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 Ruby.

Ruby — динамический, рефлективный, интерпретируемый высокоуровневый язык программирования Язык обладает независимой от операционной системы реализацией многопоточности, сильной динамической типизацией, сборщиком мусора и многими другими возможностями. По особенностям синтаксиса он близок к языкам Perl и Eiffel, по объектно-ориентированному подходу — к Smalltalk. Также некоторые черты языка взяты из Python, Lisp, Dylan и Клу.

## 2.1 Числовые и строковые константы

* -3, -2, -1, 0, 1, 2, 3 (int).
* 3.5, -2.7 (float литералы)
* "", "hello" (str литералы)
* ^"", ^"hello" (unicode литералы)

## 2.2 Типы переменных

Ruby 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 Ruby имеются встроенные типы: булевый, строка, целое число произвольной точности, число с плавающей запятой и некоторые другие. Из коллекций в Ruby 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.

def insertion\_sort(array)

(array.length).times do |j|

while j > 0

if array[j-1] > array[j]

array[j], array[j-1] = array[j-1], array[j]

else

break

end

j-=1

end

end

array

end

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

def selection\_sort(array)

n = array.length - 1

n.times do |i|

min\_index = i

for j in (i + 1)..n

min\_index = j if array[j] < array[min\_index]

end

array[i], array[min\_index] = array[min\_index], array[i] if min\_index != i

end

array

end

* break - прерывает исполнение цикла

def create\_max\_heap(array, parent, limit)

root = array[parent]

while (child = 2 \* parent) <= limit do

child += 1 if child < limit && array[child] < array[child + 1]

break if root >= array[child]

array[parent], parent = array[child], child

end

array[parent] = root

end

## 2.3 Условные операторы

* Оператор *if*

if выражение then  
 инструкция\_**1**  
 инструкция\_**2**  
 ...  
 инструкция\_n  
end

Пример:

def quick\_sort(array)

return array if array.length <= 1

pivot = array.delete\_at(rand(array.length))

left = Array.new

right = Array.new

array.each do |x|

if x <= pivot

left << x

else

right << x

end

end

return \*quick\_sort(left), pivot ,\*quick\_sort(right)

end

* Конструкция if - else

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

Пример:

def quick\_sort(array)

return array if array.length <= 1

pivot = array.delete\_at(rand(array.length))

left = Array.new

right = Array.new

array.each do |x|

if x <= pivot

left << x

else

right << x

end

end

return \*quick\_sort(left), pivot ,\*quick\_sort(right)

end

# 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** factorial n  n > 1 ? n \* factorial(n - 1) : 1  **end** |

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** |