Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ
И РАДИОЭЛЕКТРОНИКИ

КАФЕДРА ИНФОРМАТИКИ

Отчёт по лабораторной работе №1

По теме “Определение модели языка. Выбор инструментальной языковой среды.”

Выполнил:
студент гр. 555505
Сидоров С.С.

Проверил:
Ст. преподаватель кафедры информатики Шиманский В. В.

Минск 2019

Содержание

1. Цель работы

Необходимо определить подмножество языка программирования (типы констант, переменных, операторов и функций). В подмножество 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, т.е. языка программирования и операционной системы для разработки включает:

- язык программирования с указанием версии, на котором ведётся разработка (напр. Python 3.7);

- операционная система (Windows, Linux и т.д.), в которой выполняется разработка;

- компьютер (PC / Macintosh).

В отчете по лабораторной работе дается полное определение подмножества языка программирования, тексты 2-3-х программ, включающих все элементы этого подмножества. Приводится подробное описание инструментальной языковой среды.

2. Подмножество языка программирования

В качестве подмножества языка программирования выбран язык Python.

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.

2.1 Числовые и строковые константы

- -3, -2, -1, 0, 1, 2, 3 (int).
- 2+3j, 0+5j, 2j, -3-5j (complex литералы)
- 3.5, -2.7 (float литералы)
- "", "hello" (str литералы)
- u"", u"hello" (unicode литералы)

2.2 Типы переменных

Python поддерживает динамическую типизацию, то есть тип переменной определяется только во время исполнения. Поэтому вместо «присваивания значения переменной» лучше говорить о «связывании значения с некоторым именем». В Python имеются встроенные типы: булевый, строка, Unicode-строка, целое число произвольной точности, число с плавающей запятой, комплексное число и некоторые другие. Из коллекций в Python встроены: список, кортеж (*неизменяемый список*), словарь, множество и другие. Все значения являются объектами, в том числе функции, методы, модули, классы.

- Операторы цикла
  - while - выполняет тело цикла до тех пор, пока условие цикла истинно.

```
>>> i = 5
>>> while i < 15:
...     print(i)
...     i = i + 2
5
7
9
11
13
```

- for - выполняет тело цикла, итерируясь по объекту (к примеру, строке или списку)

```
>>> for i in 'hello world':
...     print(i * 2, end='')
...
hheellllloo  wwoorrlldd
```

- continue - начинает следующий проход цикла, не исполняя оставшееся тело цикла

```
>>> for i in 'hello world':
...     if i == 'o':
...         continue
...     print(i * 2, end='')
...
hheellll  wwrrlldd
```

- break - прерывает исполнение цикла

```
>>> for i in 'hello world':
...     if i == 'o':
...         break
...     print(i * 2, end='')
...
hheellll
```

2.3 Условные операторы

- Оператор *if*

```
if выражение:
    инструкция_1
    инструкция_2
```

```
    ...
    инструкция_n
```

Пример:

```
a = 3
if a > 1:
   print("hello 3")
```

- Конструкция if - else

```
if выражение:
   инструкция_1
   инструкция_2
   ...
   инструкция_n
else:
   инструкция_a
   инструкция_b
   ...
   инструкция_x
```

Пример:

```
a = 3
if a > 2:
   print("H")
else:
   print("L")
```

3. Инструментальная языковая среда

В качестве языковой среды выбран язык программирования Rust. Разработка основана на работе с операционной системой Linux на PC.

Rust — мультипарадигмальный компилируемый язык программирования общего назначения, спонсируемый Mozilla Research, сочетающий парадигмы функционального и процедурного программирования с объектной системой, основанной на типажах, и с управлением памятью через понятие «владения» (систему аффинных типов, позволяющую обходиться без сборки мусора). Объектно-ориентированное программирование как таковое языком не поддерживается, но язык позволяет реализовать большинство понятий ООП при помощи других абстракций.

Ключевые особенности языка: безопасность, скорость и параллелизм. Rust пригоден для системного программирования, в частности, он рассматривается как перспективный язык для разработки ядер операционных систем. Rust сопоставим по скорости и возможностям с C++, однако дает большую безопасность при работе с памятью, что обеспечивается механизмами ограничения. Rust также направлен на достижение «абстракции с нулевой стоимостью».

Приложение. Текст программ

1. Нахождения факториала числа, введенного пользователем

```
num = int(input("Enter a number: "))

factorial = 1

if num < 0:
   print("Sorry, factorial does not exist for negative numbers")
elif num == 0:
   print("The factorial of 0 is 1")
else:
   for i in range(1,num + 1):
       factorial = factorial*i
   print("The factorial of", num, "is", factorial)
```

2. Быстрая сортировка

```
def quickSort(alist):
   quickSortHelper(alist,0,len(alist)-1)

def quickSortHelper(alist,first,last):
   if first<last:

       splitpoint = partition(alist,first,last)

       quickSortHelper(alist,first,splitpoint-1)
       quickSortHelper(alist,splitpoint+1,last)


def partition(alist,first,last):
   pivotvalue = alist[first]

   leftmark = first+1
   rightmark = last

   done = False
   while not done:

       while leftmark <= rightmark and alist[leftmark] <= pivotvalue:
           leftmark = leftmark + 1

       while alist[rightmark] >= pivotvalue and rightmark >=
```

```
leftmark:
            rightmark = rightmark -1

        if rightmark < leftmark:
            done = True
        else:
            temp = alist[leftmark]
            alist[leftmark] = alist[rightmark]
            alist[rightmark] = temp

    temp = alist[first]
    alist[first] = alist[rightmark]
    alist[rightmark] = temp


    return rightmark

alist = [54,26,93,17,77,31,44,55,20]
quickSort(alist)
print(alist)
```