



Я хочу услышать
три главных слова...

Инкапсуляция
Наследование
Полиморфизм

ООП/С++: Лекция 2

Полиморфизм, инкапсуляция

ака «Что там под капотом у cin и cout»

Краткое содержание прошлой лекции

Было на C

```
struct stack_handle
{
    int top;
    int size;
    int* data;
};
```

```
void init(struct stack_handle* s, int size);
void finalize(struct stack_handle* s);
void push(struct stack_handle* s, int a);
int pop(struct stack_handle* s);
void clear(struct stack_handle* s);
```

Станет на C++

```
class stack
{
public:
    stack(int size);
    ~stack();
    void push(int a);
    int pop();
    void clear();
private:
    int top;
    int size;
    int* data;
};
```

<https://github.com/avasyukov/oop-2nd-term/tree/master/2019/lection02>



http://judge2.vdi.mipt.ru/cgi-bin/new-client?contest_id=911138



Полиморфизм

Мифы:

- Полиморфизм – это что-то невообразимо сложное про наследование, интерфейсы и виртуальные функции
- А вот в C++ есть волшебные `cin` и `cout`, которые сами всё считывают и печатают как нужно

Варианты определения и трактовки:

- Способность функции обрабатывать данные разных типов
- Наличие единого интерфейса для разных реализаций
- ...

Можно найти очень много разных способов решить эти задачи, поэтому видов полиморфизма много.

Даже вот так – формально уже полиморфизм

```
void func(char c)
{
    printf("func(char)▯called\n");
}
```

```
void func(int c)
{
    printf("func(int)▯called\n");
}
```

Полиморфизм и перегрузка в C++

Разбираем примеры кода к данной лекции

- 01__overload.c
- 02__simple__io.cpp
- 03__io__internals.cpp
- 04__stupid__point.cpp
- 05__semi__smart__point.cpp

Инкапсуляция

Варианты определения и трактовки:

- Упаковка данных и функций для работы с ними в единый компонент.
- Скрытие деталей реализации, ограничение доступа одних компонентов к другим.

В разных случаях могут иметь в виду только первое, только второе или оба пункта вместе.

Разбираем примеры:

- Реализация стека из прошлой лекции
- `06_smart_point_with_overloaded_operators.cpp`

Что стоит запомнить из лекции

- Полиморфизм – это когда можно сделать единый интерфейс для разных сущностей
- Перегрузка конструкторов, методов и операторов – один из видов полиморфизма
- Инкапсуляция – это когда вся логика некоторой сущности запаковывается внутрь этой самой сущности
- Инкапсуляция и полиморфизм нам нужны не для красоты (и не чтобы умничать с новыми терминами), а для того, чтобы код был более читаемый

<https://tinyurl.com/yxze5xws>





TO BE CONTINUED...