

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  
Крупенич Е.Г.

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)**

- -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 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 C# 11.0. Разработка основана на работе с ОС Windows на PC.

C# (произносится как "си шарп") - современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать разные типы безопасных и надежных приложений, выполняющихся в .NET. C# относится к широко известному семейству языков C, и покажется хорошо знакомым любому, кто работал с C, C++, Java или JavaScript.

C# - объектно-ориентированный, ориентированный на компоненты язык программирования. C# предоставляет языковые конструкции для непосредственной поддержки такой концепции работы. Благодаря этому C# подходит для создания и применения программных компонентов. С момента создания язык C# обогатился функциями для поддержки новых рабочих нагрузок и современными рекомендациями по разработке ПО. В основном C# - объектно-ориентированный язык. Вы определяете типы и их поведение.

## Приложение. Текст программы

### 1. Сортировка пузырьком

```
#include <iostream>

using namespace std;

void bubbleSort(int arr[], int n) {
    for (int i = 0; i < n-1; i++) {
        for (int j = 0; j < n-i-1; j++) {
            if (arr[j] > arr[j+1]) {
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}

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

int main() {
    int arr[] = {64, 34, 25, 12, 22, 11, 90};
    int n = sizeof(arr)/sizeof(arr[0]);

    cout << "Original array:" << endl;
    printArray(arr, n);

    bubbleSort(arr, n);

    cout << endl << "Sorted array:" << endl;
    printArray(arr, n);
    return 0;
}
```

### 2. Шифр Цезаря

```
#include <iostream>
```



```

using namespace std;

void Caesar()
{
    int key = 0;
    string str = "";

    cout << "Enter string to encode: ";
    cin >> str;

    cout << "Enter key: ";
    cin >> key;

    cout << "String to encode: " << str << endl;
    key %= 26;

    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] + key > 'z')
            str[i] -= 26;
        str[i] += key;
    }

    cout << "Encoded string: " << str << endl;

    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] - key < 'a')
            str[i] += 26;
        str[i] -= key;
    }

    cout << "Decoded string: " << str << endl << endl;
}

int main()
{
    Caesar();
    return 0;
}

```