

## Глава 2. Константы и переменные

Любое действие в программе происходит с данными, которые в программировании называются операндами. В примере с бутербродом роль операндов выполняли ингредиенты и приготовленный бутерброд. Каждый операнд может быть как переменной величиной, так и величиной постоянной (константой), выражением (например, суммой двух переменных) или функцией.

Понятно, что каждый из операндов (данных) отличается от другого своими свойствами, видом, вкусом. В программировании принято говорить, что каждый операнд имеет свой тип данных. Данные разных типов по-разному хранятся в памяти компьютера, по-разному обрабатываются, отличаются разным набором операций и функций, разрешенных для каждого типа, а также отличаются множеством значений, разрешенных для величин каждого типа.

### 2.1. Переменные

Переменная – это именованная область памяти, в которой хранятся данные определенного типа. Например, три коробки: «Овощи», «Мясо» и «Напитки» иллюстрируют, как выглядит область памяти компьютера, которая готова хранить данные определенного типа. Имя коробки служит для обращения к области памяти (например, к коробке с именем «Овощи»), в которой хранится значение переменной (например, «Огурец»). Невозможно положить огурец в коробку с именем «Напитки», но его можно положить в коробку с именем «Овощи» и тогда огурец займет все пространство коробки, а область памяти с именем «Овощи» будет не пустой. Если захочется положить в коробку помидор, то придется достать оттуда огурец и заменить его новым овощем.

*Таким образом, переменная может содержать только одно значение! При попытке записать в переменную что-то новое, старое значение не сохраняется, а заменяется новым.*

В примере по приготовлению бутерброда роль переменных величин выполняют два ингредиента из имеющихся в холодильнике (это могут быть сыр, салат, ветчина и другие) – «Ингредиент 1» и «Ингредиент 2». Величина с именем «Хлеб» должна быть в составе любого бутерброда, поэтому она не является переменной величиной.

Таким образом, представить алгоритм приготовления бутерброда можно так: ячейка памяти с именем «Хлеб» заполнена постоянным значением, две пустые ячейки памяти с именами «Ингредиент 1» и «Ингредиент 2» заполняются любыми двумя wybranными продуктами из имеющихся, то есть переменные «Ингредиент 1» и «Ингредиент 2» получают значения (данные). Третья переменная, назовем ее именем «Бутерброд», будет принимать сложение значений из переменных «Ингредиент 1», «Ингредиент 2» и значения из области памяти с именем «Хлеб».

Понятно, что во время выполнения алгоритма приготовления бутерброда значение таких типов, как «переменные» можно изменять.

Перед использованием любой переменной в программе необходимо зарезервировать для нее область памяти компьютера. Это действие называется описанием переменной.

## **2.2. Константы**

Кроме переменных величин, которые могут изменять свои значения во время выполнения программы, существуют неизменяемые величины. Такие величины называются константами. В примере с бутербродом такой константой является хлеб. Точно известно, что он используется в приготовлении любого бутерброда.

Константы также участвуют в действиях, но результат операции нельзя поместить область памяти, где хранится константа в течение программы

Математические константы часто используют в программах, например, число  $\pi \approx 3,14$ . Изменять это значение в программе никому и в голову не придет.

### **Задание для самостоятельной работы:**

- Напишите словесный алгоритм приготовления яичницы. Яйцо – константа, еще два ингредиента – переменные. Постройте блок-схему алгоритма.
- Составьте словесный алгоритм перехода дороги на пешеходном переходе по сигналу светофора. Постройте блок-схему алгоритма.
- Составьте словесный алгоритм перехода дороги на пешеходном переходе без светофора. Постройте блок-схему алгоритма.

## Глава 3. Базовые средства C++

### 3.1. Идентификаторы

Идентификатор – это имя любого объекта в программе. Имена присваиваются переменным, константам, функциям и другим объектам. Имя может состоять из латинских букв (прописные и строчные здесь различаются), цифр и знаков подчеркивания. Имя может начинаться только с буквы или знака подчеркивания. Имена Bread, bread, brEad различны.

Лучше всего использовать имена, которые передают смысл объекта. Длина идентификаторов не ограничена, но чем короче имя, там легче читается программа. Присвоение имени происходит на этапе резервирования объекта.

Например, при приготовлении бутерброда для двух переменных «Ингредиент 1» и «Ингредиент 2» и константы «Хлеб» можно использовать имена: ingr1, ingr2, bread. Далее в программе следует использовать эти идентификаторы.

### 3.2. Комментарии

В сложных программах и алгоритмах часто приходится делать пояснения, чтобы любой пользователь смог быстро понять смысл любого фрагмента программного кода. Такие пояснения называются комментариями.

Если комментарии пишутся в одну строку, то ставится символ //, после которого следует текст комментария. Текст программа не воспринимает, он существует только для пользователя.

Если комментарий большой и размещен в несколько строк, то для того, чтобы не писать в начале каждой строки символ //, в начале комментария ставится символ /\*, а в конце \*/.

Например:

```
//Это комментарий  
sandwich = ingr1 + ingr2 + bread; // И это комментарий  
/* и все...  
что написано...  
здесь...  
тоже комментарий. */
```

### 3.3. Представление комментариев в блок-схемах алгоритмов

Как и в коде программы, в блок-схемах используют комментарии.

Для этого существует специальное изображение:



Рисунок 3.1 – Комментарии в блок-схемах алгоритмов

Блок комментариев соединяется с другими блоками алгоритма пунктирной линией

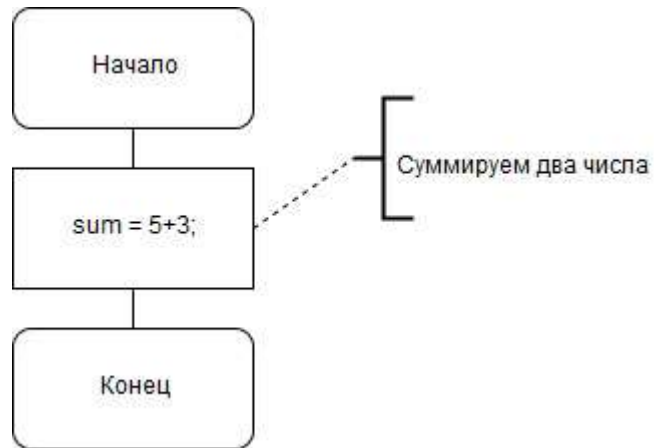


Рисунок 3.2 – Пример блок-схемы с комментарием

## Глава 4. Типы данных в C++

От типа данных зависит, сколько памяти будет выделено под данные, как они будут храниться в памяти компьютера, операции, которые разрешено применять. Типы данных можно разделить на основные и составные. Существует шесть основных типов данных, на основе которых можно создавать составные типы.

### 4.1. Основные типы данных

- `char` – символьный тип, в котором используется 256 различных символов, каждому символу ставится в соответствие целое число – код символа, переменная типа `char` также может хранить целочисленные значения из диапазона от 0 до 255
- `bool` – логический тип, может иметь только два значения: `true` (истина) и `false` (ложь);
- `short int` - целые числа, диапазон изменения которых составляет от -32 768 до 32 767;
- `int` – целые числа, диапазон изменения которых составляет от -2 147 483 648 до 2 147 483 647;
- `float` – вещественные числа, диапазон изменения которых составляет от  $3.4e-38$  до  $3.4e38$ ;
- `double` – вещественные числа, диапазон изменения которых составляет от  $1.7e-308$  до  $1.7e308$ .

### 4.2. Объявление типа переменной величины

Тип указывается при объявлении переменной:

<тип> <имя> = значение.

Например,

```
int B = 5;
```

Объявлена переменная целого типа с именем `B`, которой присвоено значение, равное 5. Если необходимо просто зарезервировать память под переменную, то есть создать пустую переменную, то её значение не указывается.

Например:

```
double C;
```

### 4.3. Спецификаторы типов

Спецификаторы типов используются для изменения диапазона значений стандартных типов:

- `signed` - знаковый
- `unsigned` – беззнаковый

Например:

```
int a;           // Имеет диапазон от -2 147 483 648 до 2 147 483 647
signed int b;   // Имеет диапазон от -2 147 483 648 до 2 147 483 647
unsigned int c; /* Имеет диапазон от 0 до 4 294 967 295 – это сумма
                модулей концов диапазона в типе со знаком,
                количество чисел не изменилось*/
```

#### Задание для самостоятельного решения:

- Какой диапазон значений будет у беззнакового типа short int?
- Как будут представлены данные типа signed char?
- Будет ли выдана ошибка при выполнении оператора char a=15?

#### 4.4. Объявление констант

Константы создаются с помощью ключевого слова const:

```
const <тип> <имя> = значение.
```

Например:

```
const float pi = 3.14;
```

#### 4.5. Выражения в C++

Знак “=” в языке C++ обозначает операцию присваивания, когда в переменную слева от знака присваивания записывается результат того, что находится справа.

Справа может находиться:

- Переменная, когда в область памяти левой переменной запишется значение правой;
- Выражение, когда в область памяти левой переменной записывается результат выполнения действий, указанных в выражении.

Например:

```
const int B = 3; /* Объявлена константа целого типа с
                  именем B и в ее область памяти
                  занесено значение 3 */
const int C = 5; /* Объявлена константа целого типа с
                  именем C и в ее
                  область памяти занесено значение 5*/
int sum;        /* Резервирована память для хранения
                  значения переменной целого типа с
                  именем sum*/
sum = B + C;    /* В область памяти переменной с именем
                  sum записана
                  сумма двух констант*/
```

В результате выполненных действий в переменной sum будет храниться число 8.

**Это важно!**

- При создании нескольких переменных одного типа удобнее использовать перечисление:

Например:

```
int a = 3, int b = 5, int c, int d = 4;
```

Значения присваиваются отдельно каждой переменной. Некоторым из них можно значение не присваивать, оставить пустыми.

- В символьные переменные значения записываются или в одинарных кавычках или в виде числа, которое соответствует данному символу в кодовой таблице ASCII, где все символы имеют свой уникальный номер.

Например:

```
char bykva1 = 'a', bykva2 = 'M', symbol = 13;
```

В последней переменной хранится символ, который находится под номером 13 в таблице ASCII. Прописные и строчные буквы имеют разные номера!

- Если переменной целого типа присвоить дробное значение, то дробная часть будет автоматически отброшена.
- Дробная часть числа отделяется от целой части точкой: `double Pi = 3.14`.
- Каждое действие в программе на алгоритмическом языке C++ заканчивается точкой с запятой.
- При использовании операции присваивания возможно использование записи такого выражения:

```
A = B - A;
```

Например:

```
int A = 3, B = 5;  
A = B - A;
```

В переменной A будет сохранено число 2, как результат выполнения действия вычитания из значения переменной B, равной 5, значения переменной A, равной 3.

### **Задания для самостоятельного решения:**

- Создайте две переменных и две константы дробного типа. В первую переменную занесите сумму констант, во вторую их разность.
- Создайте две дробных переменных и занесите в одну из них разность их значений.

#### 4.6. Объявление переменных в блок-схемах.

Резервирование памяти (объявление) для переменной происходит как действие, поэтому для него используется блок в виде прямоугольника.

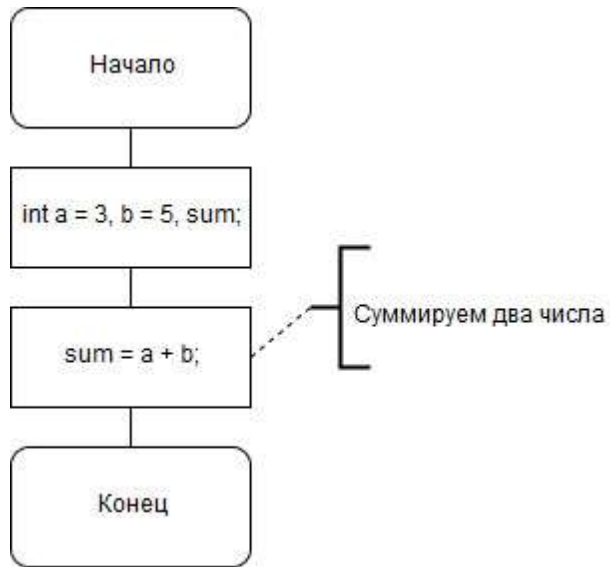


Рисунок 4.1 – Блок-схема с примером резервирования памяти