

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ
Кафедра экономической математики, информатики и статистики
(ЭМИС)

ОСНОВЫ АЛГОРИТМИЗАЦИИ. СРЕДА VISUAL STUDIO.
ВВЕДЕНИЕ В ЯЗЫК СИ

Отчёт по лабораторной работе по дисциплине “Введение в
программирование”

Студент гр. 543-1



А.Е. Мухамеджан

“ ____ ” ____ 2023

Руководитель доцент

кафедры ЭМИС

____ Е.А. Шельмина

“ ____ ” ____ 2023г.

Томск2023

Лабораторная работа №1

“Основы алгоритмизации. Среда Microsoft Visual Studio. Введение в язык Си”



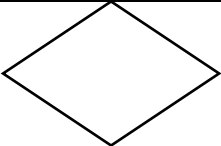
Введение: цель работы знакомство со средой разработки Visual Studio усвоение основ языка Си и Алгоритмизации.

Ход работы:

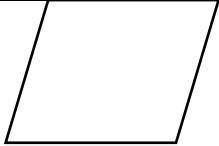
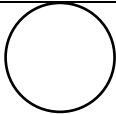
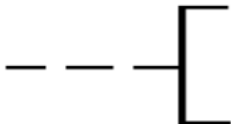
Основы Алгоритмизации: Алгоритмом называется точное предписание, определяющее последовательность действий исполнителя, направленных на решение поставленной задачи. В роли исполнителей алгоритмов могут выступать люди, роботы, компьютеры.

Блок схема – это графическое представление алгоритма при помощи стандартных обозначений. На схемах алгоритмов выполняемые действия изображаются в виде отдельных блоков, которые соединяются между собой линиями связи в порядке выполнения действий. Внутри блока дается информация о выполняемых действиях.

Таблица 1 – Основные блоки, используемые при составлении алгоритмов

Название	Обозначение	Назначение
Пуск, Остановка		Начало-конец алгоритма
Процесс		Любое вычислительное действие
Решение		Проверка условия

Окончание таблицы 1

Название	Обозначение	Назначение
Ввод-вывод		Ввод-вывод данных
Соединитель		Используется на линиях разрыва
Комментарий		Комментарий

Вычислительные процессы могут быть: линейные, разветвляющиеся и циклические.

Линейный алгоритм – алгоритм, в котором все команды выполняются последовательно друг за другом.

Часто при выполнении алгоритма должны предлагаться различные действия в зависимости от выполнения или невыполнения некоторого условия. Такие алгоритмические структуры называют ветвлением. Бывают алгоритмы с полным ветвлением (рисунок 1) и неполным (рисунок 2).

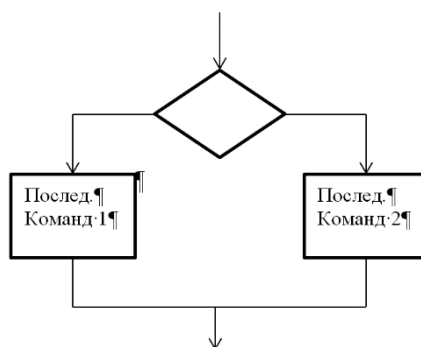


Рисунок 1 – Иллюстрация алгоритма с полным ветвлением

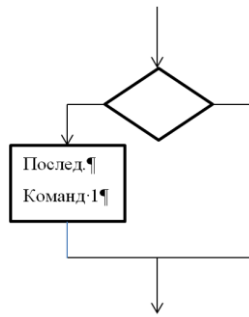


Рисунок 2 – Иллюстрация алгоритма с неполным ветвлением

Алгоритм с циклом – алгоритм, в котором присутствует многократное повторение одних и тех же действий. Такие алгоритмы подразделяются на алгоритмы с предусловием (рисунок 3) и постусловием (рисунок 4).

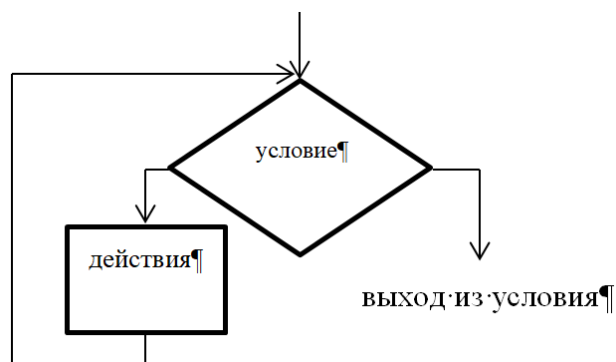


Рисунок 3 – Иллюстрация алгоритма с предусловным циклом

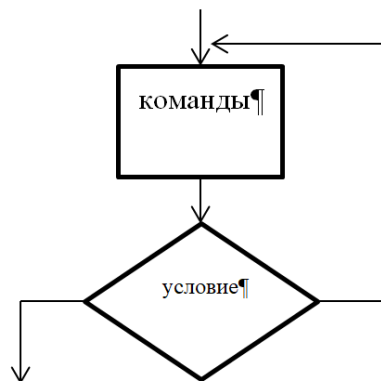


Рисунок 4 – Иллюстрация алгоритма с постусловным циклом

Среда программирования Microsoft Visual Studio: После открытия Microsoft Visual Studio (рисунок 5) можно увидеть три основные части интегрированной среды разработки: окна инструментов, меню с панелями инструментов и область главного окна. Окна 13 инструментов закреплены в левой и правой частях окна приложения, а панель Быстрый запуск, строка меню и стандартная панель инструментов закреплены в его верхней части. В

центре окна приложения находится Начальная страница. При открытии решения или проекта редакторы и конструкторы отображаются в этом пространстве. При разработке приложения чаще всего используется именно эта область.

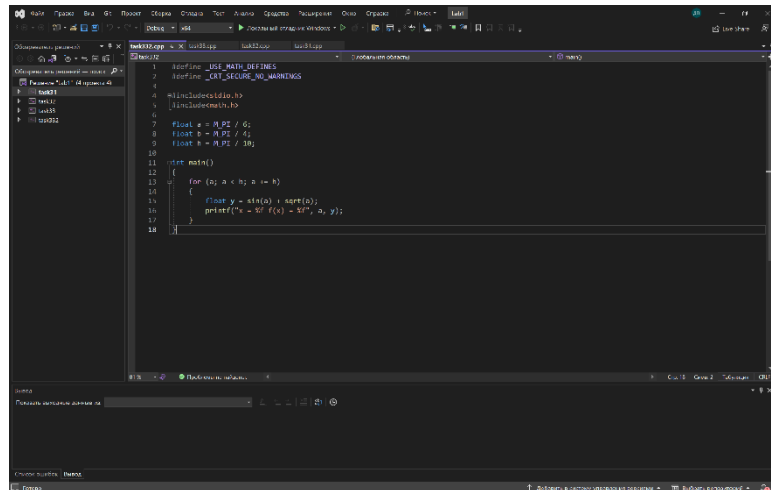


Рисунок 5 - Скриншот интегрированной среды разработки Microsoft Visual Studio

Создание простого приложения: При создании приложения в Microsoft Visual Studio необходимо сначала создать проект и решение.

Создание консольного приложения:

1. В строке меню выберите Файл, Создать, Проект.
2. В категории Visual C++ выберите шаблон Консольное приложение Win32 и назовите проект.
3. Когда появится мастер приложений Win32, нажмите кнопку Готово.

Проект и решение с основными файлами для консольного приложения Win32 создадутся и автоматически загрузятся в Обозреватель решений. Файл .cpp откроется в редакторе кода.

Добавление кода в приложение: Далее необходимо добавить код для отображения слова "Hello" в окне консоли. Для этого в окне консоли в файле .cpp введите пустую строку перед строкой return 0;, а затем введите в нее следующий код: printf("Hello\n");

Сохраните изменения в файле.

Отладка и тестирование приложения: С помощью отладки можно посмотреть, отображается ли слово Hello в окне консоли.

Отладчик запустится и выполнит код. Окно консоли (отдельное окно, подобное командной строке) отображается в течение нескольких секунд, но при остановке отладчика быстро закрывается. Чтобы просмотреть текст, необходимо установить точку останова выполнения программы.

Добавление точки останова:

1. Добавьте точку останова из меню в строке return 0.
2. Нажмите клавишу F5, чтобы начать отладку. Запускается отладчик, и появляется окно консоли, в котором выводится слово Hello.
3. Для останова процесса отладки нажмите SHIFT + F5.

Сборка окончательной версии приложения: Теперь, когда вы проверили, что все работает, можно подготовить окончательную сборку приложения.

Очистка файлов решения и сборка окончательной версии

1. Из строки меню удалите промежуточные файлы и выходные файлы, созданные во время предыдущих сборок.
2. Измените конфигурацию сборки с Отладка на Выпуск.
3. Постройте решение

Задания для самостоятельной работы:

Задание 1.1. Составить блок-схему решения линейного алгоритма:

Для выполнения задания 1 (рисунок 6) использовался вариант 16, представленный в файле лабораторной работы №1.

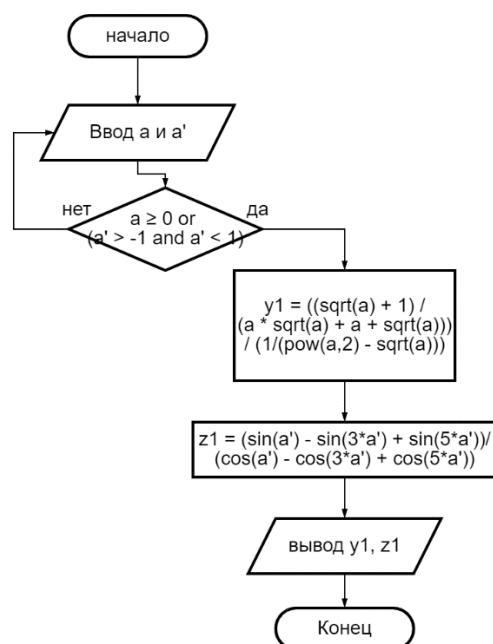


Рисунок 6 – Иллюстрация блок-схемы задания 1

Задание 1.2. Составить блок-схему решения разветвляющегося алгоритма:

Для выполнения задания 2 (рисунок 7) использовался вариант 16, представленный в файле лабораторной работы №1.

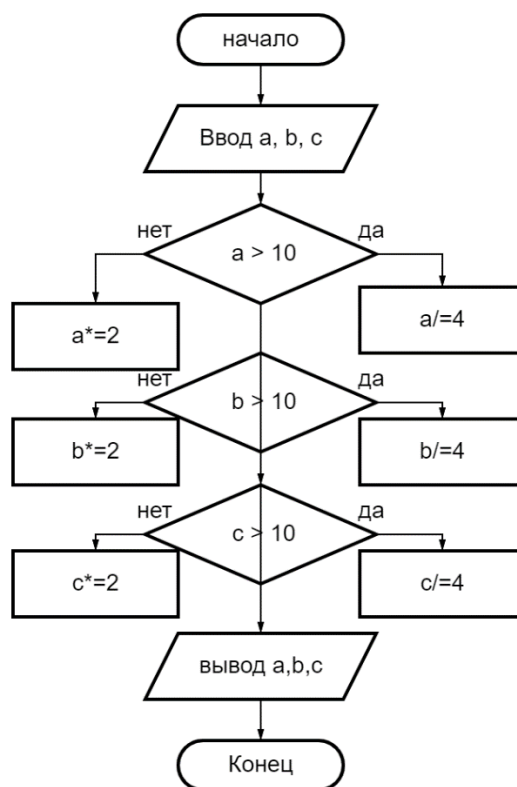


Рисунок 7 – Иллюстрация блок-схемы задания 2

Задание 1.3. Составить блок-схему решения циклического алгоритма.

Для выполнения задания 3 (рисунок 8) использовался вариант 16, представленный в файле лабораторной работы №1.

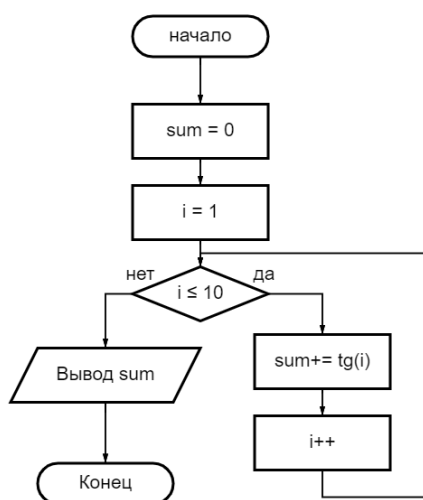


Рисунок 8 – Иллюстрация блок-схемы задания 3.1

Задание 1.3.2. Составить блок-схему алгоритма вычисления значений функции $f(x)$ на отрезке $[a;b]$ с шагом h .

Для выполнения задания 3.2 (рисунок 9) использовался вариант 16, представленный в файле лабораторной работы №1.

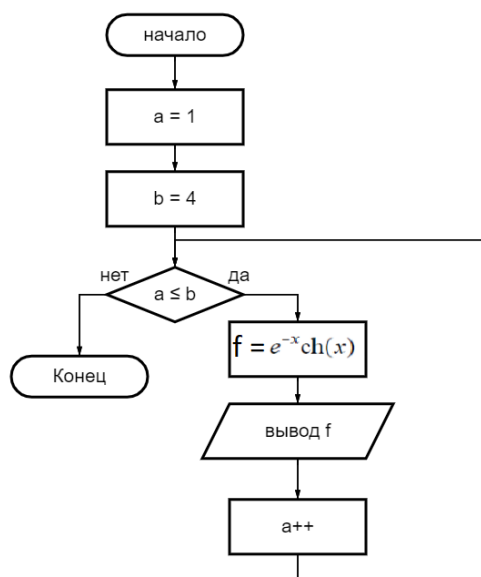


Рисунок 9 – Иллюстрация блок-схемы задания 3.2

Задание 3.1. Выполнить примеры 1-3.

Для выполнения примеров, показанных на рисунках 10, 11 и 12, использовалась среда разработки Microsoft Visual Studio и шаблон Консольное приложение.

```
7
8
9  #include <stdio.h>
10
11 int main()
12 {
13     printf("Welcome to C!\n");
14
15     return 0;
16 }
17
```

Рисунок 10 – Скриншот кода из примера 1

```
7
8
9  #include <stdio.h>
10
11 int main()
12 {
13     printf("Welcome ");
14     printf("to C!\n");
15
16     return 0;
17 }
18
```

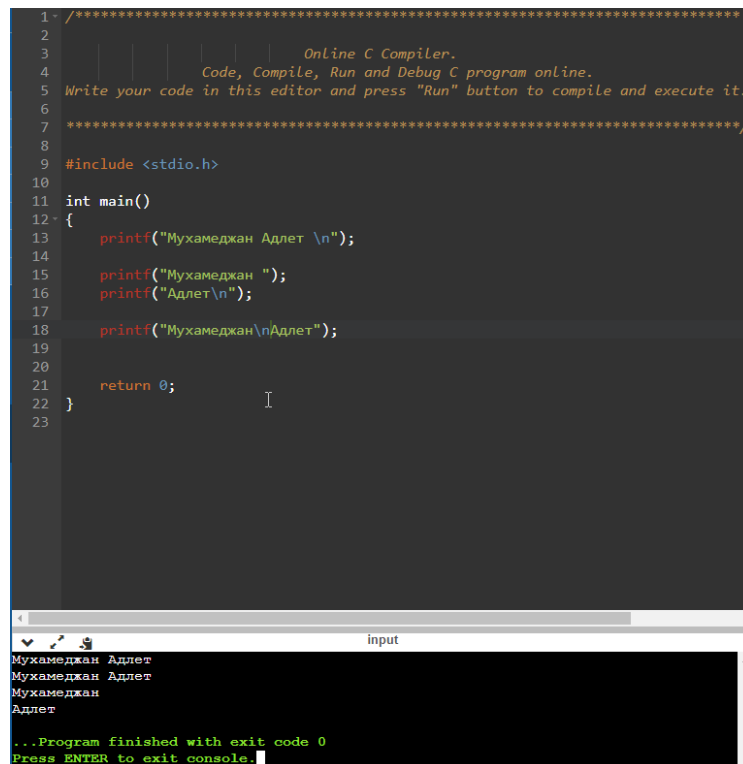
Рисунок 11 – Скриншот кода из примера 2

```
8
9  #include <stdio.h>
10
11 int main()
12 {
13     printf("Welcome\nto\nC!\n");
14
15     return 0;
16 }
```

Рисунок 12 – Скриншот кода из примера 3

Задание 3.2: Напишите программу, выводящую на печать вашу фамилию и имя тремя способами.

Для выполнения задания (рисунок 13) в одну строку, применив один оператор printf, использовалась среда разработки Microsoft Visual Studio и шаблон Консольное приложение.



```
1  /*****
2
3      Online C Compiler.
4      Code, Compile, Run and Debug C program online.
5      Write your code in this editor and press "Run" button to compile and execute it.
6
7      *****/
9  #include <stdio.h>
10
11  int main()
12  {
13      printf("Мухамеджан Адлет \n");
14
15      printf("Мухамеджан ");
16      printf("Адлет\n");
17
18      printf("Мухамеджан\nАдлет");
19
20
21      return 0;
22  }
23
```

input

Мухамеджан Адлет
Мухамеджан Адлет
Мухамеджан
Адлет

...Program finished with exit code 0
Press ENTER to exit console.

Рисунок 13 – Скриншот выполненного задания всеми тремя способами

Вывод: В результате выполнения лабораторной работы были получены навыки создания блок-схем и работы с интерфейсом Microsoft Visual Studio.