Министерство образования Республики Беларусь

Учреждения образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчет по лабораторной работе №1

По теме «Определение модели языка. Выбор инструментальной языковой среды.»

Выполнил:

студент гр. 053501

Шебеко Ю.А.

Проверил:

Гриценко Н. Ю.

Минск 2023

Содержание

1. Цель работы

Необходимо определить подмножество языка программирования (типы констант, операторов и функций). В подмножестве как минимум должны быть включены:

– числовые и текстовые константы;

– 3-4 типа переменных;

– операторы цикла (do…while, for);

– условные операторы (if…else, case).

Определение инструментальной языковой среды, то есть языка программирования и операционной системы для разработки включает:

– языки программирования с указанием версии, на котором ведется разработка.

– компьютер (PC/ Macintosh).

В отчете по лабораторной работе дается полное определение подмножества языка программирования, тексты двух трех программ, включающих все элементы этого подмножества. Приводится подробное описание инструментальной языковой среды.

2. Подмножество языка программирования

В качестве подмножества языка программирования выбран язык Java.

Java – язык программирования общего назначения. Относится к объектно-ориентированным языкам программирования, к языкам с сильной типизацией. Создатели реализовали принцип WORA: write once, run anywhere или «пиши один раз, запускай везде». Это значит, что написанное на Java приложение можно запустить на любой платформе, если на ней установлена среда исполнения Java (JRE, Java Runtime Environment). Синтаксис языка Java похож на синтаксис других си-подобных языков.

2.1 Числовые и строковые константы

При объявлении констант в Java используют ключевое слово **final –** оно показывает, что литерал не должен меняться.

Пример констант:

1. “”, “String” (строковые литералы)
   public static final String Constant = “Константа”;
2. числовая:
   2.1. Целочисленные (-1, 0, 1)
   2.2. С плавающей запятой (3.5, -2.7)
   public static final int (float/double) Number = 5;
3. символьные:

‘A’, ‘1’ (Unicode литералы)
   public static final char Symbol = ‘A’;

4. true, false (логические литералы)

2.2 Типы переменных

Java строго типизированный язык. Переменные в Java бывают двух видов: примитивные (для маленьких данных) и ссылочные (для более сложных). В состав примитивных видов входят 4 подвида и восемь типов данных:

1) целые числа (byte, short, int, long);
2) числа с плавающей точкой (float, double);
3) логический (boolean);
4) символьный (char).

2.3 Операторы цикла

1) цикл while

Выполняет тело цикла до тех пор, пока условие цикла истинно.

```
int j = 6;
while (j > 0){

    System.out.println(j);
    j--;
}
```

2) цикл do … while

В отличие от while тело цикла выполнится как минимум один раз.

```
int j = 7;
do{
    System.out.println(j);
    j--;
}
while (j > 0);
```

3) цикл for

Это цикл, в котором переменная – это счетчик итераций цикла с определенным шагом, изменяет свое значение до заданного конечного значения.

```
for(int i = 1; i < 9; i++){
     System.out.printlf("Квадрат числа %d равен %d
\n", i, i * i);
}
```

4) цикл for each

Это разновидность цикла for для итерации коллекции и массивов.

```
List<String> names = new ArrayList<>();

names.add("Snoopy");
names.add("Charlie");
names.add("Linus");
names.add("Shroeder");
names.add("Woodstock");

for(String name : names){
    System.out.println(name);
}
```

5) continue

Начинает следующий проход цикла, не исполняя оставшееся тело цикла

```
for(int i = 0; i < 10; i++){
 if (i == 5)
     continue;
 System.out.println(i);
```

```
}
```

6) break

Прерывает исполнение цикла

```
    for(int i = 0; i < 10; i++){
if (i == 5)
    break;
System.out.println(i);
}
```

2.4 Условные операторы

1) if

```
int num1 = 6;
int num2 = 4;
if(num1>num2){
    System.out.println("Первое число больше второго");
}
```

После ключевого слова if ставится условие. И если это условие выполняется, то срабатывает код, который помещен в далее в блоке if после фигурных скобок. В качестве условий выступает операция сравнения двух чисел.

2) if-else

```
int num1 = 6;
int num2 = 4;
if(num1>num2){
    System.out.println("Первое число больше второго");
}
else{
    System.out.println("Первое число меньше второго");
}
```

3) switch-case

Конструкция switch/case аналогична конструкции if/else, так как позволяет обработать сразу несколько условий:

```
int num = 8;
switch(num){

    case 1:
        System.out.println("число равно 1");
        break;
    case 8:
        System.out.println("число равно 8");
        num++;
        break;
    case 9:
        System.out.println("число равно 9");
        break;
    default:
        System.out.println("число не равно 1, 8, 9");
}
```

После ключевого слова switch в скобках идет сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после операторов case. И если совпадение найдено, то будет выполняет соответствующий блок case.

В конце блока case ставится оператор break, чтобы избежать выполнения других блоков.

3. Инструментальная языковая среда

В качестве языковой среды выбран язык программирования Python.

Разработка основана на работе с операционной системой Windows на PC.

Python— высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python является мультипарадигменным языком программирования, поддерживающим императивное, процедурное, структурное, объектно-ориентированное программирование, метапрограммирование и функциональное программирование. Задачи обобщённого программирования решаются за счёт динамической типизации. Аспектно-ориентированное программирование частично поддерживается через декораторы, более полноценная поддержка обеспечивается дополнительными фреймворками. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений с глобальной блокировкой интерпретатора (GIL), высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Приложение. Текст программ

1. Числа Фибоначчи. Вывести первые 11 членов последовательности Фибоначчи

```
public class Main {

    public static void main(String[] args) {
        //Task1
        int n0 = 1;
        int n1 = 1;
        int n2;
        System.out.println(n0+" "+n1+" ");
        for(int i=3; i<=11; i++) {
            n2=n0+n1;
            System.out.println(n2+" ");
            n0=n1;
            n1=n2;
        }
        System.out.println();
    }
}
```

2. Максимальное, минимальное и среднее значение

```
int n = 100;
double[] array = new double[n];
for (int i = 0; i < array.length; i++) {
    array[i] = Math.random();
}

double max = array[0];
double min = array[0];
double avg = 0;
for (int i = 0; i < array.length; i++) {
    if(max < array[i])
        max = array[i];
    if(min > array[i])
        min = array[i];
    avg += array[i]/array.length;
}

System.out.println("max = " + max);
System.out.println("min = " + min);
System.out.println("avg = " + avg);
```

3. Сортировка пузырьком

```
int [] mas = {11, 3, 14, 16, 7};

boolean isSorted = false;
int buf;
while(!isSorted) {
    isSorted = true;
    for (int i = 0; i < mas.length-1; i++) {
```

```
            if(mas[i] > mas[i+1]){
                isSorted = false;

                buf = mas[i];
                mas[i] = mas[i+1];
                mas[i+1] = buf;
            }
        }
    }
    System.out.println(Arrays.toString(mas));
```