­­МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

­­­

Лабораторна робота №3

з курсу «Об’єктно-орієнтоване програмування»

для студентів базового напрямку 6.08.04 "Комп’ютерні науки"

(заочна форма навчання)

Варіант 14

Виконав студент гр. КНз-2

Чалий Михайло

­­

Львів 2014

# Керуючі структури та масиви у Java

## Мета роботи

Одержати навики побудови алгоритмів лінійних обчислювальних процесів, навчитися складати алгоритми та програмувати процеси з розгалуженням, циклами, ітераційними циклами за допомогою мови Java. Одержати навички роботи із масивами.

## Теоретичні відомості

Будь-який алгоритм, призначений для виконання на комп’ютері, можна розробити, використовуючи лише лінійні обчислення, розгалуження і цикли.

Записати його можна у різних формах: у вигляді блок-схеми, на псевдокоді, на звичайній мові тощо.

Будь-яка мова програмування повинна мати свої методи запису алгоритмів. Вони називаються операторами (statements) мови. Мінімальній набір операторів повинен містити оператор для запису лінійних обчислень, умовний оператор для запису розгалужень і оператор циклу.

Звичайно склад операторів мови програмування є ширшим: для зручності записи алгоритмів у мову включаються декілька операторів циклу, оператор варіанта, оператори переходу, оператори описування об’єктів.

Набір операторів мови Java включає:

* оператори описування змінних та інших об’єктів;
* оператори-вирази;
* оператори присвоювання;
* умовний оператор if;
* три оператори циклу while, do-while, for;
* оператор варіанта switch;
* Оператори переходу break, continue і return;
* блок {};
* порожній оператор —крапка з комою.

Тут наведений не весь набір операторів Java, він буде доповнюватися у процесі виконання наступних лабораторних робіт.

У мові Java відсутній оператор goto. Будь-який оператор закінчується крапкою з комою. Можна поставити крапку з комою у кінці будь-якого виразу, і він стане оператором (expression statement). Але це має зміст тільки для операцій присвоювання, інкрементування, декрементування і виклику методів. У решті випадків це не має змісту, тому що обчислене значення виразу буде втрачено.

Лінійне виконання алгоритму забезпечується послідовним записом операторів. Перехід із рядка на рядок у вихідному тексті не має жодного значення для компілятора, він здійснюється тільки для наочності і читабельності тексту.

## Завдання

Задано натуральне число n і масив дійсних чисел a[1..n]. Визначити добуток непарних елементів, розташованих після останнього від’ємного елемента.

## Реалізація

Лістінг 1. Algorithm.java

**import** java.util.Arrays;

**import** java.util.Random;

**public** **class** Algorithm {

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

// Задано натуральне число n і масив дійсних чисел a[1..n].

// Визначити добуток непарних елементів, розташованих

// після останнього від’ємного елемента.

**int** n = 20;

**int**[] a = **new** **int**[n];

// Init

Random random = **new** Random();

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

a[i] = (random.nextInt(200) - 100); // From -100 to 100

}

System.*out*.printf("Array: %s%n", Arrays.*toString*(a));

// Calculate

**int** product = 1;

**boolean** negativeFound = **false**;

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

**if** (negativeFound){

**if** (i%2 != 0){

product \*= a[i];

}

} **else** {

**if** (a[i] < 0){

negativeFound = **true**;

}

}

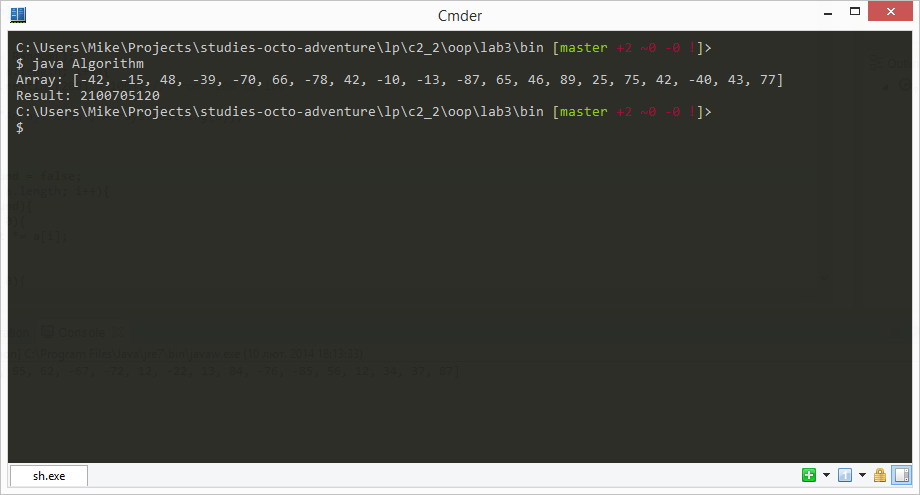
}

System.*out*.printf("Result: %d%n", product);

}

}

## Результат



## Висновки

Одержав навики побудови алгоритмів лінійних обчислювальних процесів, навчився складати алгоритми та програмувати процеси з розгалуженням, циклами, ітераційними циклами за допомогою мови Java. Одержав навички роботи із масивами.