

Основы программирования

Лабораторная работа 1

Тема: Базовые типы данных, функции ввода-вывода на консоль,
линейная последовательность действий

Цель лабораторной работы

Целями данной работы являются изучение базовых типов данных, способов ввода-вывода, обучение написанию программ с линейной последовательностью действий. Также целью является обучение выполнению анализа, формализации задачи и оформления ее решения.

Краткие теоретические сведения

1. Программирование на C\C++

Главная конструкция *структурного программирования* на языке C\C++ — **функция**, являющаяся здесь единственным видом подпрограммы. Функции позволяют определить и отдельно закодировать различные задачи, решаемые программой, благодаря чему эта программа становится модульной.

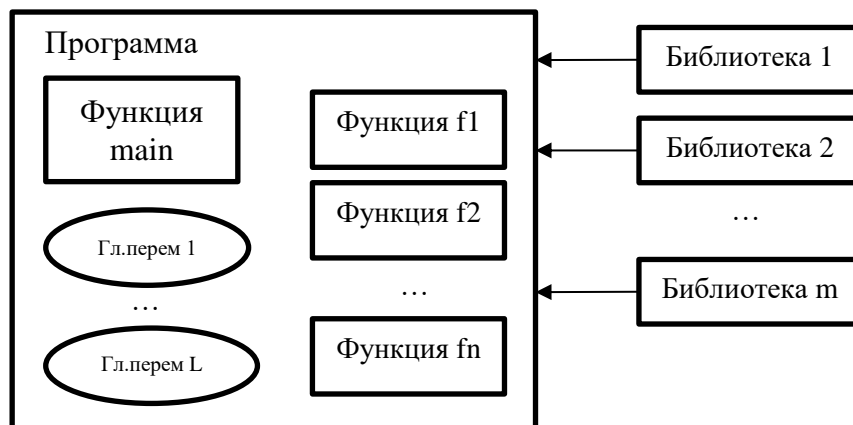


Рисунок 1 – Схема программы на C\C++ (структурное программирование)

Другой способ структурирования программы, написанной на языке C, заключается в использовании программных блоков.

Программный блок — это логически связанная группа операторов программы, которую можно рассматривать как отдельную программную единицу. В языке C блок представляет собой последовательность операторов программы, заключенную в фигурные скобки.

```
{  
    Программный блок  
}
```

В табл. 1 перечислены 32 ключевых слова, определенные стандартом C89. Они же являются ключевыми словами языка C как подмножества C++. В табл. 2 приведены ключевые слова, добавленные стандартом C99. Набор ключевых слов вместе с формальным синтаксисом C составляет язык программирования C.

Таблица 1 – Ключевые слова стандарта C89

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Таблица 2 – Ключевые слова, добавленные стандартом C99

_Bool	_Imaginary	restrict
_Complex	inline	

В программе ключевое слово может быть использовано только как ключевое слово, то есть никогда не допускается его использование в качестве переменной или имени функции.

2. Базовые типы данных

Тип данных – набор значений, хранящихся в переменных, а также набор операций, выполнение которых допускается над этими значениям.

Язык C\C++ имеет несколько встроенных типов данных. Условно их можно разделить на три группы:

- Числовые типы
- Символьные типы
- Логические типы

Числовые типы

Эти типы делятся на две группы:



У каждого типа есть свой диапазон значений, который зависит от размера типа:

Таблица 3 – Числовые типы данных

Тип данных	Диапазон	Размер в байтах
short	от -32 768 до 32 767	2
int	от -32 768 до 32 767 (при 2 байтах) или от -2 147 483 648 до 2 147 483 647 (при 4 байтах)	2 (x32) или 4 (x64)
long	от -2 147 483 648 до 2 147 483 647	4
long long	от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807	8
float	от +/- 3.4×10^{-38} до $3.4 \times 10^{+38}$	4
double	от +/- 1.7×10^{-308} до $1.7 \times 10^{+308}$	8
long double	диапазон зависит от размера	8 или больше

К некоторым типам применимы модификаторы. К примеру, следующие модификаторы применимы **только к целочисленным** значениям:

- *signed* – знаковый (по умолчанию, все числовые типы)
- *unsigned* – беззнаковый

Беззнаковые типы смещают диапазон значений в сторону положительных чисел, то есть увеличивают максимально возможное положительное значение в типе, **не меняя** размер типа:

Таблица 4 – Беззнаковые числовые типы данных

Тип данных	Диапазон	Размер в байтах
unsigned short	от 0 до 65 535	2
unsigned int	от 0 до 65 535 (при 2 байтах) или от 0 до 4 294 967 295 (при 4 байтах)	2 (x32) или 4 (x64)
unsigned long	от 0 до 4 294 967 295	4
unsigned long long	от 0 до 18 446 744 073 709 551 615	8

Символьные типы

Стандарт C89 был только один символьный тип – **char**. Он представляет один символ в кодировке **ASCII** и занимает в памяти 1 байт. То есть его диапазон от -128 до 127. На этот тип можно применить модификатор **unsigned**, что изменит его диапазон – от 0 до 255.

В современном C++ имеются также и другие символьные типы, которые значительно расширяют диапазон возможных символов:

Тип данных	Диапазон	Размер в байтах
wchar_t	от 0 до 65 535 (при 2 байтах) или от 0 до 4 294 967 295 (для 4 байт)	2 (<i>Windows</i>) или 4 (<i>Linux</i>)
char16_t	от 0 до 65 535	2
char32_t	от 0 до 4 294 967 295	4

Все три типа работают в более современной кодировке **Unicode**.

Логические типы

Логические типы в C++ представляет тип **bool**. Его размер 1 байт и он может хранить всего 2 значения: **true** (истина) и **false** (ложь). По сути это целые значения 1 и 0 соответственно.

3. Переменные

В языке C++ имена переменных, функций, меток и других объектов, определенных пользователем называются **идентификаторами**. **Идентификатор** представляет собой последовательность из одного или нескольких символов.

Переменная — представляет собой имя ячейки памяти, которую можно использовать для хранения модифицируемого значения. Все переменные должны быть объявлены до своего использования. Общая форма *объявления* имеет такой вид:

Тип_переменной имя_переменной;

Примеры:

```
char a;
int b, c;
float s = 1.5;
```

4. Создание проекта в Visual Studio

При запуске среды на экране первым отобразится окно создания/открытия проекта. В этом случае для создания нового проекта нажмем «Создание проекта».

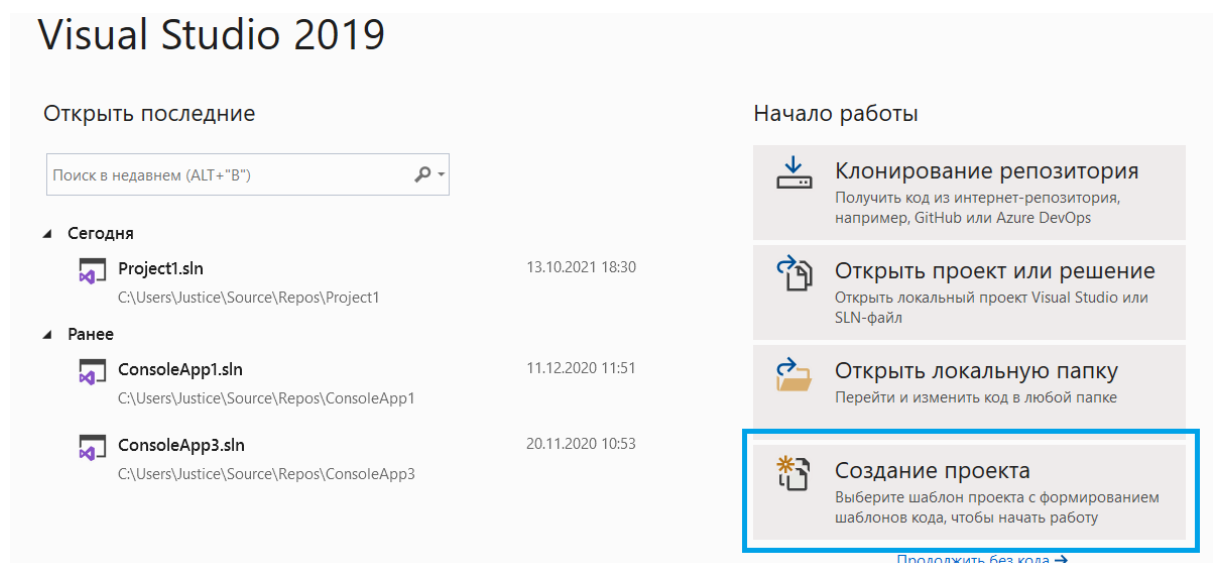


Рисунок 2 – Окно при запуске Visual Studio 2019

Вторым шагом будет выбор шаблона создаваемого проекта. Необходимо выбрать «Мастер классических приложений Windows (C++)».

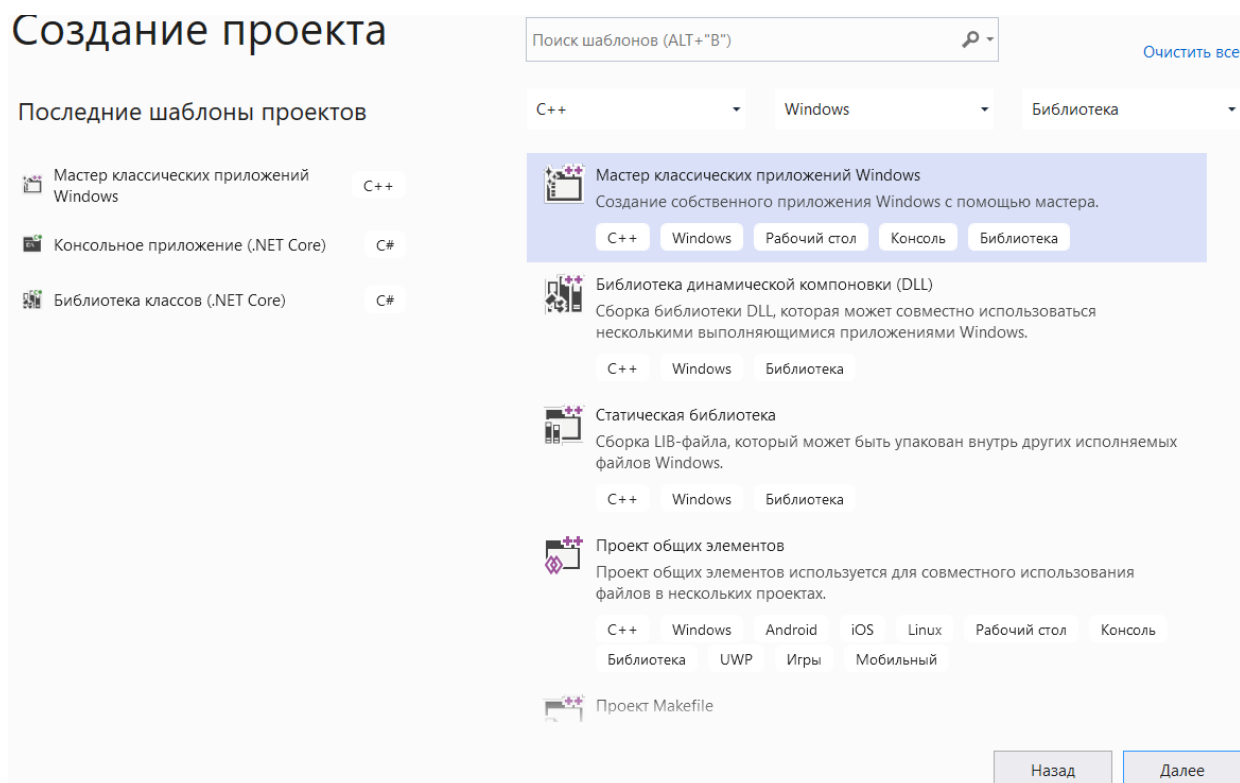
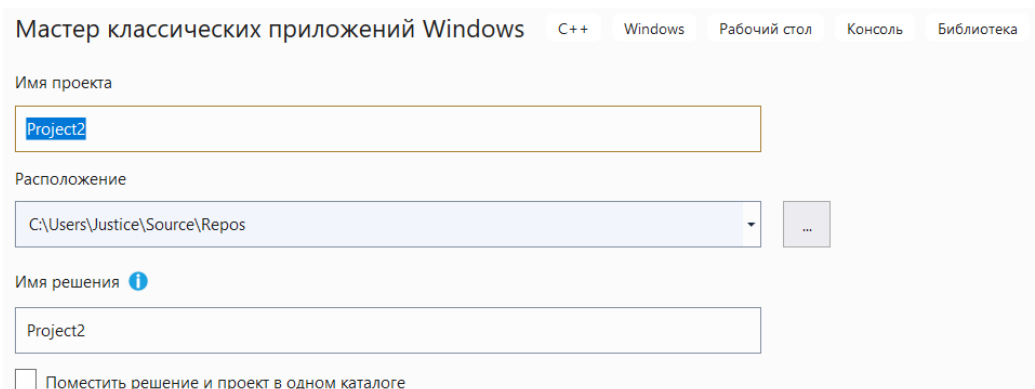


Рисунок 3 – Выбор мастера создания проектов Visual Studio

На следующем шаге можно настроить имя и расположение проекта. В случае создания первого приложения можно просто оставить настройки по умолчанию.



Далее необходимо будет выбрать тип приложения (Консольное приложение) и отметить пункт «Пустой проект», чтобы при создании проекта он не содержал в себе ничего лишнего.

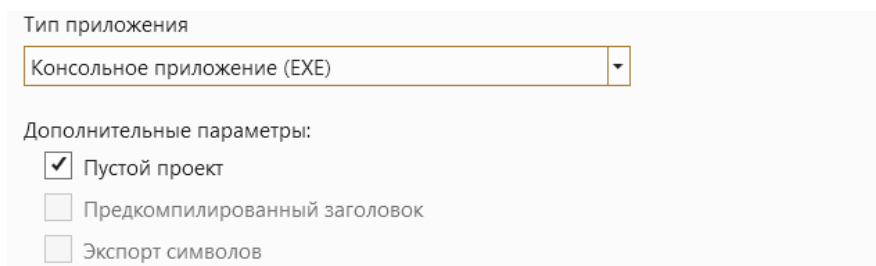


Рисунок 4 – Проект классического приложения Windows

После того как проект создан в левом (или правом) верхнем углу экрана появляется окно обозревателя решений (*solution explorer*). Основным предназначением этого окна является предоставление пользователю средств навигации в проекте.

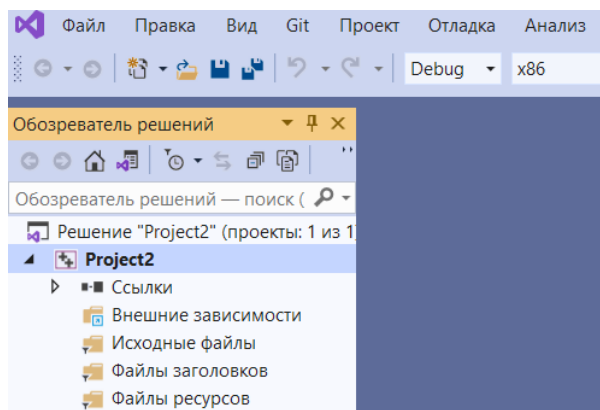


Рисунок 5 – Обозреватель решений

В окне обозревателя отображаются три папки, каждая служит для хранения файлов определенного типа:

Файлы заголовков (Header Files) – заголовочные файлы проекта. В этих файлах хранятся описания пользовательских типов (классов, структура и т.д.) с которыми вы познакомитесь далее в курсе.

Файлы ресурсов (Resource Files) – вспомогательные файлы проекта. Здесь может храниться достаточно большой перечень файлов: начиная от растровых изображений и заканчивая битовыми последовательностями. Эти файлы используются в большей степени в 32-х битных проектах и для выполнения лабораторных работ не потребуются.

Исходные файлы (Source Files) – файлы проекта (*.c/*.cpp). В этих файлах располагается программа.

Следующий шаг, который требуется сделать – открыть файл с исходным текстом программы. В случае пустого проекта в папке *Исходные файлы (Source Files)* такого файла нет. Для его создания щелкните правой клавишей мыши на папке *Исходные файлы* – появиться контекстное меню. Выберите пункт *Добавить* → *Создать элемент (new → new item)*.

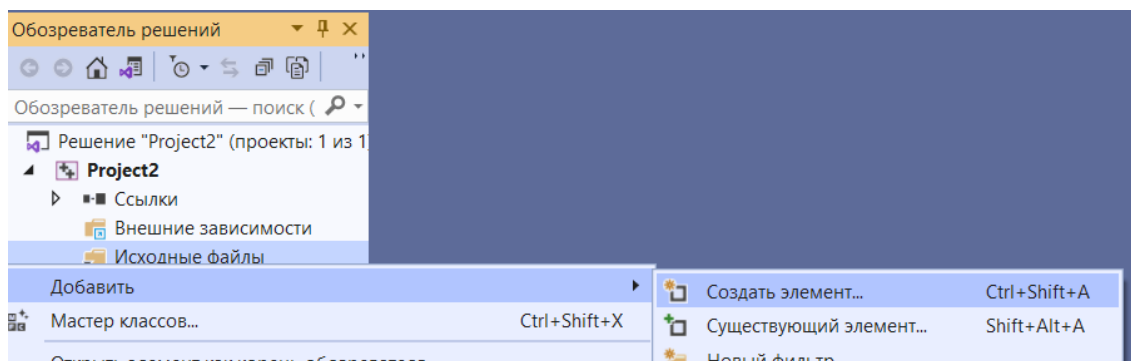


Рисунок 6 – Создание файла исходного кода

В появившемся диалоговом окне выберите файл с++, задайте имя файлу (к примеру, *Source.cpp*).

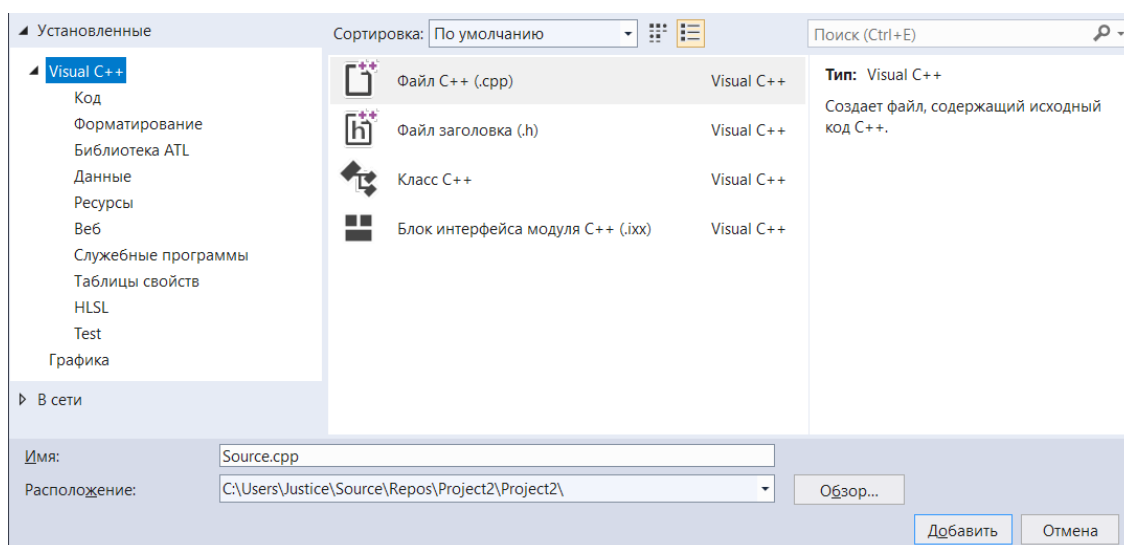


Рисунок 7 – Добавление новых исходных файлов в проект

После добавление нового файла он автоматически появиться в основном окне среды программирования Microsoft Visual Studio.

В текстовом редакторе наберите текст программы.

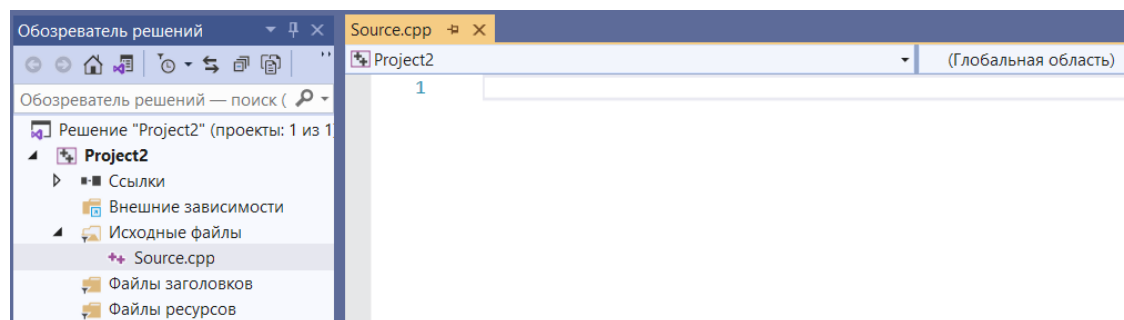


Рисунок 8 – Среда программирования Microsoft Visual Studio 2019

В начале программы объявляются библиотеки со стандартными функциями, которые будут использоваться в программе.

Для работы программы на C создается главная функция с именем `main`. Ключевое слово `void` означает, что функция `main` не возвращает значения.

Наберите текст следующей программы:

```
#include <stdio.h>
#include <conio.h>
int main() {
    int a = 1;
    int b;
    a = a + b;
    a = a + 10;
    a = a * b;
    printf_s("The value of first parametr - %i\n", a);
    printf_s("The value of second parametr - %i\n", b);
    printf_s("\nHello world");
    return 0;
}
```

Для компилирования программы нажмите клавишу **F5** (или просто нажать зеленую стрелку на верхней панели). В появившемся окне можно видеть результат работы программы.

Если в программе были **допущены синтаксические ошибки**, компилятор не даст скомпилировать программу, и выведет сообщение в окно с подробным описанием причины ошибки и строки, где она была допущена.

Откройте окно с сообщениями об ошибках – *Вид → Список ошибок (View → ErrorList* или *View → Other windows → ErrorList*).

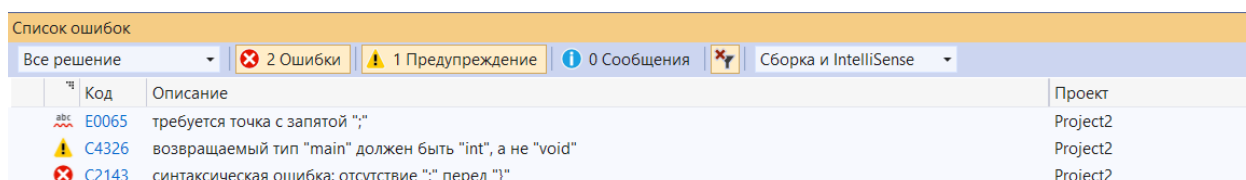


Рисунок 9 – Окно с сообщениями об ошибках в программе

Задание

Разработать программу, которая позволяет ввести с клавиатуры данные из таблицы, представленной в вашем варианте индивидуального задания, и выводит на экран таблицу, подобную той, которая находится в индивидуальном задании (включая заголовок и примечания, если есть).

При этом в каждой таблице есть **несколько** вычисляемых полей (подчеркнуты). Для получения хранения данных в программе используйте базовые типы данных (возможно с модификаторами).

Методические указания

Выполните индивидуальные задания.

При выполнении индивидуальных заданий необходимо выполнить все этапы решения задачи на ЭВМ:

- 1) словесная постановка задачи;
- 2) анализ задачи и формальная постановка задачи;
- 3) проектирование (разработка алгоритма) – лучше всего в графической форме (блок-схема), но можно и в словесной форме или псевдокод;
- 4) реализация (кодирование, отладка);
- 5) тестирование.

Результаты выполнения индивидуальных заданий оформить в виде отчета.

Пример решения варианта

1. Постановка задачи

Разработать программу, которая позволяет ввести с клавиатуры данные из таблицы, представленной в вашем варианте индивидуального задания, и выводит на экран таблицу, подобную той, которая находится в индивидуальном задании (включая заголовок и примечания, если есть).

При этом в каждой таблице есть **несколько** вычисляемых полей (подчеркнуты). Для получения хранения данных в программе используйте базовые типы данных (возможно с модификаторами).

Буддийские монастыри Японии периода Нара			
Год открытия	Школа	Количество монахов	Площадь земли (га)
728	Т	220	368.8
718	С	50	54.7
747	Д	10	12.2
Среднее кол. монахов		93	
Общая площадь земли			435.7
Примечание: Т - Тэндай; С - Сингон; Д - Дзедзицу			

2. Анализ задачи и формальная постановка задачи

2.1 Анализ задачи

В таблице, приведенной в индивидуальном задании, представлены следующие данные:

- 1) первая строка (общий заголовок таблицы), вторая строка (заголовки столбцов таблицы) и восьмая строка (примечание) – константные (постоянные, неизменяемые) текстовые строки;
- 2) фактические данные составляют третью, четвертую и пятую строки;
- 3) шестая и седьмая строки – вычисляемые поля на основе фактических данных.

Программа должна начинаться с ввода фактических данных. Для удобства пользователя программы перед вводом на экран должно выводиться приглашение, в котором перечисляется, какие именно данные ожидает программа.

За одно приглашение-ввод будет вводиться одна строка фактических данных таблицы. Т.е. приглашение ввода должно повторяться трижды. Введенные данные будут сохраняться в переменных программы.

На втором этапе программа выполняет подсчет вычисляемых полей. В данном случае это поля «Среднее кол. монахов», которое основывается на вводимых данных о количестве монахов, и «Общая площадь земли», которое основывается на вводимых данных о площади земли.

На третьем этапе программа выполняет вывод данных на экран. Среди строк, которые выводятся, будут строки без переменных значений (заголовки и линии) и строки с переменными значениями (строки с фактическими данными).

Для формирования наглядного табличного вида можно применять символы '|' и '-'.

Определение переменных является первой и самой главной задачей нашей программы. Для каждой переменной нужно подобрать такой тип, который бы обеспечивал представление в памяти ее значения с достаточным диапазоном и достаточной точностью.

Формальная постановка задачи может быть определена следующим образом.

2.2 Формальная постановка задачи

Входные данные:

g1, g2, g3 – целые числа, представляющие год открытия;

sch1, sch2, sch3 – символы, представляющие названия школ монастырей;

cnt1, cnt2, cnt3 – количество монахов в каждом монастыре;

sq1, sq2, sq3 – вещественные числа, представляющие площадь земель монастырей.

Выходные данные:

Таблица с введенной информацией о монастырях:

Год: Для представления этих данных в памяти переменные, в которых хранятся значения этого столбца, должны быть и объявлены как:

```
int g1, g2, g3;
```

Школа: Эти данные являются одним символом. Их объявление:

```
char sch1, sch2, sch3;
```

Количество монахов: Эти данные являются целыми положительными числами. Т.е. объявление соответствующих переменных:

```
unsigned int cnt1, cnt2, cnt3;
```

Площадь земель: Это число с дробной частью, следовательно, соответствующие переменные должны быть переменными с плавающей точкой. Точность данных – один знак после точки, поэтому для их представления будет достаточно типа **float**:

```
float sq1, sq2, sq3;
```

Среднее кол. монахов: В данном случае нам необходимо получить целое число (хотя среднее значение чаще вещественное число). Поэтому объявляем переменную следующим образом:

```
unsigned int mid_cnt;
```

Общая площадь земли: Точность данных такая же как у полей «Площадь земель» – один знак после точки, поэтому для их представления будет также достаточно типа **float**:

```
float sum_sq;
```

Метод получения значений вычисляемых полей:

Среднее кол. монахов:

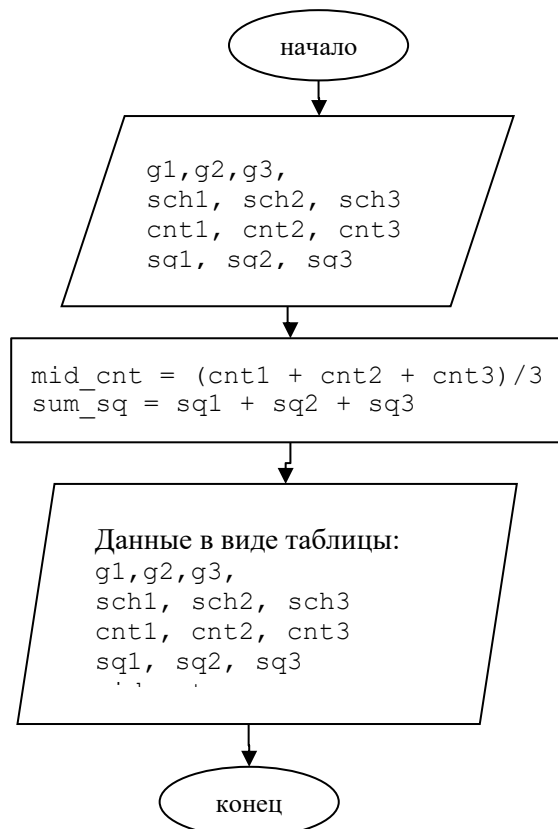
$\text{mid_cnt} = (\text{cnt1} + \text{cnt2} + \text{cnt3}) / 3$ – так как все участники деления целые числа, то и результат будет в виде целого числа, т.е. дробная часть просто отбросится.

Общая площадь земли:

$\text{sum_sq} = \text{sq1} + \text{sq2} + \text{sq3}$ – достаточно просто сложить все три площади вместе.

3. Алгоритм

Алгоритм программы линейный.



4. Разработка текста программы (кодирование)

Начинаем разработку текста программы с заголовка главной функции ***main()***:

```
int main(void)
```

Далее открывается тело функции и в нее включается описание переменных.

Кодовая часть программы начинается с приглашения - вывода строки-константы:

```
printf("1) Введите: год, школу, количество, площадь (через пробел):\n");
```

за которым считываются данные, вводимые оператором:

```
scanf_s("%d %c %d %f", &g1, &sch1, &cnt1, &sq1);
```

Поскольку описания этих функций хранятся в файле ***stdio.h***, включаем этот файл в начало программы:

```
#include <stdio.h>
```

При вводе строка вводится по спецификации типа ***%s***, один символ - по спецификации типа ***%c***, целое число - по спецификации типа ***%d***, число с плавающей точкой - по

спецификации типа `%f`. Спецификации разделяются пробелами, т.е. и данные при вводе должны разделяться пробелами или переходом на новую строку. В списке ввода перед всеми элементами ставится знак `&`.

Приглашение-ввод повторяется трижды, с разными переменными в списке ввода.

Для формирования вывода следует подсчитать ширину каждого столбца.

Первая строка - горизонтальная линия, которая состоит из символа '-', повторенного 47 раз. Вторая строка - общий заголовок, дополненный до ширины 47 пробелами. Третья строка - еще одна горизонтальная линия. Четвертая и пятая строки - заголовки столбцов, каждый из них имеет установленную ширину. Шестая строка - еще одна горизонтальная линия.

Наконец, строки с седьмой до девятой - фактические данные. В каждой строке выводятся значения набора переменных для одной строки. Например:

```
printf("|%9d |%6c |%10d | %12.1f|\n", g1, sc1, cnt1, sq1);
```

Значения года выводятся по спецификации типа `%d` с шириной 9. Значения школы выводятся по спецификации типа `%c` с шириной 6 (хоть `char` всегда занимает 1 символ). Значения количества выводятся по спецификации типа `%d` с шириной 10, а площадь – `%f` с общей шириной 12 и одним знаком после точки. Значения в этих столбцах дополняются до нужной ширины пробелами. В столбце 3 ставится признак выравнивания по правому краю.

Значения среднего количества монахов (`%d`) и общей площади земли (`%f`) выводятся в 3-ем и 4-ом столбце для наглядности:

```
printf("|Среднее кол. монахов%10d |%\n", mid_cnt);  
printf("|Общая площадь земли %12.1f |%\n", sum_sq);
```

Остальные четыре строки выводятся как текстовые константы.

Полный текст программы приведен ниже.

```
#include <stdio.h>  
  
int main() {  
    /* Объявление переменных */  
    int g1, g2, g3;  
    char sc1, sc2, sc3;  
    unsigned int cnt1, cnt2, cnt3;  
    float sq1, sq2, sq3;  
  
    // Вычисляемые поля  
    unsigned int mid_cnt;  
    float sum_sq;  
  
    /* Введение фактических данных*/  
    printf("1) Введите: год, школу, количество, площадь (через пробел):\n ");  
    scanf_s("%d %c %d %f",&g1, &sc1,1, &cnt1, &sq1);  
    printf("2) Введите: год, школу, количество, площадь (через пробел):\n ");  
    scanf_s("%d %c %d %f",&g2, &sc2,1, &cnt2, &sq2);  
    printf("3) Введите: год, школу, количество, площадь (через пробел):\n ");  
    scanf_s("%d %c %d %f",&g3, &sc3,1, &cnt3, &sq3);  
  
    /* Подсчет вычисляемых полей */  
    mid_cnt = (cnt1 + cnt2 + cnt3)/3;
```

```

sum_sq = sq1 + sq2 + sq3;
/* Вывод таблицы */
/* вывод заголовков */
printf("-----\n");
printf("|Буддийские монастыри Японии периода Нара |\n");
printf("-----|\n");
printf("| Год          | Школа | Количество|Площадь земли|\n");
printf("|              |       | монахов  | (га)        |\n");
printf("|-----|-----|-----|-----|\n");
/* вывод строк фактических данных */
printf("|%9d |%6c |%10d | %12.1f|\n", g1, sc1, cnt1, sq1);
printf("|%9d |%6c |%10d | %12.1f|\n", g2, sc2, cnt2, sq2);
printf("|%9d |%6c |%10d | %12.1f|\n", g3, sc3, cnt3, sq3);
/* вывод вычисляемых полей */
printf("-----|\n");
printf("|Среднее кол. монахов%10d |\n", mid_cnt);
printf("|Общая площадь земли %12.1f |\n", sum_sq);
/* вывод примечаний */
printf("-----|\n");
printf("| Примечание: Т - Тэндай; С - Сингон; |\n");
printf("|              Д - Дзедзицу |\n");
printf("-----|\n");

return 0;
}

```

5. Отладка программы

При отладке программы можно использовать пошаговый режим отладки с отслеживанием значений переменных – тех, которые вводятся. Если возникнут проблемы с вводом переменных, есть смысл вводить каждую переменную отдельным оператором.

6. Тестирование

При работе программы на экран было выдано следующее:

```

1) Введите: год, школу, количество, площадь (через пробел):
   728 Т 220 368.8
2) Введите: год, школу, количество, площадь (через пробел):
   718 С 50 54.7
3) Введите: год, школу, количество, площадь (через пробел):
   747 Д 10 12.2
-----
|Буддийские монастыри Японии периода Нара |
|-----|
| Год          | Школа | Количество|Площадь земли|
|              |       | монахов  | (га)        |
|-----|-----|-----|-----|
|      728     | Т     |      220  |    368.8    |
|      718     | С     |       50  |     54.7    |
|      747     | Д     |       10  |     12.2    |
|-----|-----|-----|-----|
|Среднее кол. монахов          93|
|Общая площадь земли          435.7|
|-----|-----|-----|-----|
| Примечание: Т - Тэндай; С - Сингон; |
|              Д - Дзедзицу          |
|-----|-----|-----|-----|
Для продолжения нажмите любую клавишу . . .

```

Также в отчете можно просто предоставлять скриншоты тестирования. При необходимости они должны сопровождаться пояснениями.

Варианты индивидуальных заданий

Вариант 1

Прайс-лист			
Номер товара	Категория	Цена (руб)	Срок поставки (дн)
11	К	1128.00	8
12	С	170.50	11
13	М	1735.50	10
<u>Общая цена</u>		3034.00	
<u>Средний срок</u>			9
Примечание: К - компьютеры, С - комплектующие, М - множительная техника			

Вариант 2

Фирмы			
Номер фирмы	Количество продуктов	Объем продажи (млн. \$)	Часть рынка (%)
1	3	248.80	21.1
2	5	239.25	19.9
3	2	104.82	13.1
<u>Общее количество продуктов</u>	10		
<u>Общий объем продаж</u>		592.87	
<u>Суммарная часть рынка</u>			54.1

Вариант 3

Отдел кадров			
Номер сотрудника	Должность	Возраст (лет)	Оклад (тыс. руб.)
1	М	40	51.50
2	С	35	25.65
3	И	25	30.00
<u>Средняя дата рождения</u>		30	
<u>Средний оклад</u>			35.71
Примечание: М - менеджер, С - секретарь, И - инженер			

Вариант 4

Ведомость деталей				
Номер детали	Тип	Количество	Вес 1 детали (г)	<u>Общий вес (г)</u>
102	З	3	450	1350
133	П	8	74	1942
212	О	1	1170	1170
<u>Общее количество</u>		12		
Примечание: О - оригинальная, П - покупная, З - заимствованная				

Вариант 5

Характеристики ПЭВМ			
Код процессора	Частота (Mgz)	RAM (Gb)	Тип
444	233	1.0	C
122	166	2.5	C
333	2000	4	R
<u>Средняя частота</u>	799		
<u>Средняя RAM</u>		2.5	
Тип: C - CISC-процессор, R - RISC-процессор			

Вариант 6

Ведомость комплектующих				
Код	Тип	Номинал	Количество	Вес (г)
1124	R	10000	12	20
1124	R	50000	10	40
1200	C	17.5	23	30
<u>Общее количество</u>			45	
<u>Средний вес</u>				30
Примечание: R - резистор; C - конденсатор				

Вариант 7

Офисные пакеты			
Номер	Производитель	Количество сост. частей	Цена (\$)
1	M	4	87.5
2	L	5	102.0
3	S	3	19.8
<u>Среднее количество частей</u>		3	
<u>Средняя цена</u>			69.7
Примечание: M – Microsoft, L – Lotus, S - Sun			

Вариант 8

Ведомость общественного транспорта			
Вид транспорта	N маршрута	Протяженность маршрута (км)	Время в дороге (мин)
Т	12	27.55	75
М	17	13.60	57
А	12	57.30	117
<u>Общая протяженность</u>		98.45	
<u>Среднее время</u>			83
Примечание: Т - трамвай, М – маршрутное такси, А - автобус			

Вариант 9

Приблизительное количество звезд разных спектральных классов в Галактике			
Спектральный класс	Приблизительная масса (отн. Солнца)	Часть %	Численность (млн)
О	32.00	0.00002	0,055
F	1.25	2.90000	12000,000
М	0.20	73.20000	293000,000
<u>Средняя масса</u>	11.15		
<u>Средняя численность</u>			101666,685

Вариант 10

Время выполнения некоторых программ, которые применяют параллельные алгоритмы			
Название прграммы	Количество строк кода	Время исп. на SGI Challenge (сек)	Время исп. на SGI Indy (сек)
О	1917	8.70	18.20
D	3386	6.95	22.00
Т	417	1.05	2.98
<u>Общее количество строк кода</u>	5720		
<u>Общее среднее время исполнения</u>		9.98	
Примечание: О - OCEAN, D – DYFESM, Т - TRFD			

Вариант 11

Коэффициенты теплопроводимости материалов			
Вещество	Тип	Влажность (%)	Коэффициент
А	М	70	209.300
С	Т	89	0.035
Г	Д	17	0.730
<u>Средняя влажность</u>		58.6	
<u>Средний коэффициент</u>			70.021
Примечание: А – Алюминий, С – Стекловата, Г – Глина, Ж – Железо, П – Пенопласт. М - металлы, Т - термоизоляционные материалы, Д - другие материалы.			

Вариант 12

Сельскохозяйственные культуры			
Культура	Тип	Посевная площадь (га)	Урожайность (ц/га)
С	Б	13000	45
Ч	З	8000	17
Р	З	25650	24
<u>Общая площадь</u>		46650	
<u>Средняя урожайность</u>			28,6
Примечание: С – Соя, Ч – Чумиза, Р – Рис, Г – Горох, Ф – Фасоль. З - зерновые, Б – бобовые.			

Вариант 13

Ведомости о присвоении квалификации			
Номер	Год рождения	Оценка результатов экзаменов	Присвоен уровень квалификации
1	1970	4.3	6
2	1975	5.2	5
3	1966	3.1	3
<u>Средняя оценка</u>		4.2	
<u>Средний уровень квалификации</u>			4.6
Примечание: показана итоговая оценка по результатам экзаменов по теоретической и практической подготовке			

Вариант 14

Журнал регистрации аварий:			
Код предприятия	Категория	Убытки от аварии (тыс.руб)	Срок простоя (дн)
П	1	12.7	14
Т	1	0.71	5
А	2	177.15	25
<u>Общие убытки</u>		190.56	
<u>Средний срок простоя</u>			14.6
Примечание: П – Птицефабрика, Т – Теплоцентраль, А – Автосервис			