

-  – ОБЯЗАТЕЛЬНО
  – ДОПУСТИМО
  – НЕЖЕЛАТЕЛЬНО
 – ЖЕЛАТЕЛЬНО
  – ЗАПРЕЩЕНО

Правила компоновки

-  Программа собирается из модулей, которые могут быть описаны в файлах двух типов:
- заголовочные файлы (с расширением `.h`) который содержит объявление и определение одного класса
 - файлы с реализацией функций членов класса (с расширением `.cpp`).
- При этом пара файлов имеет одинаковые имена, назначенные по имени класса, но разные расширения.
-
-  Использование глобальных данных и функций крайне нежелательно.
-
-  Заголовочный файл должен начинаться с комментированного заголовка, имеющего следующую структуру:
- ```
// filename.h
// ФИО
//
// Краткое описание содержимого файла
```
- 
-  Файл с реализацией функций-членов классов должен начинаться с комментированного заголовка, имеющего следующую структуру:
- ```
// filename.cpp
// ФИО
```
-
-  Использование директив препроцессору, кроме `#include` для подключения заголовочных файлов и `#define` для определения констант, запрещается.
-
-  Длина строки в тексте программы ограничивается 80 символами. Для длинных строк использовать логические переносы с выравниванием.
-
-  Для структурирования текста программы обязательно использовать табуляцию. Величина табуляции должна составлять 4 пробела. Следить за тем чтобы в качестве табуляции использовался именно символ табуляции, а не 4 пробельных символа.

Комментарии

-  Комментарии должны составлять примерно 30% текста. Желательно комментировать каждый блок операторов заключенный в `{...}`;
-
-  Обязательно комментировать объявление классов, шаблонов, структур и их членов, функций и их аргументов, переменных.
-
-  Использование однострочного комментария заканчивающего строку допускается при объявлении переменных и других подобных случаях. Например:
- ```
int count; //целочисленный счетчик
```
- 
-  Многострочный комментарий можно использовать для комментирования лишнего или неотлаженного блока операторов таким образом, чтобы символы `/*` и `*/` были расположены точно друг под другом (как открывающая и закрывающая скобки) со сдвигом влево на величину табуляции:
- ```
/*
    i++;
*/
```

Идентификаторы

-  Для имен идентификаторов обязательно использовать символы нижнего регистра.
-  Для имен идентификаторов желательно использовать как можно более короткие англоязычные слова, например: `count`, `row`, `area`, `top` и т.п. Для составных имен использовать символ `'_'`.
-
-  Объявление переменных осуществляется в начале логического блока операторов.
-
-  Объявление переменной цикла для пошаговой конструкции `for` внутри конструкции `for` допустимо, если она используется только в этом цикле.
-
-  Использование `typedef` для введения более коротких и понятных синонимов типов данных допустимо.
-
-  Следует избегать идентификаторов с одинаковыми именами в различных областях видимости.
-
-  Неявное преобразование типов переменных недопустимо.

Выражения и операторы

-  На одной строке не желательно использование более чем одного оператора, т.е. более одного символа `'='` и более одного символа `','`
-
-  Следует избегать сложных арифметических выражений. В арифметических и логических выражениях желательна явная расстановка скобок `()`.
-
-  Автоинкремент и автодекремент использовать в арифметических выражениях запрещается.
-
-  Использование бесполезных и опасных ключевых слов и связанных с ними конструкций `auto`, `goto`, `register`, `signed`, `union` запрещается.

Условные конструкции

-  В конструкции `if-else` желательно заключение оператора (даже если он один) в скобки:
- ```
if (i<0)
{
 i--;
}
```

## Циклические конструкции

-  Для организации цикла с предусловием используется следующая конструкция:
- ```
while (i<=10)
{
    sum+=i;
}
```
-
-  Для организации цикла с постусловием используется следующая конструкция:

```
do
{
    sum+=i;
}while (i<=10);
```

Для организации пошагового цикла с предусловием используется следующая конструкция:

```
for (i=0; i<max; i++)
{
    sum+=i;
}
```

В конструкции `for ()` круглые скобки используются только для того, чтобы инициализировать, модифицировать переменную цикла и задать условие для окончания цикла. Использование оператора `,` в данной конструкции нежелательно.

Для бессмысленных переменных вложенных циклов использовать имена в следующем порядке: `i, j, k, l, m, n`

Организация бесконечного цикла нежелательна, но при необходимости используется конструкция на основе цикла с предусловием:

```
while (true)
{
    break;
}
```

Селективные конструкции

При небольшом количестве возможных вариантов предпочтительнее организация селективной конструкции на основе конструкции `if-else if`:

```
if (i>0)
{
    flag=1;
}
else if (i<0)
{
    flag=-1;
}
else
{
    flag=0;
}
```

Когда для проверки используется переменная типа `char`, или необходимо предусмотреть большое количество разнообразных вариантов, то можно использовать конструкцию `switch` строго соблюдая пунктуацию, показанную ниже:

```
switch (ch)
{
    case 'r':
    {
        flag=true;
        break;
    }
    case 'w':
    {
        flag=false;
        break;
    }
    default:
    {
        flag=false;
    }
}
```

Функции

Имена функций могут состоять не менее чем из трех символов, которые должны быть латинскими буквами, и при особой необходимости арабскими цифрами.

Имена функций должны состояться из одного, двух или трех как можно более коротких англоязычных слов, каждое из которых записывается с большой буквы, например: `GetCount`, `SetTop` и т.п.

Использовать в функциях аргументы по умолчанию нежелательно.

Желательно как можно чаще использовать передачу аргументов в функции по константной ссылке.

Когда есть выбор, то предпочтительнее использовать функций стандартной библиотеки.

Указатели и ссылки

Лучше ссылки, чем указатели.

Классы

Имена классов могут состоять не менее чем из трех символов, которые могут быть только латинскими буквами.

Имена классов должны состояться из одного, в крайнем случае двух как можно более коротких англоязычных слов. Имя класса записывается с большой буквы, например: `Sample`, `Matrix` и т.п.

Имена переменных-членов класса должны начинаться с префикса `'m_'`.

Желательно оформлять классы таким образом, чтобы было только по одному разделу `private`, `protected` и `private`.

У класса должно быть реализовано как минимум три конструктора: конструктор по умолчанию, полный конструктор и конструктор копирования.

Если функция-член не модифицирует значение переменных членов класса, то она объявляется как константная.

Использование открытых переменных-членов нежелательно.