

Императивная  
последовательность  
действий

Декларативная  
представление  
результата

Процедурное  
программирование  
(C)

ООП

объект, у  
которого есть  
ф-ции.

- C++, Java, C#, Python, JS...

① Процедурная: - X (кнопка)  
- и т.д.

- микроархитектура.

② ООП: и взаимодействие  
с user

- games, прикл. ПО...

③ Функциональная: - криптовалюта, блокчейн.

④ Декларативная: графич. интерфейсы

CSS, qml

функциональные

созд. новых

гаммисс.

Haskell, Scala,

Erlang.



Трунгин 0017

## ① Инкапсуляция

- ограничение доступа к компонентам объектов (аттрибутам и методам)

- объединение данных в един.

структуру (класс)

C

C++

```
struct A {
    int n;
```

```
class A {
    int n;
```

```
public
```

```
int get N();
```

```
void set N(int);
```

## ② Наследование

- создание нового класса на основе существующих. (производный)

## ③ Полиморфизм

- перегрузка метода



④ Абстракции

- виртуальные ф-ции, шаблоны  
достигается путем создания  
абстракт. классов, интерфейсов,  
методов, которые определяют  
шаблоны, которые должны следовать  
производные классы

Тюпестуанство ииен в C++  
namespace - область ииен  
Name:: func...

```
namespace ииен {  
    void func();  
}
```

```
using namespace ииен;
```

! никогда не делать так в заголовочных  
файлах.



одинаковые названия, разные параметры, одинаковое или совсем разное.

- переопределение методов:

одинаковые названия, параметры, разные реализации.

(C)

int pow1(int x, ...)

double pow2(int x, ...)

(C++)

pow(int)

pow(double)

class A {

void func();

};

class B: A {

void func();

};



Синтаксис

1. std::endl;
2. std::flush;

```
#include <iostream>
```

```
int main() {
```

```
    std::cout << "Hello World" << endl;
```

```
    return 0;
```

```
}
```

Уникализация переменных

① int a = 5; — инициализация и конструирование

② int a{a};

Суммирование преобразования:

int a = 3.2;

int a{3.2};

char b = 300;



Стандартные потоки ввода-вывода. #include <iostream>

① cin - поток ввода.

>> - оператор извлечения  
int a;

std::cin >> a;

② cout - поток вывода

<< - оператор вставки.

cout << a;

← с буферизацией  
без

③ std::cerr

- вывод ошибок

Буферизация - процесс временного хранения выходящих данных в памяти перед их фактической отправкой на устройство вывода.



Ссылки:

① указатели - переменная, содержащая в себе адрес.  
- выделяется память.

- она может содержать null ptr

② Ссылки-псевдонимы

- не выделяют памяти

- не имеют ни какого ссылача

int a;		int &ref = a;
int *p = &a;		ref = 5;
*p = 5;		

```
void func(int &a) {
```

```
    a = 10;
```

```
}
```

```
void func(const int &a)
```