Содержание

[**1. Цель работы**](#_k5vdaq8hmnxz) **3**

[**2. Подмножество языка программирования**](#_1fob9te) **4**

[**2.1 Числовые и строковые константы**](#_3th6y9ax3o2g) **4**

[**2.2 Типы переменных**](#_57w84elv0kxk) **4**

[**2.3 Условные операторы**](#_i73qw3zbwd8v) **5**

[**3. Инструментальная языковая среда**](#_5swn028am6em) **7**

[**Примечание. Код программ**](#_uyw0m9k4f85u) **8**

# 

# 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 c указанием версии, на котором ведётся разработка (напр. Python 3.6);

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

- компьютер;

[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.](https://otvet.mail.ru/question/43083633)

# 2. Подмножество языка программирования

В качестве подмножества языка программирования выбран язык Python.

Python — динамический, интерпретируемый высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Язык обладает чётким и последовательным синтаксисом, продуманной модульностью и масштабируемостью, сильной динамической типизацией, сборщиком мусора и многими другими возможностями.

## 2.1 Числовые и текстовые константы

* -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 (int).
* 8.1, 4.3, -2.9, -11.7 (float литералы)
* "", "hello" (str литералы)
* True, False (boolean 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, целое число произвольной точности, число с плавающей запятой 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.

## 2.3 Операторы цикла

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

def collection\_sum(array):

i = 0

sum = 0

while i < len(array):

sum += array[i]

i += 1

return sum

* for - выполняет тело цикла, итерируясь по объекту (к примеру, строке, списку и т.д.)

def collection\_sum(array):

sum = 0

for element in array:

sum += element

return sum

* break - прерывает исполнение цикла

def example():

for value in “string“:

if value == “i“:

break

print(value)

print(“The end“)

## 2.3 Условные операторы

* Оператор *if*

if выражение:  
 инструкция\_**1**  
 инструкция\_**2**  
 ...  
 инструкция\_n

Пример:

def example():

num = 3

if num > 0:

print(num, “is a positive number.“)

print(“This is always printed.“)

* Конструкция if - else

if выражение:  
 инструкция\_1  
 инструкция\_2  
 ...  
 инструкция\_n  
else:  
 инструкция\_a  
 инструкция\_b  
 ...  
 инструкция\_x

Пример:

def example(array):

for element in array:

if element % 2 == 0:

print(element, “even“)

else:

print(element, “odd“)

# 3. Инструментальная языковая среда

В качестве языковой среды выбран язык программирования Python.

Разработка основана на работе с операционной системой Linux 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 поддерживает структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное программирование. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

Эталонной реализацией Python является интерпретатор CPython, поддерживающий большинство активно используемых платформ. Он распространяется под свободной лицензией Python Software Foundation License, позволяющей использовать его без ограничений в любых приложениях, включая проприетарные. Есть реализация интерпретатора для JVM с возможностью компиляции, CLR, LLVM, другие независимые реализации. Проект PyPy использует JIT-компиляцию, которая значительно увеличивает скорость выполнения Python-программ.

# Примечание. Код программ

1. Нахождения факториала числа, введенного пользователем

def converter(kilometers)

|  |
| --- |
| **def** converter(kilometers):  if value < 0:  return 0  conv\_fac = 0.621371  miles = kilometers \* conv\_fac  return miles |

1. Быстрая сортировка

|  |
| --- |
| **def** sort(array)  **return** [] **if** array.empty?  left, right = array[1..-1].partition { |y| y <= array.first }  sort(left) + [ array.first ] + sort(right)  **end** |