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             К. А. Тимофеев

Проверил                          Н. Ю. Гриценко

Минск 2024

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc158646945)

[2 Опеределение модели языка программирования С 4](#_Toc158646946)

[2.1 Общие правила синтаксиса 4](#_Toc158646947)

[2.2 Переменные, константы и литералы 4](#_Toc158646948)

[2.3 Типы данных 5](#_Toc158646949)

[2.4 Операторы 7](#_Toc158646950)

[2.6 Функции 8](#_Toc158646951)

[2.7 Подключение библиотек 8](#_Toc158646952)

[2.8 Пользовательские типы 8](#_Toc158646953)

[3 Определение инструментальной языковой среды 9](#_Toc158646954)

[Выводы 10](#_Toc158646955)

[Список использованных источников 11](#_Toc158646956)

[Приложение А](#_Toc158646957) (обязательное) [Пример реализации программ на языке программирования С 12](#_Toc158646959)

## 1 ПОСТАНОВКА ЗАДАЧИ

Целью выполнения данной лабораторной работы является определить подмножество выбранного языка программирования, предоставить тексты нескольких программ, включающих все элементы этого подмножества, а также определить инструментальную языковую среду, которая включает в себя язык программирования с указанием версии, на котором ведется разработка, операционную систему, в которой выполняется разработка, и компьютер.

## 2 ОПРЕДЕЛЕНИЕ МОДЕЛИ ЯЗЫКА             ПРОГРАММИРОВАНИЯ С

## 2.1 Общие правила синтаксиса

Язык программирования *C* является языком со строгой статической типизацией. Это означает, что в каждой переменной может хранится только заранее определенный тип значений и что язык программирования не будет сам приводить типы при необходимости. Таким образом, работа с типами данных находится в руках разработчика.

Текст программы на языке *C* представляет собой набор инструкций, которые выполняются сверху вниз. Инструкции должны быть разделены символом «;». Программа начинается во входной точке (в большинстве случаев это функция «*main»*), и работает до тех пор, пока не выйдет из нее. За одну инструкции может быть выполнено несколько операций. Блоки кода обрамляются фигурными скобками.[1]

## 2.2 Переменные, константы и литералы

Для хранения данных в программе на языке *С* используются переменные, которые представляют собой именованные участки памяти. Переменная имеет тип, который определяет, какие именно данные может хранить переменная, имя и значение.

У каждой переменной должно быть имя. Имя переменной может состоять из букв латинского алфавита, цифр и знака подчеркивания. Имя переменной не может начинаться с цифры. Кроме того,   
в качестве имени переменной нельзя использовать ключевые слова языка *С*.

В общем случае определение переменной в языке программирования *С* представлено на рисунке 2.1.



Рисунок 2.1 – Определение переменной

Язык программирования *C* относится к регистрозависимым языкам,   
то есть строчные и заглавные символы в именах переменных считаются языком за разные символы.

После определения переменной можно присвоить ей некоторое значение. Значение можно присвоить во время определения. Общий вид такого определения представлен на рисунке 2.2.



Рисунок 2.2 – Определение переменной с изначально заданным значением

Присваивание переменным значения после инициализации при помощи оператора присваивания «=». Пример нотации присваивания на языке программирования *C* представлен на рисунке 2.3.



Рисунок 2.3 – Пример нотации присваивания на языке программирования *С*

Во время объявления переменной ее можно сделать константой. Константные переменные не могут изменить своего значения во время выполнения программы.

Для присваивания значений на этапе написания программы используют литералы. В языке *C* есть всего 4 типа литералов:

– числовые целые;

– числовые с плавающей точкой;

– символьные;

– строковые.

Каждый из этих литералов соответствует определенным типам данных.

## 2.3 Типы данных

Каждая переменная имеет определенный тип. И этот тип определяет, какие значения может иметь переменная, какие операции с ней можно производить и сколько байт в памяти она будет занимать. В языке *C* определены следующие базовые типы данных:

– целочисленные типы;

– тип чисел с плавающей точкой;

– символьные типы.[2]

К целочисленным типам, помимо типов непосредственно отвечающих за работу с числами, относится тип *char*. Это связано с особенностями представления символов через таблицу символов *ASCII*.

Типы данных языка *C*, их размер в байтах и соответствующие литералы представлены в таблице 2.1.

Таблица 2.1 – Типы данных в языке *C*

|  |  |  |  |
| --- | --- | --- | --- |
| Имя типа | Размер, байт | Примеры литералов | Описание |
| *char* | 1 | ‘c’, ‘\n’ | Тип данных для одного символа |
| *short* | 2 | 32, -13 | Целочисленный тип данных |
| *int* | Зависит от системы | 45, -16 | Целочисленный тип данных с большим чем у *short* диапазоном значений |
| *long* | 4 | 68, -32 | Целочисленный тип данных с большим чем у *short* диапазоном значений и независящим от системы размером |
| *long long* | 8 | 23, -78 | Целочисленный тип данных с большим диапазоном значений |
| *float* | 4 | 1.78, 1.7E-8 | Тип чисел с плавающей точкой |
| *double* | 8 | 1.78, 1.7E-8 | Тип чисел с плавающей точкой с большими чем у *float* диапазоном значений и точностью |

Продолжение таблицы 2.1

|  |  |  |  |
| --- | --- | --- | --- |
| Имя типа | Размер, байт | Примеры литералов | Описание |
| *long double* | 10 | 11.78, 1.7E-8 | Тип чисел с плавающей точкой с самыми большими диапазоном значений и точностью |
| *void* | Тип не имеет размера | Тип не имеет литералов | Особый тип, обозначающий отсутствие типа |

Типам данных *char*, *int*, *short*, *long*, *long* могут добавлены префиксы *signed* или *unsigned*. Благодаря ним можно увеличить диапазон положительных значений или определить логику выполнения операций.

## 2.4 Операторы

Операторы представляют собой специальный символы, использующиеся для выполнения операций над базовыми типами данных. Их можно разделить на 7 групп:

– арифметические («+», «\*», «/», «-», «--», «++», «%»);

– сравнения («==», «!=», «>=», «<=», «>», «<»);

– побитовые («&», «|», «^», «~», «>>», «<<»);

– логические («&&», «||», «!»,);

– присваивания («+=», «-=», «/=», «%=», «\*=», «&=», «|=», «^=», «<<=», «>>=»);

– доступа («[]», «.», «->»);

– особые («*sizeof*», «()», «?:»).[3]

Арифметические операции представляют собой операции сложения, вычитания, умножения, деления, вычисления остатка от деления.

Операторы сравнения используются для сравнения чисел в условных конструкциях.

Побитовые операции позволяют манипулировать переменными как наборами битов.

Логические операторы позволяют составлять составные условия в условных конструкциях.

Операторы присваивания позволяют записывать в переменные новые значения.

Операторы доступа обеспечивают доступ к переменным в массивах, структурах и объединениях.

В группе особых операторов находится оператор «*sizeof*», возвращающий размер переменной или какого-либо типа. Оператор «()» используется при вызовах функций. Тернарный оператор «?:» позволяет коротко записывать условные конструкции.

## 2.6 Функции

По сути функция – это именованный блок кода. Первая строка представляет заголовок функции. Вначале указывается возвращаемый тип функции. Если функция не возвращает никакого значения, то используется тип *void*.

Для возвращения результата функция применяет оператор «*return*». Если функция имеет в качестве возвращаемого типа любой тип, кроме «*void*», то она должна обязательно с помощью ключевого слова «*return*» возвращать какое-либо значение.

При вызове функций в них можно передать параметры. Переданные параметры станут переменными, к которым можно будет обратиться из кода функции.

## 2.7 Подключение библиотек

В языке программирования С для подключения библиотек используется директива «*#include*». Эта директива позволяет включать содержимое заголовочных файлов в программу перед компиляцией.   
После подключения библиотеки заголовочные файлы становятся доступными для использования в программе, что позволяет обращаться к их функциям, классам и другим элементам.

## 2.8 Пользовательские типы

Язык программирования *C* предоставляет ограниченные возможности создания собственных типов данных. Пользователь может создавать объединения, перечисления и структуры.

Перечисления представляют собой именованный набор констант. Они используются для ограничения множества возможных значений переменных.

Объединения представляют собой области памяти, порядок битов в которых определяет значения членов объединения.

Структуры позволяют сгруппировать несколько переменных в единую переменную.

# 3 ОПРЕДЕЛЕНИЕ ИНСТРУМЕНТАЛЬНОЙ ЯЗЫКОВОЙ             СРЕДЫ

Для разработки транслятора был выбран язык программирования *Python* 3.12.1. *Python* – это современный и мощный язык программирования общего назначения, который отличается высокой производительностью разработки   
и читаемостью кода.

Одним из основных преимуществ *Python* является автоматическое управление памятью с помощью механизмов подсчета ссылок и циклического сборщика мусора. При написании программ на языке программирования Python нет необходимости явно выделять и освобождать память, что упрощает процесс программирования и снижает вероятность ошибок.

*Python* также имеет динамическую строгую типизацию. Данный язык программирования обладает динамическим разрешением имен, что означает, что имена методов и переменных связываются с их объектами во время выполнения программы.

Кроме того, *Python* – это кроссплатформенный язык программирования, что означает, что программы, написанные на *Python*, могут быть запущены   
на различных операционных системах без изменений их исходного кода.

В качестве интегрированной среды разработки был выбран текстовый редактор *Visual Studio Code* от *Microsoft*.

Операционной системой выступает *Windows*. Работа проводится на *PC*.

## ВЫВОДЫ

В ходе выполнения данной лабораторной работы было проведено изучение и перечисление основных аспектов языка программирования *C*. Были описаны типы данных, операторы, особые возможности выбранного языка программирования. В результате определения модели языка программирования *C* были описаны основные принципы его работы, благодаря которым он завоевал свою популярность.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

[1] C Syntax [Электронный ресурс]. – Режим доступа: https://www.w3schools.com/c/c\_syntax.php. – Дата доступа: 09.02.2024.

[2] Data Types in C [Электронный ресурс]. – Режим доступа: https://www.geeksforgeeks.org/data-types-in-c/. – Дата доступа: 09.02.2024.

[3] С – Operators [Электронный ресурс]. – Режим доступа: https://www.tutorialspoint.com/cprogramming/c\_operators.htm. – Дата доступа: 09.02.2024.

## ПРИЛОЖЕНИЕ А

## (обязательное)

## Пример реализации программ на языке программирования С

Листинг 1 – Программа 1

#include “stdio.h”

int main() {

int a = 15;

int b = 4;

int c = a – b;

printf(“Hello world!\n”);

printf(“%i - %i = %i”, a, b, c)

return 0;

}

Листинг 2 – Программа 2

#include “stdio.h”

#include “stdlib.h”

struct A{

int a;

float b;

}

int main(){

struct A\* obj = (struct A\*) malloc(sizeof(struct A));

obj->a=17;

obj->b=1.7;

printf(“a = %i b = %f”, obj->a, obj->b);

return 0;

}