



ДзЗ

1. Поведение переменных, указателей и ссылок при передаче в функцию.

При передаче **переменных** в функцию функция получает копию переменной: при компиляции функции для её переменных выделяются отдельные участки памяти, при вызове функции вычисленные параметры заменяют переменные на этих участках памяти. Передача переменной функцию по значению позволяет преобразовывать тип значения при вызове функции.

Указатели передаются в функцию по значению: функция получает копию указателя; отличие от поведения переменной в том, что копия будет иметь тот же адрес, что и указатель.

При передаче аргументов **по ссылке** функция использует сам аргумент, а не его значение → при передаче по ссылке в функцию можно менять сам объект.

2. Как ведут себя статические переменные, объявленные внутри функций.

Статистические переменные инициализируются только раз и сохраняют своё значение всю жизнь программы (тк хранятся в статистической области памяти, а не в стеке).

3. Аргументы функций по умолчанию.

— аргумент функции, который можно не указывать при вызове функции (добавляется компилятором автоматически).

Общий вид объявления:

```
ret_type FuncName(type1 var1 = val1, ..., typeN varN = valN)
{
    тело функции
}
```

ret_type — тип возвращаемой функции

FuncName — имя функции

type* — тип переменной var*

val* — значение переменной var*

4. *На основе чего осуществляется выбор перегруженной функции. (Не как, а на основе чего!)*

Выбор перегруженной функции осуществляется в зависимости от типов и количества аргументов, которые были переданы в функцию при вызове.

5. *Отличия inline функций от макросов.*

```
// реализация inline-функции
inline return_type funct_name (параметры)
{
    тело функции
}

// реализация макроса
#define macro_name char_sequence
```

inline-функции	макросы
анализируются компилятором	обрабатываются во время препроцессинга
может быть определена внутри или вне класса	всегда определяется в начале программы
считывает аргумент только раз	считывает аргумент каждый раз, когда встречается его в коде
короткие функции, определенные внутри класса, автоматически превращаются в встроенные	макросы должны быть определены специально

inline-функции	макросы
функции	
исправить баг несложно, тк проверка ошибок выполняется во время компиляции	проверка ошибок выполняется до компиляции, поэтому отладка затруднена