

Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет  
информатики и радиоэлектроники»

Факультет компьютерных систем и сетей

Кафедра информатики

Дисциплина: Методы трансляции

ОТЧЁТ  
по лабораторной работе  
на тему

Определение модели языка. Выбор инструментальной языковой среды

Выполнил  
Студент гр. 053502  
Аралин И.О.

Проверил  
Ассистент кафедры информатики  
Гриценко Н.Ю.

Минск 2023

## СОДЕРЖАНИЕ

1. ЦЕЛЬ РАБОТЫ .....	3
2. ПОДМНОЖЕСТВО ЯЗЫКА ПРОГРАММИРОВАНИЯ.....	4
2.1 Литералы (константы).....	4
2.2 Типы данных .....	5
2.3 Операторы цикла .....	5
2.4 Условные операторы .....	5
3. ИНСТРУМЕНТАЛЬНАЯ ЯЗЫКОВАЯ СРЕДА .....	7
Приложение. Текст программы.....	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 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 C++.

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

C++ является мощным языком, унаследовав от Си богатые возможности по работе с памятью. Поэтому нередко C++ 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 C++. Но только системным программированием применение данного языка не ограничивается. C++ 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 C++ тоже нашел свое применение. И даже в веб-разработке также можно использовать C++ для создания веб-приложений или каких-то вспомогательных сервисов, которые обслуживают веб-приложения. В общем C++ - язык широкого пользования, на котором можно создавать практически любые виды программ.

C++ является компилируемым языком, а это значит, что компилятор транслирует исходный код на C++ 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 C++ 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 C++ в приложения под эти платформы.

### **2.1 Литералы (константы)**

- -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5 (int литералы)

- -1.5, -1.25, -1.05, 0.0, 1.05, 1.25, 1.5 (float литералы)
- '0', '1', '2', '3', '4', 'a', 'b', 'c', 'd', 'e' (символьные литералы)
- "aa", "ab", "abb", "abbb" (строковые литералы)

## 2.2 Типы данных

Поддерживаются следующие типы данных:

- Целые типы данных: int, short, long, long long и т.д.
- Типы данных с плавающей точкой: float, double, long double и т.д.
- Логический тип данных: bool
- Тип данных символа: char
- Тип данных строки: string
- Типы данных для хранения адресов памяти: pointer

## 2.3 Операторы цикла

while (условие)

```
{
    // тело цикла
}
```

do

```
{
    // тело цикла
}
```

while (условие);

for (оператор 1; оператор 2; оператор 3)

```
{
    // 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 MacOS 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> и могут распространяться без ограничений.

## Приложение. Текст программы

### 1. Быстрая сортировка

```
#include <iostream>
using namespace std;

void quickSort(int *array, int first, int last)
{
    int mid, count;
    int f = first, l = last;
    mid = array[(f + l) / 2];

    do {
        while (array[f] < mid) f++;
        while (array[l] > mid) l--;
        if (f <= l)
        {
            count = array[f];
            array[f] = array[l];
            array[l] = count;
            f++;
            l--;
        }
    } while (f < l);

    if (first < l)
    {
        quickSort(array, first, l);
    }

    if (f < last)
    {
        quickSort(array, f, last);
    }
}

void printArray(int arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        cout << arr[i] << " ";
    }
}
```



```

    cout << endl;
}

int main()
{
    int arr[] = {64, -322, 10, 22, -1, 4, 100, 100, 21};
    int n = sizeof(arr) / sizeof(int);

    cout << "Original Array: \n";
    printArray(arr, n);

    quickSort(arr, 0, n);

    cout << "\nSorted Array: \n";
    printArray(arr, n);

    return 0;
}

```

## 2. Шифр Цезаря

```

#include <iostream>
#include <string>
using namespace std;

string encryptCaesar(const string &text, int cipher)
{
    string enc_text;

    for (char i: text)
    {
        if (isalpha(i))
        {
            char c = i;

            if (isupper(c))
            {
                c = (c - 'A' + cipher) % 26 + 'A';
            }
            else
            {
                c = (c - 'a' + cipher) % 26 + 'a';
            }
        }
    }
}

```

```

        enc_text += c;
    }
    else
    {
        enc_text += i;
    }
}

return enc_text;
}

string decryptCaesar(const string& enc_text, int cipher)
{
    string dec_text;

    for (char i: enc_text)
    {
        if (isalpha(i))
        {
            char c = i;

            if (isupper(c))
            {
                c = (c - 'A' - cipher + 26) % 26 + 'A';
            }
            else
            {
                c = (c - 'a' - cipher + 26) % 26 + 'a';
            }

            dec_text += c;
        }
        else
        {
            dec_text += enc_text[i];
        }
    }

    return dec_text;
}

int main()
{
    string text;

```

```
cout << "Enter the text sample: ";
getline(cin, text);

int cipher;
cout << "Enter the cipher: ";
cin >> cipher;
cipher %= 26;

cout << "Resulting encrypted Text: " << encryptCaesar(text, cipher) << endl;
cout << "Resulting decrypted Text: " << decryptCaesar(encryptCaesar(text,
cipher), cipher) << endl;

return 0;
}
```