

План занятия (08.09.17)

- тернарный оператор
- inline
- функции с переменным числом аргументов

Тернарный оператор

Тернарный оператор

```
condition ? expression1 : expression2;
```

```
int a = 1;
```

```
int b = 1;
```

```
int c = 1;
```

```
a = true ? ++b : ++c;
```

Тернарный оператор. Нюансы

```
int i;  
int j;
```

```
// Все хорошо :)
```

```
(true ? i : j) = 45;
```

VS

```
short i;  
int j;
```

```
// Ошибка компиляции
```

```
(true ? i : j) = 45;
```

Тернарный оператор. Нюансы

```
struct C{};
```

```
struct D{};
```

```
// Ошибка компиляции
```

```
(true ? C() : D());
```

Тернарный оператор. Плюшки

// Ошибка компиляции

```
struct S
```

```
{
```

```
    S() : if(true) i_(1) else i_(0) {}
```

```
    int i_;
```

```
};
```

Тернарный оператор. Плюшки

```
// спасибо тернарному оператору!  
struct S  
{  
    S() : i_(some_condition ? 0 : 1) {}  
    int i_;  
};
```

Тернарный оператор. Плюшки

// Ошибка компиляции

```
int& i;
```

```
if(some_condition)
```

```
    i = a;
```

```
else
```

```
    i = b;
```

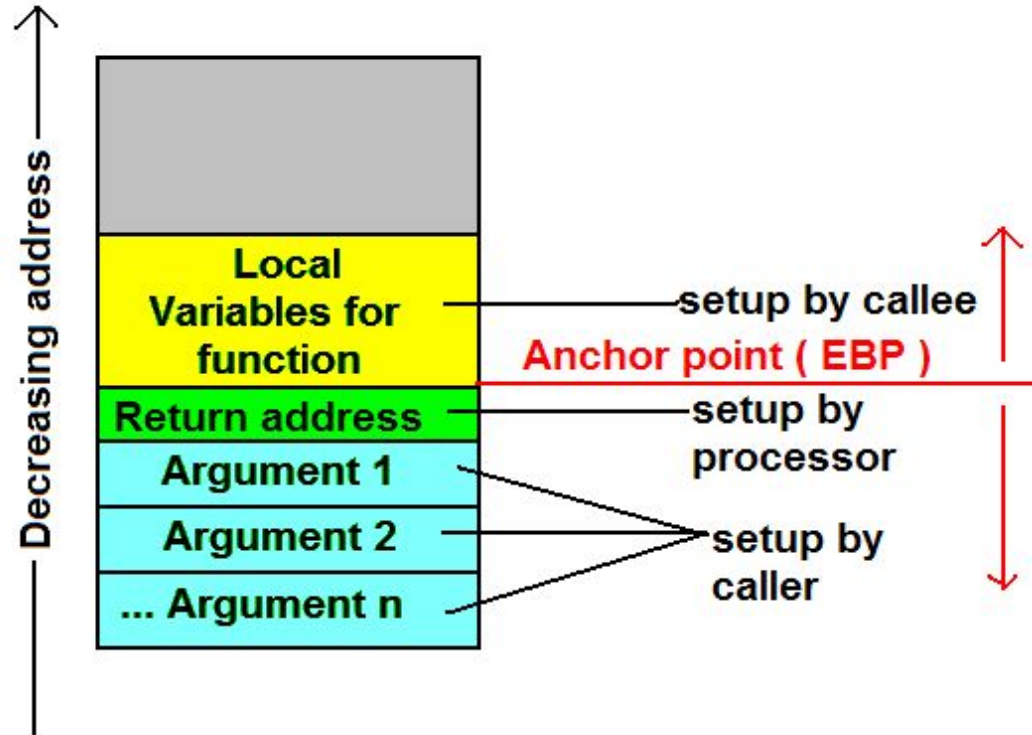

Тернарный оператор. Плюшки

```
// спасибо тернарному оператору!
```

```
int& i = (some_condition ? a : b);
```

inline

Механизм вызова функции



Когда функция является inline

- функция объявлена как inline (явно)

```
inline int sum( int a, int b )  
{  
    return a + b;  
}
```

Когда функция является inline

- функция определена в теле класса/структуры

```
struct A {  
  
    void foo()  
    {  
        // код  
    }  
};
```

Variadic functions

Variadic functions. Примеры функций

```
int printf( const char* format, ... );
```

```
int scanf( const char* format, ... );
```

```
int fscanf( std::FILE* stream,  
            const char* format, ... );
```

Variadic functions. Пример использования

```
std::printf( "Decimal:\t%i %d %.6i %i %.0i\n",  
            1, 2, 3, 0, 0 );
```


Variadic functions. Реализация

```
int printf( const char* format, ... );
```

```
std::va_list
```

```
std::va_start
```

```
std::va_copy
```

```
std::va_arg
```

```
std::va_end
```