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. 853504

Кузьма В.В.

Проверил:

Ст. преподаватель КИ Шиманский В. В.

Минск 2021

Содержание

[1. Цель работы 3](#_Toc63359209)

[2. Подмножество языка программирования 4](#_Toc63359210)

[2.1 Числовые и строковые константы 4](#_Toc63359211)

[2.2 Типы переменных 5](#_Toc63359212)

2.3 Операторы цикла 5

2.4 Условные операторы 5

[3. Инструментальная языковая среда 7](#_Toc63359213)

[Приложение. Текст программ 8](#_Toc63359214)

# 

# 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 3.7);

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

- компьютер ([PC / Macintosh).](https://otvet.mail.ru/question/43083633)

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 C++.

Язык программирования С++ представляет высокоуровневый компилируемый язык программирования общего назначения со статической типизацией, который подходит для создания самых различных приложений. На сегодняшний день С++ является одним из самых популярных и распространенных языков.

С++ является мощным языком, унаследовав от Си богатые возможности по работе с памятью. Поэтому нередко С++ находит свое применение в системном программировании, в частности, при создании операционных систем, драйверов, различных утилит, антивирусов и т.д. К слову сказать, ОС Windows большей частью написана на С++. Но только системным программированием применение данного языка не ограничивается. С++ можно использовать в программах любого уровня, где важны скорость работы и производительность. Нередко он применяется для создания графических приложений, различных прикладных программ. Также особенно часто его используют для создания игр с богатой насыщенной визуализацией. Кроме того, в последнее время набирает ход мобильное направление, где С++ тоже нашел свое применение. И даже в веб-разработке также можно использовать С++ для создания веб-приложений или каких-то вспомогательных сервисов, которые обслуживают веб-приложения. В общем С++ - язык широкого пользования, на котором можно создавать практически любые виды программ.

С++ является компилируемым языком, а это значит, что компилятор транслирует исходный код на С++ в исполняемый файл, который содержит набор машинных инструкций. Но разные платформы имеют свои особенности, поэтому скомпилированные программы нельзя просто перенести с одной платформы на другую и там уже запустить. Однако на уровне исходного кода программы на С++ по большей степени обладают переносимостью, если не используются какие-то специфичные для текущей ос функции. А наличие компиляторов, библиотек и инструментов разработки почти под все распространенные платформы позволяет компилировать один и тот же исходный код на С++ в приложения под эти платформы.

## 2.1 Числовые и строковые константы

* --3, -2, -1, 0, 1, 2, 3 (int литералы)
* -1.3, -1.2, -1.1, 0.0, 1.1, 1.2, 1.3 (float литералы)
* ‘1’, ‘2’, ‘3’, ‘a’, ‘b’, ‘c’ (символьные литералы)
* “aa”, “ab”, “abb”, “abbb” (строковые литералы)

2.2 Типы переменных

В языке C++ есть следующие типы данных:

* bool
* byte
* int
* float
* char
* string
  1. Операторы цикла

do

{

//тело цикла

}

while (условие);

while (условие)

{

//тело цикла

}

for (;;) {

//тело цикла

}

## 2.4 Условные операторы

* Конструкция if / else

if (условие){

//инструкция

}

else if (условие) {

//инструкция

}

else {

//инструкция

}

* Конструкция switch / case

switch(сравнимое значение) {

case 1: //инструкция; break;

…

case n: //инструкция; break;

default: //инструкция; break;

}

# 3. Инструментальная языковая среда

В качестве языковой среды выбран язык программирования Python(3.9).

Разработка основана на работе с операционной системой Windows на PC.

Python — мощный и простой для изучения язык программирования. Он позволяет использовать эффективные высокоуровневые структуры данных и предлагает простой, но эффективный подход к объектно-ориентированному программированию. Сочетание изящного синтаксиса, динамической типизации в интерпретируемом языке делает Python идеальным языком для написания сценариев и ускоренной разработки приложений в различных сферах и на большинстве платформ.

Интерпретатор Python и разрастающаяся стандартная библиотека находятся в свободном доступе в виде исходников и двоичных файлов для всех основных платформ на официальном сайте Python [http://www.python.org](http://www.python.org/) и могут распространяться без ограничений.

## 

# Приложение. Текст программ

1. Бинарный поиск

|  |
| --- |
| #include<iostream> using namespace std; int main(){  int a[10] = {1, 2, 3, 4, 5, 6, 7, 8, 10};  int n = 10, key = 5;  int l = 0;  int r = n;  int mid;   while (l < r) {  mid = (l + r) / 2;   if (a[mid] > key) r = mid;  else l = mid + 1;  }   r--;   if (a[r] == key) cout << "Индекс элемента " << key << " в массиве равен: " << r;  else cout << "Извините, но такого элемента в массиве нет"; } |

1. Числа Смита

|  |
| --- |
| #include<iostream> using namespace std; int n,p,coun; int sum(int n) {  int s = 0;  while (n > 0) {  s += n % 10;  n /= 10;  }  return s; }  int main() {  while (cin >> n) {  int k1 = sum(n), k2 = 0, coun = 0;  p = 2;  while (p\*p <= n)  if (n%p == 0) {  k2 += sum(p);  n /= p;  coun++;  }  else p++;  k2 += sum(n);  if (k2 == k1 && coun>0) cout << 1; else cout << 0;   } } |

1. Минимальное остовное дерево в несвязном графе

#include <bits/stdc++.h>  
  
using namespace std;  
int x,color[100000],n,i,m,h[100000],ne[100000],y,ans=0,cost,c[100000];  
struct kek{  
 int h;  
 int ne;  
 int c;};  
kek graf[100000];  
  
bool cmp(kek l, kek r) {  
 return l.c<r.c;  
}  
  
int get(int x) {  
 if (x==color[x]) return x;  
 else {int c=get(color[x]);  
 color[x]=c;  
 return c;}  
}  
  
  
int main()  
{  
 cin>>n>>m;  
 for (i=1; i<=n; i++)  
 color[i]=i;  
 for (i=1; i<=m; i++)  
 cin>>graf[i].h>>graf[i].ne>>graf[i].c;  
 sort(graf+1, graf+m+1, cmp);  
 for (i=1; i<=m; i++) {  
 x=graf[i].h;  
 y=graf[i].ne;  
 cost=graf[i].c;  
 x=get(x);  
 y=get(y);  
 if (x!=y) {  
 ans+=cost;  
 color[x]=y;  
  
 }  
 }  
 cout<<ans;  
  
  
  
 return 0;  
}