанням суми від'ємних елементів у рядку. Сортування методом лінійного сортування. Знайти максимум у кожному стовпці матриці.

6. *Дано масив цілих чисел, що містить n елементів. Обчислити значення багаточлена ступеня n у точці X_0 , вважаючи елементи масиву коефіцієнтами цього багаточлена. "Стиснути" масив, видаливши з нього всі нульові елементи. Упорядкувати новий масив за зростанням **методом вставки**.

*Дано двовимірний цілий масив. Упорядкувати рядки матриці за зростанням початкових елементів у рядках. Сортування методом вставки. Знайти максимальний елемент у кожному стовпці й кількість непарних елементів у всій матриці.

7. *Дано масив дійсних чисел, що містить n елементів. Знайти суму добутків непарних елементів (елементів з непарними номерами) на наступні за

ними парні елементи (з парними номерами). Переставити елементи масиву так, щоб на початку стояли додатні, потім нульові і від'ємні значення в порядку їхнього розташування. Від'ємні елементи упорядкувати за зростанням **методом лінійного сортування**.

*Дано двовимірний дійсний масив. Упорядкувати рядки матриці за убутанням мінімальних елементів у рядках. Сортування методом вставки. Знайти кількість від'ємних елементів у кожному стовпці матриці.

8. *Дано масив дійсних чисел, що містить n елементів. Знайти кількість елементів більших за середнє арифметичне. "Стиснути" масив, видаливши з нього всі від'ємні елементи. Упорядкувати першу половину нового масиву за убутанням **методом вибору**.

*Дано двовимірний цілий масив. Упорядкувати рядки матриці за зростанням кількості парних елементів у рядках. Сортування методом лінійного сортування. Знайти мінімальний елемент серед додатних елементів у кожному стовпці матриці.

9. *Дано масив додатних цілих чисел, що містить n елементів. Перевірити, чи виконується для елементів масиву наступне відношення $a[i] \leq a[i+1] \leq a[i+2]$ і видати відповідне повідомлення. Утворити новий масив, елементами якого будуть елементи вихідного, кратні k . Упорядкувати новий масив за зростанням **методом вставки**.

*Дано двовимірний дійсний масив. Упорядкувати рядки матриці за убутанням кількості нульових елементів у рядках. Сортування методом вибору. Знайти максимальний елемент серед від'ємних елементів у кожному стовпці матриці.

10. *Дано масив цілих чисел, що містить n елементів. Знайти номер останнього парного від'ємного елемента. Якщо цей елемент перебуває в першій половині масиву, то знайти кількість елементів рівних першому парному від'ємному елементу. Упорядкувати другу половину масиву за убутанням **методом лінійного сортування**.

*Дано двовимірний дійсний масив. Упорядкувати стовпці матриці за зростанням максимальних елементів у стовпцях. Сортування методом вставки. Знайти середнє арифметичне значення елементів у кожному рядку матриці.

11. *Дано масив дійсних чисел, що містить n елементів. Знайти найбільший елемент, який має парний номер. Визначити чи є він єдиним. "Стиснути" масив, залишивши тільки різні елементи (тобто видаливши всі однакові елементи). Упорядкувати новий масив за зростанням **методом вибору**.

*Дано двовимірний цілий масив. Упорядкувати стовпці матриці за убутанням середніх арифметичних значень елементів у стовпцях. Сортування методом вибору. Знайти добуток додатних елементів у кожному рядку матриці.

12. * Дано масив дійсних чисел, що містить n елементів. Підрахувати кількість елементів, для яких виконується нерівність $i \cdot i < a[i] < i!$. "Стиснути" масив, видаливши елементи, що задовольняють даній умові. Упорядкувати новий масив за зростанням **методом вставки**.

*Дано двовимірний дійсний масив. Упорядкувати стовпці матриці за зростанням кількості елементів, що лежать у діапазоні від A до B , у стовпцях. Сортування методом лінійного сортування. Знайти максимум у кожному рядку матриці.

13. *Дано масив додатних цілих чисел, що містить n елементів. Підрахувати кількість елементів, які збігаються зі своїм номером i при цьому кратні 3. "Стиснути" масив, видаливши елементи, що задовольняють даній умові. Упорядкувати новий масив за зростанням **методом лінійного сортування**.

*Дано двовимірний дійсний масив. Упорядкувати стовпці матриці за зростанням мінімальних за модулем елементів у стовпцях. Сортування методом вибору. Знайти кількість ненульових елементів у кожному рядку матриці.

14. * Дано масив дійсних чисел, що містить n елементів. Змінної t привласнити значення "істина", якщо в масиві є хоча б один елемент, що ділиться без остачі на X і не ділиться без остачі на Y (X і Y треба ввести). Упорядкувати другу половину масив за зростанням **методом вибору**.

*Дано двовимірний цілий масив. Упорядкувати стовпці матриці за зростанням добутку елементів у стовпцях. Сортування методом вставки. Знайти мінімум у кожному рядку матриці.

15. * Дано масив дійсних чисел, що містить n елементів. Розрахувати середнє арифметичне тих елементів масиву, які розташовані ліворуч від першого максимального елемента. Упорядкувати елементи масиву ліворуч від першого максимального елемента за зростанням **методом вставки**.

*Дано двовимірний дійсний масив. Упорядкувати стовпці матриці за зростанням максимальних за модулем елементів у стовпцях. Сортування методом лінійного сортування. Знайти добуток парних елементів у кожному рядку матриці.

3. Прилади, устаткування та інструменти

Для виконання лабораторної роботи використовується ПЕОМ з установленим пакетом Sun Microsystems JDK 1.5 і вище та інтегрованим середовищем розробки BlueJ або Eclipse. Для написання програми на Java може бути використаний будь-який текстовий редактор, наприклад, Notepad, WordPad в MS Windows і ін.

4. Правила техніки безпеки та охорони праці

Правила техніки безпеки при виконанні лабораторної роботи

регламентуються «Правилами техніки безпеки при роботі в комп'ютерній лабораторії».

5. Порядок проведення лабораторної роботи

Для виконання роботи кожен студент повинен:

1. Відповісти на контрольні питання та пройти усне опитування за теоретичним матеріалом лабораторної роботи;
2. Пройти інструктаж за правилами охорони праці;
3. Отримати варіант завдання у викладача;
4. Скласти алгоритм розв'язання задачі;
5. Записати код програми на комп'ютері;
6. Відкомпілювати програму та виправити всі помилки;
7. Запустити програму на виконання;
8. Отримати результати роботи програми і показати їх викладачу;
9. Підготувати і захистити звіт до лабораторної роботи.

6. Оформлення і захист звіту

Підготовлений до захисту звіт до лабораторної роботи повинен містити:

1. титульний лист, де вказані номер і назва лабораторної роботи, відомості про виконавця;
2. номер варіанта роботи та текст завдання;
3. відповіді на контрольні запитання до лабораторної роботи;
4. текст програми алгоритмічною мовою Java;
5. лістинг результатів виконання програми.

Лабораторна робота № 5

«Оголошення і виклик методів. Передача в метод параметрів примітивних типів. Рекурсивні методи»

1. Мета роботи

Метою лабораторної роботи є отримання практичних навичок оголошення і виклику статичних методів. Вивчення особливостей створення рекурсивних методів.

2. Завдання до лабораторної роботи

1. Створіть клас, що містить метод, який дозволяє з точністю $Eps=1\cdot 10^{-4}$ розрахувати таблицю значень функції дійсної змінної x . Результати роботи порівняйте з результатами розрахунку відповідного методу класу `Math`.

2. При рішенні другого завдання використовуйте **рекурсивний метод**.

2.1 Методичні вказівки

Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Методи. Оголошення і виклик методів. Сигнатура методу.
- Модифікатори членів класу.
- Передача аргументів за посиланням та за значенням.
- Рекурсія. Особливості рекурсивних програм.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

Наведемо нижче декілька важливих визначень, які слід пам'ятати під час виконання лабораторної роботи.

■ **Методи** – це підпрограми, які приєднані до конкретних визначень класів. Вони описуються усередині визначення класу на тому ж рівні, що і змінні класу. При оголошенні методу задаються тип результату, що повертається ним, і список параметрів. Загальна форма оголошення методу має наступний вигляд:

тип ім'я_методу (список формальних параметрів) { тіло методу; }

■ **Сигнатура** (signature) методу визначається ім'ям методу і його параметрами (кількістю, типом та порядком слідування). Якщо для полів забороняється збіг імен, то для методів у класі заборонене створення двох методів з однаковими сигнатурами. Пам'ятайте, що тип значення, що повертає метод, не входить у сигнатуру, тому методи не можуть розрізнятися тільки типом результату їхньої роботи.

■ **Модифікатори** – службові слова, які можуть використовуватися перед описом поля або методу класу.

1. Додаванням модифікатора **private** до опису члена класу, дозволяє зробити його закритим, тобто доступ до таких членів класу (полів або методів) буде дозволений тільки усередині того класу, в якому вони оголошені.

2. У протилежність закритості можна оголосити деякі члени класу відкритими, записавши замість слова **private** модифікатор **public**. До таких методів і змінних дозволяється доступ з будь-якого класу.

3. Іноді треба визначити поле, загальне для всього класу, зміна якого в одному екземплярі викличе зміну того ж поля у всіх екземплярах. Такі поля називаються **змінними класу** (class variables). Для змінних класу виділяється тільки одна комірка пам'яті, загальна для всіх екземплярів. Змінні класу в Java утворюються модифікатором **static**.

Для **статичних методів** виконавча система Java завжди створює в пам'яті тільки одну копію машинного коду методу. Статичні методи виконуються відразу у всіх екземплярах класу. Більш того, вони можуть виконуватися, навіть якщо не створений жоден екземпляр класу. Досить уточнити ім'я методу ім'ям класу (а не ім'ям об'єкту), щоб метод міг працювати. Саме так ми користувалися методами класу **Math**, не створюючи його екземпляри, а просто записуючи **Math.abs(x)**, **Math.sqrt(x)**.

Тому статичні методи називаються **методами класу**, на відміну від нестатичних методів, які звуться **методами екземпляра**.

■ **Рекурсія** – це такий спосіб організації обробки даних, при якому програма викликає сама себе безпосередньо, або за допомогою інших програм. Метод **P** називається **рекурсивним**, якщо при виконанні тіла методу відбувається виклик методу **P**.

Прикладами рекурсивних функцій можуть служити факторіал числа і числа Фібоначчі. У кожному з цих випадків значення функції для всіх значень аргументу, починаючи з деякого, визначається через попередні значення. Вирішуючи деяку задачу, рекурсивний алгоритм викликає сам себе для вирішення її підзадач.

■ Розглянемо декілька прикладів розв'язання задач за допомогою рекурсивних методів:

Рекурсивне обчислення значення факторіала натурального числа

Факторіал $n!$ цілого невід'ємного числа n задається наступними співвідношеннями: $0!=1$, $n!=n \cdot (n-1)!$ для $n>0$

```
import java.util.*; public class FactN {
    static int fac(int x) { return (x == 0) ? 1 : x * fac(x-1);
}
public static void main(String[] args){ Scanner scan = new
    Scanner(System.in); System.out.print("n=");
    System.out.printf("n!=%d", fac(scan.nextInt())); }}
```

Рекурсивне обчислення чисел Фібоначчі

Числами Фібоначчі f_n називають послідовність величин 0, 1, 1, 2, 3, 5, 8, ... , яка визначається наступним чином:

$$\begin{aligned} f_0 &= 0, \\ f_1 &= 1, & \text{для } n > 1 \\ f_n &= f_{n-1} + f_{n-2} \end{aligned}$$

```
import java.util.*;public class FibN {
    static int fib(int x) {
        return (x > 1) ? fib(x-2) + fib(x-1) : (x == 1) ? 1 : 0;
    }
    public static void main(String[] args){ Scanner scan = new
        Scanner(System.in);System.out.print("Введіть n ->");
        int n = scan.nextInt(); System.out.printf("fib(%d) =
        %d",n,fib(n));}}
```

Рекурсивне обчислення найбільшого спільного дільника двох чисел

```
import java.util.*;public class Nsd {
    static int nsd(int x, int y) {
        return (x==y)? x:(x>y)?nsd(x-y,y):nsd(x,y-x);
    }
    public static void main(String[] args){Scanner scan = new
        Scanner(System.in);System.out.print("Введіть a ->");
        int a = scan.nextInt(); System.out.print("Введіть b ->");
        int b = scan.nextInt();
        System.out.printf("НСД = %d",nsd(a,b)); }}
```

Рекурсивний метод піднесення до ступеня

```
import java.util.*;public class Rec {
    static int step(int x, int n) { return
        (n==0)?1:x*step(x,n-1);
    }
    public static void main(String[] args){ Scanner scan = new
        Scanner(System.in);System.out.print("Введіть a ->");
        int a=scan.nextInt(); System.out.print("Введіть n ->");
        int n=scan.nextInt();
        System.out.printf("Значення %d^%d=%d",a,n,step(a,n)); }}
```

Приклад програми

* Розрахуйте $\ln x = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \dots + (-1)^{n+1} \frac{(x-1)^n}{n}$, де $0 < x \leq 2$,

```
public class Zad1 {
    static final double eps=1e-4;

    public static double my_log(double x) {
        double sum=x-1,c=x-1;
        int n=1;
        while(Math.abs(c/n)>eps) {
            c*=- (x-1); sum+=c/++n; }
        return sum;
    }
    public static void main(String[] args) {
        System.out.printf("-----%n");
        System.out.printf("|      x      | my_log(x) | log(x) |%n");
        System.out.printf("-----%n");
        for(double x=0.1;x<=2;x+=(2-0.1)/5) {
            System.out.printf("|%8.4f |%8.4f |%8.4f |%n", x, my_log(x),
Math.log(x));}
        System.out.printf("-----%n");
    }
}
```

x	my_log(x)	log(x)
0,1000	-2,3019	-2,3026
0,4800	-0,7339	-0,7340
0,8600	-0,1508	-0,1508
1,2400	0,2151	0,2151
1,6200	0,4824	0,4824
2,0000	0,6931	0,6931

* Обчисліть (у виразі присутні n радикалів):

```
import java.util.*;
public class Rec {
    static double fac(int n) {
        double f;
        return (n==1)?Math.sqrt(2):Math.sqrt(2+fac(--n)); }

    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Введіть n:");
        System.out.printf("Значення=%f", fac(scan.nextInt()));
    }
}
```

Перевірка: при n=3, вираження дорівнює 1.9615705608...

2.3 Контрольні питання

1. Приведіть формат оголошення методу.
2. Що таке сигнатура методу? Які з наведених нижче методів несумісні в одному класі?

```
void get() {}  
int get() {}  
void get(int x) {}  
void get(int y) {}  
public int get() {}  
private int get() {}
```
3. У яких випадках у тілі методу використовується return-вираження?
4. Чим відрізняються формальні і фактичні параметри?
5. Дайте визначення глобальним і локальним змінним. Приведіть приклади.
6. У яких випадках при оголошенні полів і методів класу використовуються наступні модифікатори доступу: private, public, static, final?

2.4 Варіанти завдань

1. * $\ln \frac{x+1}{x-1} = 2 \sum_{n=0}^{\infty} \frac{1}{(2n+1)x^{2n+1}} = 2 \left(\frac{1}{x} + \frac{1}{3x^3} + \frac{1}{5x^5} + \dots \right)$, при $|x| > 1$
де $2 \leq x \leq 5$

* Напишіть програму обчислення

$$\sum_{k=1}^n (-1)^k \cdot k!, \text{ де } k!! = \begin{cases} 1 \cdot 3 \cdot 5 \cdot \dots \cdot k, & \text{якщо } k - \text{непарне} \\ 2 \cdot 4 \cdot 6 \cdot \dots \cdot k, & \text{якщо } k - \text{парне} \end{cases}$$

Використовуйте рекурсивний метод обчислення $k!!$

2. * $e^{-x} = \sum_{n=0}^{\infty} \frac{(-1)^n x^n}{n!} = 1 - x + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} - \dots$, при $|x| < \infty$
де $0.5 \leq x \leq 0.9$

* Скласти програму обчислення суми: $2! - 4! + 6! - \dots + n!$ ($n \leq 16$, n – парне). Використовуйте рекурсивний метод обчислення $n!$

3. * $\ln(x+1) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{n+1}}{n+1} = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots$, при $-1 < x \leq 1$
де $-0.5 \leq x \leq 0.5$

* Числа Фібоначчі u_0, u_1, u_2, \dots визначаються наступним чином: $u_0=0$; $u_1=1$; $u_n=u_{n-1}+u_{n-2}$, $n=2,3,4,\dots$. Напишіть програму обчислення першого числа Фібоначчі, що більше за m ($m>1$). Використовуйте рекурсивний метод, заснований на безпосередньому використанні співвідношення $u_n=u_{n-1}+u_{n-2}$.

4. * $\ln \frac{1+x}{1-x} = 2 \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = 2 \left(x + \frac{x^3}{3} + \frac{x^5}{5} + \dots \right)$, при $|x| < 1$

де $-0.8 \leq x \leq 0.8$

* Знайти двозначні числа (за винятком тих чисел, що закінчуються на нуль 10, 20, 30,...), у яких НСД цифр рівний 3. Використовуйте рекурсивний метод знаходження найбільшого спільного дільника двох натуральних чисел.

5. * $\ln(1-x) = - \sum_{n=1}^{\infty} \frac{x^n}{n} = - \left(x + \frac{x^2}{2} + \frac{x^3}{3} + \dots \right)$, при $-1 \leq x < 1$

де $-0.5 \leq x \leq 0.8$

* Знайдіть всі тризначні числа, які можна представити у вигляді суми факторіалів їх цифр. Використовуйте рекурсивний метод обчислення $n!$

6. * $\text{arccotg}(x) = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1} x^{2n+1}}{2n+1} = \frac{\pi}{2} - x + \frac{x^3}{3} - \frac{x^5}{5} + \dots$, при $|x| \leq 1$

де $-1 \leq x \leq 0.5$

* Числа Фібоначчі u_0, u_1, u_2, \dots визначаються наступним чином: $u_0=0$; $u_1=1$; $u_n=u_{n-1}+u_{n-2}$, $n=2,3,4,\dots$. Знайдіть суму перших n чисел Фібоначчі ($n \leq 10$). Використовуйте рекурсивний метод, заснований на безпосередньому використанні співвідношення $u_n=u_{n-1}+u_{n-2}$.

7. * $\text{arctg}(x) = \frac{\pi}{2} + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{(2n+1)x^{2n+1}} = \frac{\pi}{2} - \frac{1}{x} + \frac{1}{3x^3} - \frac{1}{5x^5} + \dots$, при $x > 1$

де $\frac{\pi}{2} \leq x \leq \pi$

* Опишіть рекурсивний метод піднесення до ступеня $\text{pow}(x,n)$, формальними параметрами якого є дійсна змінна x і натуральна змінна n , використовуйте його

для обчислення значення виразу $\sum_{k=1}^{10} k^k$

8. * $e^{-x^2} = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{n!} = 1 - x^2 + \frac{x^4}{2!} - \frac{x^6}{3!} + \frac{x^8}{4!} - \dots$, при $|x| < \infty$

де $0.2 \leq x \leq 0.8$

* Серед перших $n+1$ елементів послідовності

$$\begin{cases} x_0 = a \\ x_k = q \cdot x_{k-1} + b, \text{ де } k = 1, 2, \dots \end{cases}$$

знайти кількість тих елементів, які належать інтервалу (c,d) .

9. * $\cos(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n)!} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots$, при $|x| < \infty$

де $0.1 \leq x \leq 0.8$

* Числа Фібоначчі u_0, u_1, u_2, \dots визначаються наступним чином: $u_0=0$; $u_1=1$; $u_n=u_{n-1}+u_{n-2}$, $n=2,3,4,\dots$. Серед перших 10-ти чисел Фібоначчі знайдіть два таких, які в сумі дорівнюють 42. Використовуйте рекурсивний метод, заснований на безпосередньому використанні співвідношення $u_n=u_{n-1}+u_{n-2}$.

10. * $\frac{\sin x}{x} = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n}}{(2n+1)!} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \dots$, при $|x| < \infty$

де $\frac{\pi}{6} \leq x \leq \frac{\pi}{4}$

* Відомо, що $x_0=a$, $x_k=qx_{k-1}+b$, де $k=1,2,\dots$. Обчисліть x_n для заданого натурального n .

11. * $\ln x = 2 \sum_{n=0}^{\infty} \frac{(x-1)^{2n+1}}{(2n+1)(x+1)^{2n+1}} = 2 \left(\frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots \right)$, $x > 0$

де $2 \leq x \leq 5$

* Напишіть програму обчислення P за формулою:

$$P = \begin{cases} (n!+4)^2, & \text{якщо } n < 5 \\ n!, & \text{якщо } n \geq 5 \end{cases}$$

де n – задане натуральне число. Використовуйте рекурсивний метод обчислення $n!$

12. * $\ln x = \sum_{n=0}^{\infty} \frac{(x-1)^{n+1}}{(n+1)(x+1)^{n+1}} = \frac{x-1}{x} + \frac{(x-1)^2}{2x^2} + \frac{(x-1)^3}{3x^3} + \dots$, при $x > \frac{1}{2}$

де $0.6 \leq x \leq 0.9$

* Числа Фібоначчі u_0, u_1, u_2, \dots визначаються наступним чином: $u_0=0$; $u_1=1$; $u_n=u_{n-1}+u_{n-2}$, $n=2,3,4,\dots$. Серед перших 10-ти чисел Фібоначчі знайдіть кількість чисел, що належать відрізку $[5 \div 45]$. Використовуйте рекурсивний метод, заснований на безпосередньому використанні співвідношення $u_n=u_{n-1}+u_{n-2}$.

13. * $\text{Arth}(x) = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{2n+1} = x + \frac{x^3}{3} + \frac{x^5}{5} + \frac{x^7}{7} + \dots$, при $|x| < 1$

де $0.2 \leq x \leq 0.5$

Зауваження: $\text{Arth}(x)$ – зворотний гіперболічний тангенс;

$$\text{Arth}(x) = \frac{1}{2} \ln \frac{1+x}{1-x}, \text{ при } |x| < 1$$

* Опишіть рекурсивний метод піднесення до ступеня $\text{pow}(x,n)$, формальними параметрами якого є дійсна змінна x і натуральна змінна n , використовуйте

його для обчислення суми $\sum_{i=1}^{10} \frac{x^i}{i}$.

$$14. ** \quad * \arcsin(x) = x + \sum_{n=1}^{\infty} \frac{1 \cdot 3 \cdot \dots \cdot (2n-1) \cdot x^{2n+1}}{2 \cdot 4 \cdot \dots \cdot 2n \cdot (2n+1)} = x + \frac{x^3}{2 \cdot 3} + \frac{1 \cdot 3 \cdot x^5}{2 \cdot 4 \cdot 5} + \frac{1 \cdot 3 \cdot 5 \cdot x^7}{2 \cdot 4 \cdot 6 \cdot 7} + \dots$$

при $|x| < 1$

де $0.1 \leq x \leq 0.3$

* Числа Фібоначчі u_0, u_1, u_2, \dots визначаються таким чином: ($u_0=0$; $u_1=1$; $u_n=u_{n-1}+u_{n-2}$, $n=2,3,4,\dots$). Перевірте на декількох прикладах, чи буде u_{5k} , де $k=1,2,3,4,5$ ділитися на 10. Використовуйте рекурсивний метод заснований на використанні співвідношення $u_n=u_{n-1}+u_{n-2}$.

$$15. * \operatorname{arctg}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n x^{2n+1}}{(2n+1)} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots, \text{ при } |x| \leq 1$$

де $-1 \leq x \leq 1$

* Скласти програму обчислення суми: $1! - 2! + 3! - \dots + n!$ ($n \leq 15$).

Використовуйте рекурсивний метод обчислення $n!$

3. Прилади, устаткування та інструменти

Для виконання лабораторної роботи використовується ПЕОМ з установленим пакетом Sun Microsystems JDK 1.5 і вище та інтегрованим середовищем розробки BlueJ або Eclipse. Для написання програми на Java може бути використаний будь-який текстовий редактор, наприклад, Notepad, WordPad в MS Windows і ін.

4. Правила техніки безпеки та охорони праці

Правила техніки безпеки при виконанні лабораторної роботи регламентуються «Правилами техніки безпеки при роботі в комп'ютерній лабораторії».

5. Порядок проведення лабораторної роботи

Для виконання роботи кожен студент повинен:

1. Відповісти на контрольні питання та пройти усне опитування за теоретичним матеріалом лабораторної роботи;
2. Пройти інструктаж за правилами охорони праці;
3. Запустити на комп'ютері інтегроване середовище розробки BlueJ;
4. Отримати варіант завдання у викладача;
5. Скласти алгоритм розв'язання задачі;
6. Записати код програми на комп'ютері;
7. Відкомпілювати програму та виправити всі помилки;
8. Запустити програму на виконання;
9. Отримати результати роботи програми і показати їх викладачу;
10. Підготувати і захистити звіт до лабораторної роботи.

** Варіант підвищеної важкості

6. Оформлення і захист звіту

Підготовлений до захисту звіт до лабораторної роботи повинен містити:

1. титульний лист, де вказані номер і назва лабораторної роботи, відомості про виконавця;
2. номер варіанта роботи та текст завдання;
3. відповіді на контрольні запитання до лабораторної роботи;
4. текст програми алгоритмічною мовою Java;
5. лістинг результатів виконання програми.

Лабораторна робота № 6

«Файлове введення/виведення.

Рядки. Робота з методами класу String і StringBuffer»

1. Мета роботи

Метою лабораторної роботи є отримання практичних навичок роботи з текстовими файлами та методами класів String, StringBuffer і StringTokenizer.

1. Завдання до лабораторної роботи

Використовуючи алгоритми, розглянуті на практичному занятті, скласти програму згідно отриманого варіанта.

Зауваження: у програмі ім'я текстового файлу повинно бути введено як параметр виклику методу main().

2.1 Методичні вказівки

■ Лабораторна робота спирається на знання й уміння, отримані при вивченні наступних тем лекційного курсу:

- Поняття конструктора. Операція new.
- Консольне введення – виведення.
- Робота з рядками. Приклади рішення завдань із використанням алгоритмів обробки рядків.
- Методи класа String.
- Методи класа StringBuffer і StringTokenizer.

Тому під час підготовки до лабораторної роботи рекомендується повторити зазначені розділи дисципліни.

■ **Конструктор** – це спеціальний метод, який викликається інтерпретатором Java завжди, коли оператором new створюється черговий екземпляр класу. Конструктор завжди носить ім'я класу.

Перерахуємо деякі особливості конструктора.

- Конструктор є в будь-якому класі. Навіть якщо ви його не написали, компілятор Java сам створить конструктор за умовчанням (default constructor). Проте, потрібно пам'ятати, що якщо ви вже визначили деякий конструктор або декілька (з аргументами або без), то компілятор більше не буде створювати конструктор за умовчанням.
- Конструктор виконується автоматично при створенні екземпляра класу, після розподілу пам'яті, але до початку використання об'єкту.

- Конструктор не повертає ніякого значення. Тому в його описі не пишуть навіть слово `void`, але можна задати один з трьох модифікаторів `public`, `protected` або `private`.
- Конструктор не є методом, він навіть не вважається членом класу.
- У класі може бути декілька конструкторів. Оскільки у них одне і те ж ім'я, яке співпадає з ім'ям класу, то вони повинні відрізнятися типом і/або кількістю параметрів.

■ Введення з консолі

Базові класи для введення/виведення розташовані в пакеті `java.io`. Тому для роботи з ними потрібно записати `import java.io.*`, а в програмі оголосити екземпляри відповідних класів.

```
InputStreamReader is = new InputStreamReader(System.in);
BufferedReader br = new BufferedReader(is);
```

Для введення даних типу `String` слід використовувати метод `readLine()` класу `BufferedReader`.

При введенні числових даних доведеться проводити перетворення типу `String` у відповідний тип. Для цього використовуються наступні методи:

```
Integer.valueOf(numberStr).intValue();
Long.valueOf(numberStr).longValue();
Float.valueOf(numberStr).floatValue();
Double.valueOf(numberStr).doubleValue();
```

Класи-оболонки `Integer`, `Short`, `Long`, `Float` і `Double`, розташовані в пакеті `java.lang`, містять також статичні методи

```
parseInt(String s);
parseShort(String s);
parseLong(String s);
parseFloat(String s);
parseDouble(String s);
```

які повертають числові еквіваленти рядка `s`.

Розглянемо приклад введення з клавіатури значення цілої змінної.

```
import java.io.*;
class Vvod{
    public static void main (String[] args) throws Exception{
        int a;
        BufferedReader br = new BufferedReader(new
            InputStreamReader (System.in));
        System.out.println("Введіть число a:");
        int a = Integer.parseInt(br.readLine());
        System.out.println(a); }}

```

■ Файлове введення/виведення

Розглянемо організацію файлового потоку за допомогою класів `FileReader` (для читання з файлу) і `FileWriter` (для запису у файл).

У конструкторах файлового потоку задається ім'я файлу у вигляді рядка типу String. Конструктори не тільки створюють об'єкт, але і відшуковують та відкривають файл. Наприклад:

```
String fileName = "c:\\Vixod.txt";  
FileReader fr = new FileReader(fileName);
```

Після відкриття вихідного потоку типу FileWriter вміст файлу, якщо він був не порожній, стирається. Для того, щоб можна було робити дозапис в кінець файлу, в класі передбачений конструктор з двома аргументами. Якщо другий аргумент рівний true, то відбувається дозапис в кінець файлу, якщо false, то файл заповнюється новою інформацією. Наприклад:

```
FileWriter fw = new FileWriter("c:\\Vixod.txt", true);
```

Пам'ятайте, що вміст файлу, відкритого на запис конструктором з одним аргументом, стирається.

Класи файлового введення/виведення не займаються буферизацією. Для цієї мети служать класи

```
BufferedReader br = new BufferedReader(fr);  
BufferedWriter bw = new BufferedWriter(fw);
```

Потоки fr і fw були визначені вище. Вони повинні приєднуватися до потоків введення/виведення.

Відразу після виконання конструктора можна читати файл методом br.readLine(); або записувати в нього методом bw.write();

Для того, щоб виведення в текстовий файл можна було організовувати за допомогою методів print(), println() і printf() слід використовувати клас PrintWriter pw=new PrintWriter(bw);

Після закінчення роботи з файлом потік слід закрити методом close().

Розглянемо приклад виведення символічних даних у текстовий файл.

```
import java.io.*;  
class Input{  
    public static void main(String[] args) {  
        String fileName = "c:\\file.txt";  
        // Рядок, який буде записаний у файл  
        String data = "Рядок для запису";  
        try{  
            FileWriter fw = new FileWriter(fileName);  
            BufferedWriter bw = new BufferedWriter(fw);  
            PrintWriter pw=new PrintWriter(bw);  
            // 3 рази запишемо рядок у файл  
            for(int i=3;--i>=0;) pw.println(data);  
            bw.close();  
  
            FileReader fr = new FileReader(fileName);  
            BufferedReader br = new BufferedReader(fr);
```



```

        String s = null;
        // Читаємо дані з файлу, відображаючи їх на екран
        while((s=br.readLine())!=null)
            System.out.println(s);
        br.close();
    }catch(Exception e){ }
}

```

Для зчитування даних з текстового файлу також можна використовувати клас **Scanner**. Зазвичай це роблять за допомогою логічних методів класу, наприклад, методу **hasNext()**, який повертає true, якщо в потоці є ще дані для зчитування, а також інших варіантів логічних методів для контролю доступності зчитування даних різних типів (**hasNextDouble()**, **hasNextInt()**, **hasNextBoolean()** та ін.)

Розглянемо приклад використання класу **Scanner** для розрахунку середнього арифметичного чисел, що зчитуються з текстового файлу.

```

import java.util.*;
import java.io.*;
class Zadanie{
    public static void main(String[] args)throws IOException {
        int count = 0;
        double sum =0.;
        //Запис в текстовий файл
        String name = "test.txt";
        FileWriter fout = new FileWriter(name);
        fout.write("2 3.4 5 6 7.4 9.1 10.5");
        fout.close();
        //Зміна локалі з руської на англійську
        Locale newLocale = new Locale ("en", "GB");
        Locale.setDefault(newLocale);
        FileReader fin = new FileReader(name);
        Scanner src = new Scanner (fin);
        //Зчитуємо дані та розраховуємо суму
        while(src.hasNext()) {
            if(src.hasNextDouble()) {
                sum+=src.nextDouble();
                count++;
            }
        }
        fin.close();
        System.out.printf("Average is %5.3f",sum/count);
    }
}

```

Можливо також використовувати клас **Scanner** для зчитування з файлу даних різних типів, навіть коли порядок їх слідування наперед невідомий. Треба просто перевіряти тип даних, що доступні для зчитування, перед тим як зчитати їх.

■ Клас String

Клас String є основним класом, призначеним для зберігання і обробки рядків символів.

Для створення екземплярів класу String може бути використаний один з наступних конструкторів:

```
String ()  
String (String str)  
String (StringBuffer strbuf)  
String (char[] arr)  
String (char[] arr, int first, int count)
```

Перший з них створює порожній рядок, другий і третій копіюють вміст об'єктів класів String і StringBuffer в створений об'єкт. Останні два конструктори дозволяють створити рядок на основі символьного масиву або його частини. Крім того, будь-яке об'єктне посилання типу String може бути проініціалізоване за допомогою привласнення йому рядкового літерала. Наприклад,

```
String filename = "data.txt";
```

■ Основні методи класу String

Метод	Опис
int length() char charAt(int index) char[] toCharArray()	Отримання довжини рядка Пошук символу Отримання рядка у вигляді символьного масиву
boolean equals(String str) boolean equalsIgnoreCase(String str) boolean startsWith(String prefix) boolean endsWith(String suffix)	Порівняння рядків на рівність Порівняння рядків без урахування регістру Перевірка, чи починається рядок із заданого підрядка Перевірка, чи закінчується рядок заданим підрядком
int indexOf(String subStr) int indexOf(String subStr, int fromIndex) int lastIndexOf(String subStr) int lastIndexOf(String subStr, int fromIndex)	Пошук першого входження підрядка в рядок з початку рядка Теж саме, але із заданої позиції Пошук останнього входження підрядка в рядок з початку рядка Теж саме, але із заданої позиції

Метод	Опис
String substring(int beginIndex, int endIndex) String substring(int beginIndex) String concat(String str) String toUpperCase() String toLowerCase() String trim() String replace(String target, String replacement)	Отримання підрядка (символ endIndex не входить до підрядка!) Отримання хвоста рядка Конкатенація рядків Перетворення рядка до верхнього/нижнього регістру Видалення провідних і завершуючих пробілів в рядку Заміна підрядка іншим рядком

■ Клас StringBuffer

Клас StringBuffer призначений для роботи з рядками, що модифікуються. Кожен об'єкт цього класу містить деякий буфер, що зберігає рядок. При змінах рядка розмір буфера може автоматично змінюватися.

Для створення екземплярів класу StringBuffer використовуються наступні конструктори.

```
StringBuffer ()
StringBuffer (int capacity)
StringBuffer (String str)
```

Перші два з них створюють об'єкт, що зберігає порожній рядок з початковим буфером довжини 16 в першому випадку і заданої довжини в другому випадку. Третій конструктор ініціалізує буфер заданим рядком. Розмір буфера в цьому випадку встановлюється рівним довжині рядка, збільшеного на 16.

■ Основні методи класу StringBuffer

Метод	Опис
int length() char charAt(int index) char setCharAt(int index)	Отримання довжини рядка Пошук символу Зміна символу
StringBuffer append(String str) StringBuffer append(int i) StringBuffer append(Object obj) ... StringBuffer insert(int index, String str) StringBuffer delete(int beginIndex, int endIndex) StringBuffer replace(int beginIndex, int endIndex, String str) StringBuffer reverse()	Додавання рядка до кінця (існують перевантажені версії для всіх примітивних типів і типу Object) Вставка рядка в задану позицію Видалення підрядка Заміна одного підрядка іншим Звернення рядка

■ Клас StringTokenizer

У пакет `java.util` входить простий клас `StringTokenizer`, що полегшує синтаксичний розбір рядків. У ньому три конструктори.

```
StringTokenizer (String str)
StringTokenizer (String str, String delimiters)
StringTokenizer (String str, String delimiters, boolean flag)
```

Перший конструктор створює об'єкт, готовий розбити рядок на слова, які розділені пробілами, символами табуляції `'\t'`, переводу рядка `'\n'` і повернення каретки `'\r'`. Роздільники не включаються до числа слів. Другий конструктор задає роздільники другим параметром `delimiters`, наприклад:

```
StringTokenizer ("Казнить, нельзя: пробелов-нет", " \t\n\r, :-");
```

Тут перший роздільник — пробіл. Потім йдуть символ табуляції, символ переводу рядка, символ повернення каретки, кома, двокрапка, дефіс. Порядок розташування роздільників в рядку `delimiters` не має значення. Роздільники не включаються до числа слів. Третій конструктор дозволяє включити роздільники до числа слів. Якщо параметр `flag` рівний `true`, то роздільники включаються до числа слів, якщо `false` — ні.

■ Основні методи класу StringTokenizer

Метод	Опис
<code>String nextToken ()</code>	Повертає у вигляді рядка наступне слово
<code>boolean hasMoreTokens ()</code>	Повертає <code>true</code> , якщо в рядку ще є слова, і <code>false</code> , якщо слів більше немає
<code>int countTokens ()</code>	Повертає число слів, що залишилися
<code>String nextToken(String newDelimiters)</code>	Дозволяє під час виконання програми міняти роздільники. Наступне слово буде виділено згідно нових роздільників <code>newDelimiters</code>

Приклад розбиття рядка на слова:

```
String s = "Рядок, який ми хочемо розібрати на слова";
StringTokenizer st = new StringTokenizer(s, " \t\n\r,.");
while(st.hasMoreTokens()){
// Отримуємо слово і виводимо на екран
System.out.println(st.nextToken()); }
```

2.2 Приклад програми

Прочитати рядки з текстового файлу і поміняти місцями перше і останнє слово в кожному рядку, новий текст записати в той же файл.

```
import java.io.*;
class Primer{
    public static void main(String[] args){String
        fileName=args[0];
        try{
            FileReader fr = new FileReader(fileName); BufferedReader
            br = new BufferedReader(fr); FileWriter fw = new
            FileWriter(fileName,true);;BufferedWriter bw = new
            BufferedWriter(fw); PrintWriter pw = new
            PrintWriter(bw);
            String s = null;int
            count = 0;
            pw.println();
            while((s=br.readLine())!=null){int
                nom1=s.indexOf(' ');
                String s1=s.substring(0,nom1+1);// Перше словоint
                nom2=s.lastIndexOf(' ');
                String s2=s.substring(nom2,s.length());//Останнє словоString
                s3=s.substring(nom1,nom2+1);
                s=s2+s3+s1; pw.println(s);}
            br.close(); pw.close();
        }catch(Exception e){ e.printStackTrace();}
    }}
```

Другий варіант програми з використанням методів класу StringTokenizer

```
import java.io.*;
import java.util.*;
class Primer{
    public static void main(String[] args){String
        fileName=args[0];
        try{
            FileReader fr = new FileReader(fileName);
            BufferedReader br = new BufferedReader(fr); FileWriter
            fw = new FileWriter(fileName,true);;BufferedWriter bw
            = new BufferedWriter(fw); PrintWriter pw = new
            PrintWriter(bw);

            String s = null,s1=null,s2=null,s3=null;int
            count = 0;
            pw.println();
            while((s=br.readLine())!=null){
                s2="";
                StringTokenizer st = new StringTokenizer(s, " \t\n\r,.");int n=0;
                while(st.hasMoreTokens()){
                    if (n==0) s1=st.nextToken(); //Перше словоelse {
                        if (st.countTokens()==1) s3=st.nextToken();//Останнє словоelse
                        s2+=st.nextToken();
                    }
                    n++;}
                s=s3+" "+s2+" "+s1;
                pw.println(s);}
            br.close(); pw.close();
        }catch(Exception e){ e.printStackTrace();}
    }}
```

2.3 Контрольні питання

1. За допомогою яких методів можна перетворити в рядок змінні примітивних типів і масив символів? Як перетворити з String в примітивний тип? Приведіть приклади.

2. Які методи класу String дозволяють провести заміну регістра букв?

3. У яких випадках використовується клас StringBuffer? Як можна встановити ємність буфера для об'єкту класу StringBuffer? Опишіть конструктори даного класу.

4. Дайте короткий опис конструкторам і методам класу StringTokenizer.

5. У чому відмінність між наступними фрагментами коду:

```
1. String s1 = "abc";  
String s2 = new String("abc");  
boolean result = (s1==s2);
```

```
2. String s1 = new String("abc");  
String s2 = new String("abc");  
boolean result = (s1.equals(s2));
```

6. Як в Java можна організувати файлові потоки введення-виведення?

2.4 Варіанти завдань

7. Прочитати текст Java-програми і всі слова "public" в оголошенні атрибутів класу замінити на "private". Новий текст записати в той же файл.

8. Прочитати текст Java-програми і записати в інший файл у зворотному порядку символи кожного рядка.

9. Прочитати текст Java-програми і в кожному слові коротше за п'ять символів всі символи замінити знаком *. Результат записати в інший файл.

10. Напишіть програму для проведення пошуку і заміни слів в тексті. Вона повинна приймати три параметри виклику методу main(): ім'я файлу, рядок для пошуку і рядок для заміни. В результаті її роботи створюється файл з ім'ям res.txt, що містить результати заміни.

11. Прочитати текст Java-програми і видалити з нього всі слова що починаються з символу 'р'. Новий текст виведіть в інший файл.

12. Прочитати текст Java-програми і в кожному слові довше за два символи всі рядкові символи замінити прописними. Результат записати в інший файл.

13. З файлу видалити всі слова, що містять від трьох до п'яти символів, але при цьому з кожного рядка треба видалити тільки максимальну парну кількість таких слів.

14. У файлі, що містить в кожному окремому рядку прізвища студентів і їх оцінки, записати великими буквами прізвища тих студентів, які мають середній бал більше "4". Приклад файлу:

Петренко 5 4 5 3 3

Шевченко 5 5 4 3

15. Прочитати текст Java-програми і визначити, скільки в ньому оголошених класів. Видалити в оголошенні класів модифікатор "public". Результат вивести в той же файл.

16. З тексту Java-програми видалити коментарі виду `//` і `/* */`. Новий текст записати в той же файл.

17. Створити і заповнити файл випадковими цілими числами. Відсортувати вміст файлу за збільшенням і результати вивести в цей же файл.

18. Прочитати текст Java-програми і видалити з нього всі "зайві" пробіли, залишивши тільки значущі.

19. Прочитати текст Java-програми і у вихідний файл записати всі статично виділені в програмі змінні типу `int` і `double` в наступному форматі:

Змінні типу `int`: a,b,c.

Змінні типу `double`: x,y,z.

20. Прочитати текст Java-програми і у вихідний файл записати всі цикли разом з їх вмістом (у відповідності з синтаксисом циклу `for`). Цикли розділяти порожнім рядком. **Вказівка:** врахувати можливість існування вкладених циклів.

21. Прочитати текст Java-програми і у кожному рядку всі слова, довжина яких перевищує середню довжину слів в рядку, скоротити до середньої довжини. Результат записати в інший файл.

3. Прилади, устаткування та інструменти

Для виконання лабораторної роботи використовується ПЕОМ з установленим пакетом Sun Microsystems JDK 1.5 і вище та інтегрованим середовищем розробки BlueJ або Eclipse. Для написання програми на Java може бути використаний будь-який текстовий редактор, наприклад, Notepad, WordPad в MS Windows і ін.

4. Правила техніки безпеки та охорони праці

Правила техніки безпеки при виконанні лабораторної роботи регламентуються «Правилами техніки безпеки при роботі в комп'ютерній лабораторії».

5. Порядок проведення лабораторної роботи

Для виконання роботи кожен студент повинен:

1. Відповісти на контрольні питання та пройти усне опитування за теоретичним матеріалом лабораторної роботи;
2. Пройти інструктаж за правилами охорони праці;
3. Запустити на комп'ютері інтегроване середовище розробки BlueJ;
4. Отримати варіант завдання у викладача;
5. Скласти алгоритм розв'язання задачі;
6. Записати код програми на комп'ютері;
7. Відкомпілювати програму та виправити всі помилки;
8. Запустити програму на виконання;
9. Отримати результати роботи програми і показати їх викладачу;
10. Підготувати і захистити звіт до лабораторної роботи.

6. Оформлення і захист звіту

Підготовлений до захисту звіт до лабораторної роботи повинен містити:

1. титульний лист, де вказані номер і назва лабораторної роботи, відомості про виконавця;
2. номер варіанта роботи та текст завдання;
3. відповіді на контрольні запитання до лабораторної роботи;
4. текст програми алгоритмічною мовою Java;
5. лістинг результатів виконання програми.

Додаток.

Етапи виконання Java-програми та особливості роботи із середовищем розробки BlueJ

Установка BlueJ

BlueJ є середовищем розробки мовою Java, що було створено командою розроблювачів з Університету Monash, Мельбурн, Австралія спеціально для навчання на початковому рівні. Система вільна для поширення. Додаткова інформація про BlueJ доступна в Internet за адресою <http://www.bluej.monash.edu>.

BlueJ поширюється у вигляді архіву класів Java, у «jar» форматі. Обов'язковою вимогою до установки є наявність пакета Sun Microsystems JDK1.3 і вище. Дистрибутивний файл із BlueJ має ім'я *bluej-xxx.jar*, де xxx

– номер версії. При установці середовища розробки в Windows слід виконати подвійне клацання мишею на файлі *bluej-xxx.jar*. Далі буде створений пакетний файл *bluej* у каталозі BlueJ, після чого можна запустити BlueJ просто подвійним клацанням з GUI інтерфейсу.

Розглянемо найбільш важливі на першому етапі вивчення редактора особливості створення, редагування і виконання програм в BlueJ.

Створення проекту

Для створення нового проекту з меню слід вибрати Project – New... Відкриється діалогове вікно, у якому можна вказати ім'я та розташування нового проекту. Після натискання ОК, буде створена директорія з відповідним ім'ям і відкриється порожнє вікно нового проекту подібно тому, що представлене на рис.1.

Створення класу

Будь-яка програма мовою Java оформляється у вигляді класу. Створити свій власний клас можна, клацнувши на кнопку *New Class* на панелі інструментів у вікні проекту. BlueJ запросить ім'я класу, яке повинне бути припустимим в Java ідентифікатором. Після створення класу, він буде представлений окремою піктограмою у вікні проекту. Двічі клацнувши на даній піктограмі можна відкрити редактор з новим класом, у якому частина класу вже створена для полегшення початку роботи (рис.2). Далі можна приступати до написання коду програми.

Збільшити розмір шрифту у вікні редактора класу можна нажавши в головному меню редактора кнопку *Options-Preferences...* У діалоговому вікні *Preferences* у вкладці *Editor* можна задати розмір шрифту.

Щоб видалити клас із проекту, потрібно виділити його й потім вибрати *Remove Class* з меню *Edit* або вибрати *Remove* з контекстного меню класу.

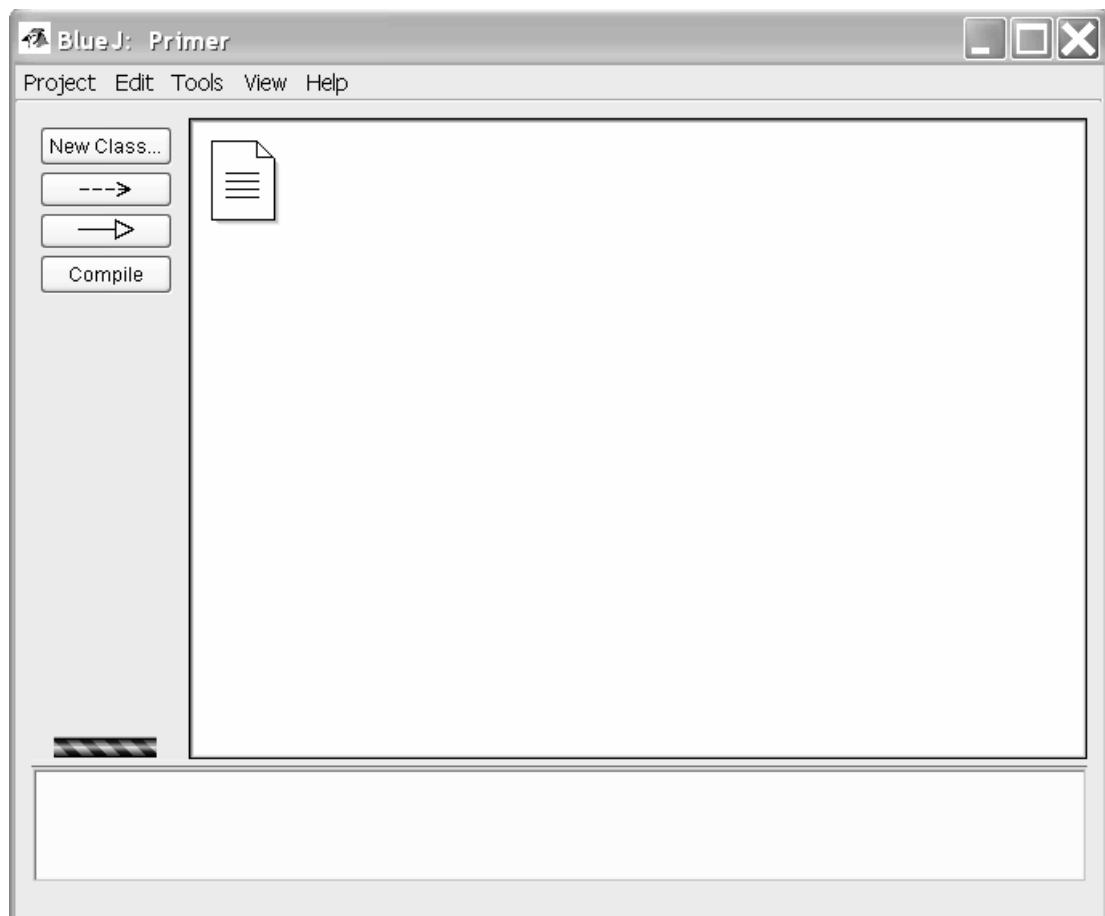


Рисунок 1 – Вид головного вікна BlueJ

Компіляція

Перед запуском програми на виконання її треба перевірити на наявність синтаксичних помилок, тобто відкомпілювати. Зверніть увагу, що на піктограмі класу `my_class` рис.2 є заштрихована область. Штрихування позначає клас, що не був відкомпільований після останніх змін.

На панелі інструментів вікна редактора, розташовуються кнопки для функцій, що часто використовуються. Одна з них *Compile*. Ця кнопка дозволяє відкомпілювати клас прямо з вікна редактора. Клацнувши на кнопку *Compile*, за умови, що в коді програми немає помилок, унизу вікна редактора з'явиться повідомлення, про те, що компіляція виконана (рис.3). Якщо є синтаксичні помилки, рядок, що містить її, буде підсвічений й повідомлення про помилку з'явиться в інформаційній області.

Панель інструментів вікна проекту також має кнопку *Compile*. Однак натискання на неї приводить до компіляції всього проекту. Відкомпілювати ж окремий клас можна з вікна проекту викликавши на відповідній піктограмі класу контекстне меню і пункт *Compile*.

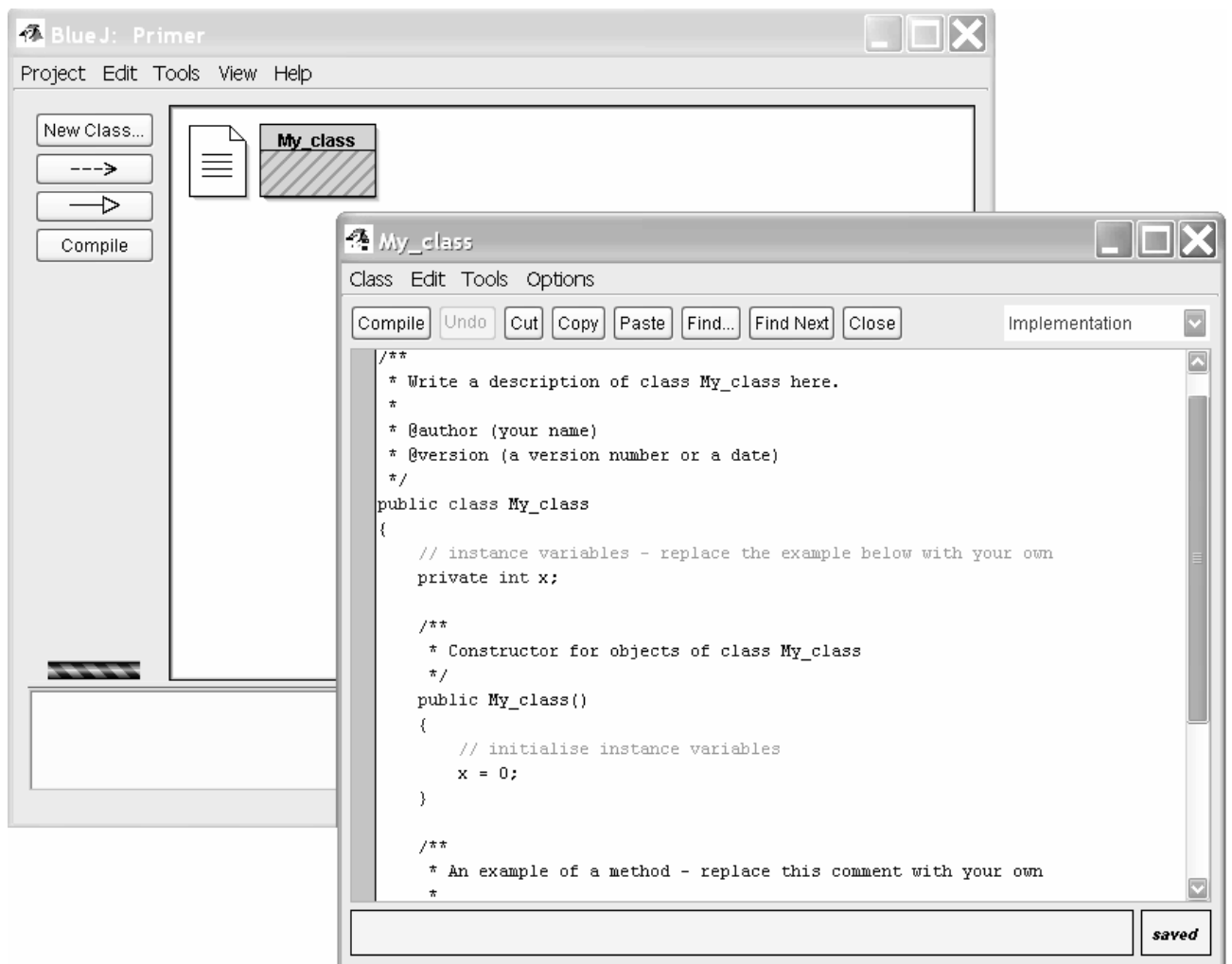


Рисунок 2 – Вид вікна редактора класу

Пам'ятайте, що немає необхідності примусово зберігати вихідний код. Він зберігається автоматично при закритті редактора та перед компіляцією.

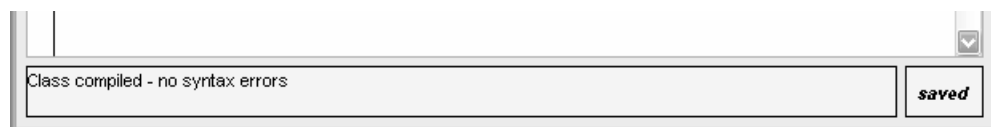


Рисунок 3 – Вигляд повідомлення про успішну компіляцію програми

Допомога при помилках компіляції

Повідомлення про помилки компіляції висвітлюються в інформаційній області. У правому нижньому куті з'являється знак питання, клацнувши на який, можна одержати додаткову інформацію про тип даної помилки (рис.4). На жаль, додаткова інформація доступна не для всіх помилок, але багато з них описані.

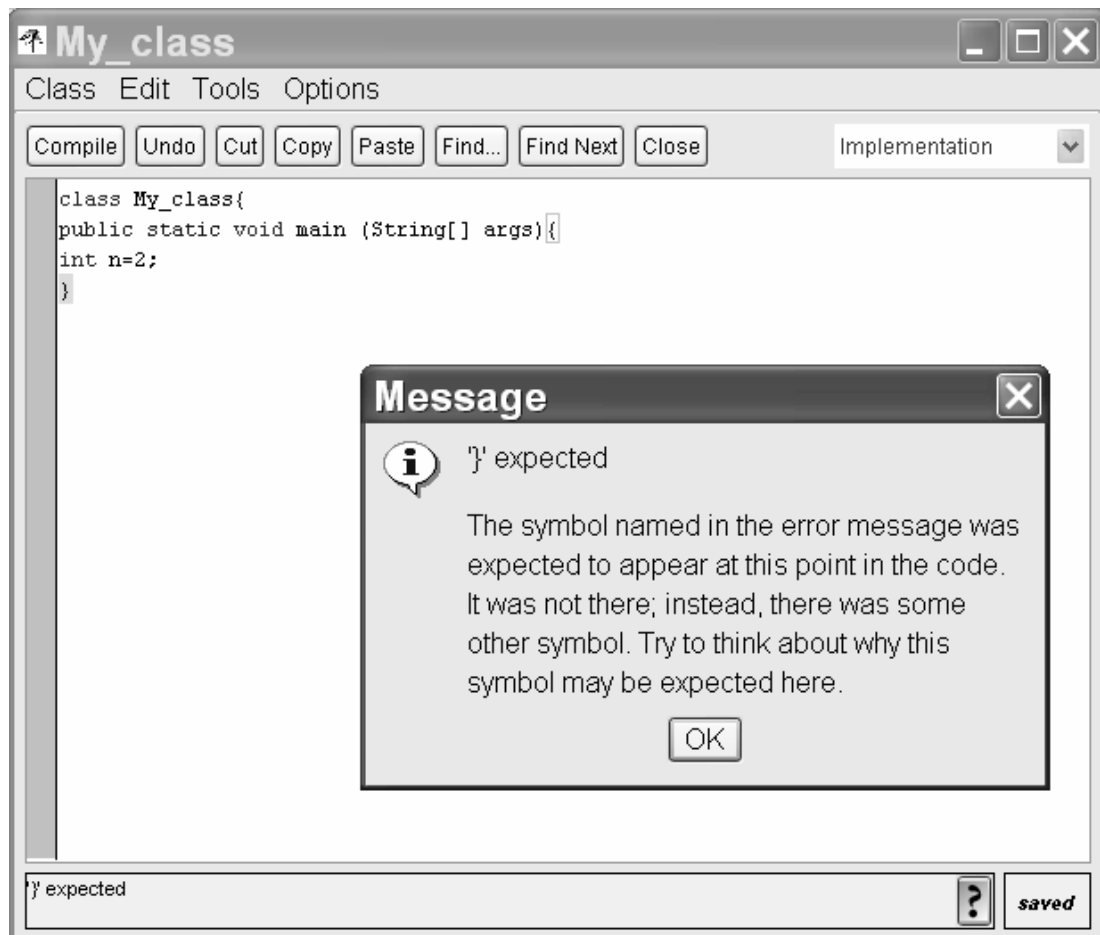


Рисунок 4 – Помилка компіляції і кнопка допомоги

Запуск програми на виконання і виклик методу main()

Запуск програми на виконання звичайно починається з виклику методу main(), який в BlueJ може бути викликаний безпосередньо з контекстного меню класу. Якщо клацнути на класі правою кнопкою миші, то метод main() з'явиться в контекстному меню (мал.5). Стандартний метод main() припускає передачу в якості параметра масиву рядків. Можна передати масив рядків, використовуючи стандартну нотацію Java для масивів констант. Наприклад: {"one", "two", "three"} обов'язково, включаючи, лапки.

Додавання існуючих класів у проект

Часто виникає необхідність включати в проект раніше створені класи, які використовуються десь в іншому місці. Для того, щоб можна було їх викликати,

треба додати даний клас у проект. Це можна зробити використовуючи меню – Edit – Add Class from File. ... Необхідний Java файл буде знайдений і включений у проект.

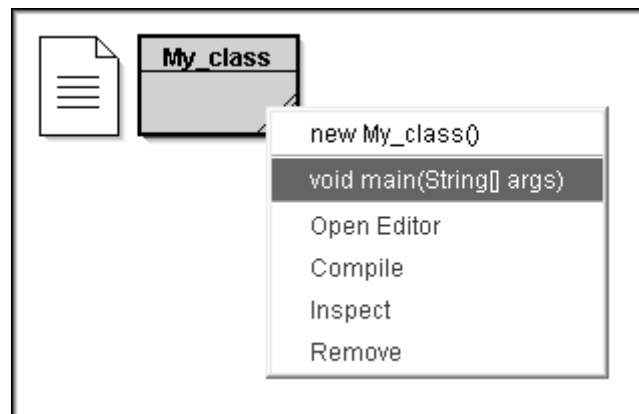


Рисунок 5 – Виклик методу main()

Після імпортування класу в проект, його копія буде створена в каталозі цього проекту. Альтернативний метод – скопіювати вихідний код класу в каталог без участі BlueJ.

Переривання виконання програми

Іноді, виникає необхідність примусово перервати виконання програми. Це особливо актуально в ситуації її "зависання". Стан виконання програми відображається у світному барі, розташованому внизу вікна редактора проекту. Перервати виконання програми можна викликавши контекстне меню на барі стану та вибрав пункт *Reset* (мал.6) або набравши комбінацію *Ctrl+Shift+R*.



Рисунок 6 – Переривання виконання програми

Налагодження програми

Розглянемо найбільш важливі аспекти налагодження програми в BlueJ. Функціональність відладчика зведена до трьох завдань:

- призначення крапок переривання;

- покроковий прохід по коду;
- перевірка значень змінних.

Установка крапок переривання

Установка крапок переривання дозволяє перервати виконання в певній крапці коду. Після того, як переривання здійсниться, стає можливим контролювати стан змінних. Це часто допомагає зрозуміти, що відбувається в коді. У редакторі, ліворуч від тексту, розташовується зона для установки крапок переривання (рис.7). Їх можна встановити просто клацнувши в цій області мишею. Невелика піктограма "stop" відзначить крапку переривання.

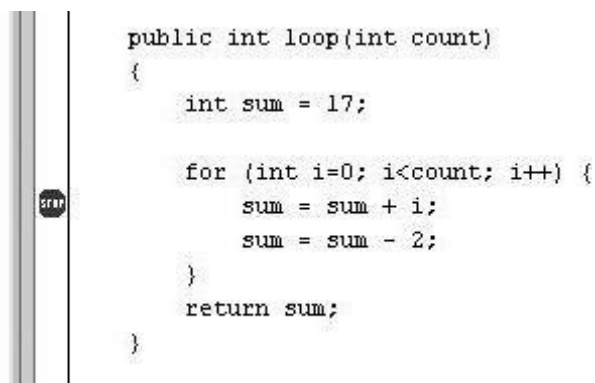


Рисунок 7 – Крапка переривання

Як тільки виконання коду дійде до цього рядка – відбудеться припинення виконання програми. Як тільки крапка "stop" буде досягнута, відкриється редактор і вікно відладчика *Debugger*, як на рис.8.

Підсвічений рядок указує на рядок, що буде виконаний в наступний момент.

Покрокове виконання коду

Після припинення програми можна здійснити покрокове виконання коду. Для того, щоб виконати наступний рядок програми, треба клацнути на кнопку Step у вікні відладчика. Підсвічений рядок зрушиться на крок униз. Щоразу при натисканні кнопки Step, один рядок коду буде виконуватися, а виконання програми припинятися. Значення змінних у вікні відладчика теж змінюються, у такий спосіб можна просуватися крок за кроком і спостерігати, що при цьому відбувається зі змінними. Якщо потрібно перервати покрокове виконання програми, то слід клацнути на крапку переривання знову, щоб усунути її. Для продовження роботи додатка у звичайному режимі треба клацнути на кнопку *Continue*.

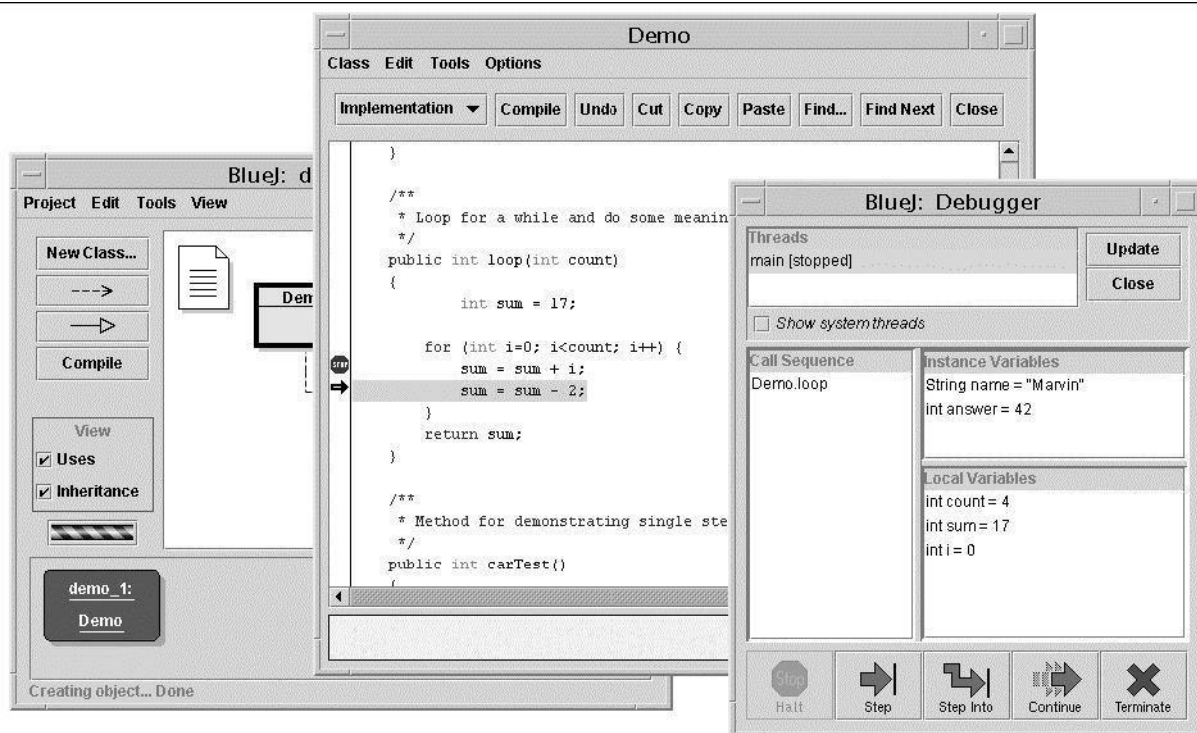


Рисунок 8 – Вікно відладчика

У випадку коли програма працює занадто довго (перебуває в режимі зависання), можна відкрити вікно відладчика з меню *View* – *ShowDebugger* і клацнути на кнопку *Halt*. Виконання перерветься, начебто вставили крапку переривання. Далі можна зробити декілька кроків по програмі і переконатися, що все в порядку. Можна використовувати *Continue* і *Halt* стільки разів, скільки буде необхідно. Якщо немає необхідності продовжувати роботу додатка, то можна клацнути на кнопку *Terminate*, і виконання припиниться. *Terminate* не можна використовувати часто, тому що ця операція залишає об'єкти в невизначеному стані, тому цей засіб краще застосовувати для кінцевої зупинки додатка.

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт
з навчальної дисципліни “Алгоритмізація і
програмування” для студентів I року навчання
денної та заочної форми навчання
спеціальності 122 «Комп’ютерні науки»

Укладачі: доц. Кузніченко С.Д.
асист. Клепатська В.В.

Підп. до друку

Формат 60x84/16 Папір офс.

Умовн. а.а.

Тираж

Замовл.

Одеський державний екологічний університет,
65016, м. Одеса, вул. Львівська, 15
