

ПРАКТИЧНЕ ЗАНЯТТЯ № 4

Тема: Робота із масивами даних на прикладі скалярного добутку векторів

Мета: Навчитись застосовувати статичний та звичайний метод класу для роботи із масивами.

Обладнання: комп'ютери Pentium, Celeron.

Програмне забезпечення: MS Windows XP, пакет Java SDK.

У наведеному нижче прикладі клас DoubleVector (файл DoubleVector.java) призначений для роботи з векторами. Реалізовано тільки скалярне множення векторів, але при необхідності цей клас можна розширити іншими операціями над векторами.

```
public class DoubleVector {
    private double[] vector = null;
    public DoubleVector(double[] vector){
        this.vector = vector;
    }
    // Скалярний добуток векторів
    public double mult(DoubleVector anotherVector){
        double s = 0;
        for ( int i = 0; i < vector.length; i++ ){
            s += vector[i] * anotherVector.vector[i];
        }
        return s;
    }
    public static double mult(DoubleVector a, DoubleVector b){
        return a.mult(b);
    }
    public static void main(String[] args){
        double[] a = {1, 2, 3, 4};
        double[] b = {1, 1, 1, 1};
        double[] c = {2, 2, 2, 2};
        DoubleVector v1 = new DoubleVector(a);
        DoubleVector v2 = new DoubleVector(b);
        DoubleVector v3 = new DoubleVector(c);
        System.out.println("v 1*v2=" + v1.mult(v2));
        System.out.println("v 1*v2=" + DoubleVector.mult(v1, v2));
        System.out.println("v 1*v3=" + v1.mult(v3));
    }
}
```

У класі DoubleVector є поле - масив vector, конструктор для побудови вектора з масиву, два методи mult(..), один з яких статичний, а також метод main(..) для перевірки працездатності класу.

Слід звернути увагу на методи mult(..). Перший з них призначений для множення цього вектора на інший вектор. Він використовується в методі main(..) в рядках
System.out.println("v 1*v2=" + v1.mult(v2)); і System.out.println("v 1*v3=" + v1.mult(v3));
Другий для множення двох векторів. Він використовується в рядку
System.out.println("v 1*v2=" + DoubleVector.mult(v1, v2));

Статичний метод mult(..) реалізований на базі звичайного методу, шляхом повернення "a.mult(b)". Альтернативою є реалізація алгоритму множення в статичному методі і виклик його із звичайного методу. Це виглядало б так.

```
public double mult(DoubleVector anotherVector){
    return mult(this, anotherVector);
}
public static double mult(DoubleVector a, DoubleVector b){
    double s = 0;
    for ( int i = 0; i < a.vector.length; i++ ){
        s += a.vector[i] * b.vector[i];
    }
    return s;
}
```