Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей   
Кафедра информатики

Дисциплина: Методы трансляции

**ОТЧЕТ**

к лабораторной работе № 1

на тему «Определение модели языка. Выбор инструментальной языковой среды»

Выполнил:

студент гр. 153504

Михалевич M.П.

Проверил:

Гриценко Н.Ю.

Минск 2024

СОДЕРЖАНИЕ

[1 Цели работы 3](#_Toc158054789)

[2 Определение подмножества языка программирования C 4](#_Toc158054790)

[2.1 Числовые и текстовые константы 4](#_Toc158054791)

[2.2 Типы переменных 4](#_Toc158054792)

[2.3 Операторы цикла 5](#_Toc158054793)

[2.4 Условные операторы 5](#_Toc158054794)

[2.5 Структуры данных 5](#_Toc158054795)

[2.6 Подключение библиотек 6](#_Toc158054796)

[2.7 Функции 6](#_Toc158054797)

[3 Определение инструментальной языковой среды 7](#_Toc158054798)

[Вывод 8](#_Toc158054799)

[Список использованных источников 9](#_Toc158054800)

[Приложение А (обязательное) Пример реализации программ на языке                                        программирования С 10](#_Toc158054801)

## ЦЕЛИ РАБОТЫ

Целью данной лабораторной работы является определение подмножества языка программирования, тексты 2-3-х программ, включающих все элементы этого подмножества. Также определение инструментальной языковой среды, т.е. языка программирования c указанием версии и операционной системы для разработки.

## ОПРЕДЕЛЕНИЕ ПОДМНОЖЕСТВА ЯЗЫКА ПРОГРАММИРОВАНИЯ C

## Числовые и текстовые константы

Константы в C – это фиксированные значения, которые не меняются во время выполнения программы. Они представляют собой основные строительные блоки для переменных и выражений в программировании. В C константы могут быть разных типов, включая:

1. Целочисленные константы. Это числа без десятичных точек. Они могут быть представлены в десятичном (база 10), восьмеричном (база 8, начинается с 0), или шестнадцатеричном (база 16, начинается с 0x или 0X) формате. Примеры: 123, 0x7B, 0173.
2. Константы с плавающей точкой. Это числа с десятичными точками или в экспоненциальном формате, представляющие вещественные числа. Примеры: 3.14, 2.5e2, 3.0E-5.
3. Символьные константы. Одиночные символы, заключенные в одинарные кавычки, представляющие символы в кодировке ASCII. Примеры: 'a', '1', '\n'.
4. Строковые константы. Последовательности символов, заключенные в двойные кавычки. В конце каждого строкового литерала автоматически добавляется символ нулевого терминатора ('\0'), который указывает на конец строки. Пример: "Hello, World!".[1]

## Типы переменных

В языке программирования C каждая переменная ассоциируется с конкретным типом данных, который задаёт допустимый диапазон значений переменной, возможные операции над ней и объём памяти, который она займёт. Основные типы данных, предусмотренные в C, включают в себя:

1. Логический тип: В C логическое значение представляется через целочисленные значения, где 0 интерпретируется как false (ложь), а любое ненулевое значение – как true (истина).
2. Целочисленные типы: Включают в себя int, short, long и long long, как знаковые, так и беззнаковые вариации. Они предназначены для работы с целыми числами различного размера, от маленьких до очень больших.
3. Типы чисел с плавающей точкой: float, double и long double представляют числа с дробной частью. Они различаются по размеру памяти, который занимают, и по точности представления значений.
4. Символьные типы: char используется для хранения отдельных символов, таких как буквы алфавита или знаки пунктуации. char может быть как знаковым, так и беззнаковым, в зависимости от реализации.
5. Строковые типы: В языке C строки представляют собой массивы символов типа char, завершающиеся нулевым символом ('\0'). Хотя строки не являются отдельным типом данных в строгом смысле, они являются важной частью работы с текстовыми данными.[2]

## Операторы цикла

Операторы цикла в языке программирования C используются для выполнения блока кода многократно под определёнными условиями. В C есть три основных типа циклов:

1. Цикл for. Цикл for используется, когда число итераций известно заранее. Он состоит из трёх частей: инициализации, условия и итерации.
2. Цикл while. Цикл while выполняется, пока его условие остаётся истинным. Инициализация переменной счётчика обычно происходит до начала цикла, а изменение счётчика – внутри тела цикла.
3. Цикл do-while. Цикл do-while аналогичен циклу while, но его тело выполняется хотя бы один раз перед проверкой условия.[3]

## Условные операторы

Условные операторы в языке программирования C позволяют программе принимать решения на основе выполнения или невыполнения определённых условий. Эти операторы оценивают выражения, которые истинны или ложны, и на основе этой оценки определяют, какие блоки кода следует выполнить. Основных условные операторы:

1. Оператор if. Оператор if используется для выполнения блока кода, если и только если заданное условие истинно.
2. Оператор if-else. Оператор if-else расширяет оператор if, позволяя выполнить альтернативный блок кода, если условие ложно.
3. Оператор if-else if-else. Для проверки нескольких условий последовательно используется цепочка операторов if-else if-else.
4. Оператор switch. Оператор switch предназначен для выполнения одного из множества блоков кода на основе значения выражения. Он особенно полезен, когда нужно сравнить одну переменную с несколькими константными значениями.
5. Тернарный оператор. Тернарный оператор позволяет выбирать между двумя выражениями в зависимости от истинности или ложности заданного условия. Синтаксис тернарного оператора включает три части и выглядит следующим образом: условие ? выражение1 : выражение2. Если условие истинно, выполняется выражение1; если ложно – выражение2.[4]

## Структуры данных

Структуры данных в языке программирования C позволяют организовать и хранить данные для их эффективного использования. Основные структуры данных в C:

1. Массивы. Массив – это набор элементов одного и того же типа, хранящихся в памяти последовательно. Массивы используются для хранения коллекций данных одинакового типа.
2. Структуры (Structs). Структуры позволяют группировать переменные разных типов под одним именем. Они используются для представления записей данных, которые имеют несколько свойств.
3. Указатели. Указатель – это переменная, значение которой является адресом другой переменной. Указатели широко используются для динамического управления памятью, работы со строками и массивами, а также для передачи функциям адресов переменных и структур.[5]

## Подключение библиотек

В языке программирования C подключение библиотек осуществляется с помощью директив препроцессора #include. Эти директивы указывают компилятору включить содержимое указанного файла в место, где находится директива #include, до начала компиляции исходного кода.[6]

## Функции

В языке программирования C функции представляют собой блоки кода, предназначенные для выполнения конкретной задачи. Функции позволяют структурировать программы на более мелкие и управляемые части, обеспечивая повторное использование кода и упрощение отладки.

Функция определяется с указанием типа возвращаемого значения, имени функции и списка параметров (аргументов) в круглых скобках. Тело функции заключается в фигурные скобки.

Функции могут принимать параметры (аргументы), которые позволяют передавать значения в функцию. Эти параметры могут быть переданы «по значению», что означает копирование значения аргумента в параметр функции, или «по ссылке» (используя указатели), что позволяет функции изменять значение аргумента.

Функция может возвращать значение. Тип возвращаемого значения указывается перед именем функции. Если функция не возвращает значение, используется ключевое слово void.[7]

## ОПРЕДЕЛЕНИЕ ИНСТРУМЕНТАЛЬНОЙ ЯЗЫКОВОЙ СРЕДЫ

Для разработки транслятора был выбран язык программирования Python 3.11.7. Выбор языка программирования Python для разработки транслятора не является случайным, а скорее обусловлен рядом значительных преимуществ, которые этот язык предлагает разработчикам в контексте такой сложной и ответственной задачи. Python – это высокоуровневый, интерпретируемый язык программирования с динамической типизацией, который обладает чистым и лаконичным синтаксисом, делающим код легко читаемым и понятным. Эти качества делают Python идеальным инструментом для создания сложных программных систем, таких как трансляторы, интерпретаторы и компиляторы.

Разработка будет выполнятся на PC в операционной системе Windows.

## ВЫВОД

В ходе выполнения лабораторной работы были определены подмножества языка программирования, тексты 2-3-х программ, включающих все элементы этого подмножества. Также определена инструментальная языковая среда, т.е. язык программирования c указанием версии и операционная система для разработки.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Константы и переменные в языке Си [Электронный ресурс]. – Режим доступа: https://prog-cpp.ru/c-constants/. – Дата доступа: 05.02.2024.
2. Типы данных [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/2.3.php. – Дата доступа: 05.02.2024.
3. Циклы [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/2.12.php. – Дата доступа: 05.02.2024.
4. Условные конструкции [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/2.11.php. – Дата доступа: 05.02.2024.
5. Структуры [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/6.1.php. – Дата доступа: 05.02.2024.
6. Библиотеки Си [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/8.1.php. – Дата доступа: 05.02.2024.
7. Функции [Электронный ресурс]. – Режим доступа: https://metanit.com/c/tutorial/4.1.php. – Дата доступа: 05.02.2024.

## ПРИЛОЖЕНИЕ А (обязательное) Пример реализации программ на языке программирования С

Листинг 1 – Использование числовых констант и условных операторов

#include <stdio.h>

int main() {

const int MIN\_AGE = 18;

int age = 20;

if (age >= MIN\_AGE) {

printf("Access granted.\n");

} else {

printf("Access denied.\n");

}

return 0;

}

Листинг 2 – Программная реализация пузырьковой сортировки

#include <stdio.h>

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]) {

// Меняем элементы местами

int temp = arr[j];

arr[j] = arr[j+1];

arr[j+1] = temp;

}

}

int main() {

int arr[] = {64, 34, 25, 12, 22, 11, 90};

int n = sizeof(arr)/sizeof(arr[0]);

bubbleSort(arr, n);

printf("Sorted array: \n");

for (int i=0; i < n; i++)

printf("%d ", arr[i]);

printf("\n");

return 0;

}  
  
Листинг 3 – Программная реализация структуры Student

#include <stdio.h>

struct Student {

char name[50];

int age;

float grade;

};

int main() {

struct Student student1 = {"John Doe", 20, 4.0};

printf("Name: %s\nAge: %d\nGrade: %.2f\n", student1.name, student1.age, student1.grade);

return 0;

}