OTUS

Онлайн образование

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Ставим “+”, если все хорошо
“-”, если есть проблемы

Тема вебинара

ООП в C++

Андрей Рыжиков

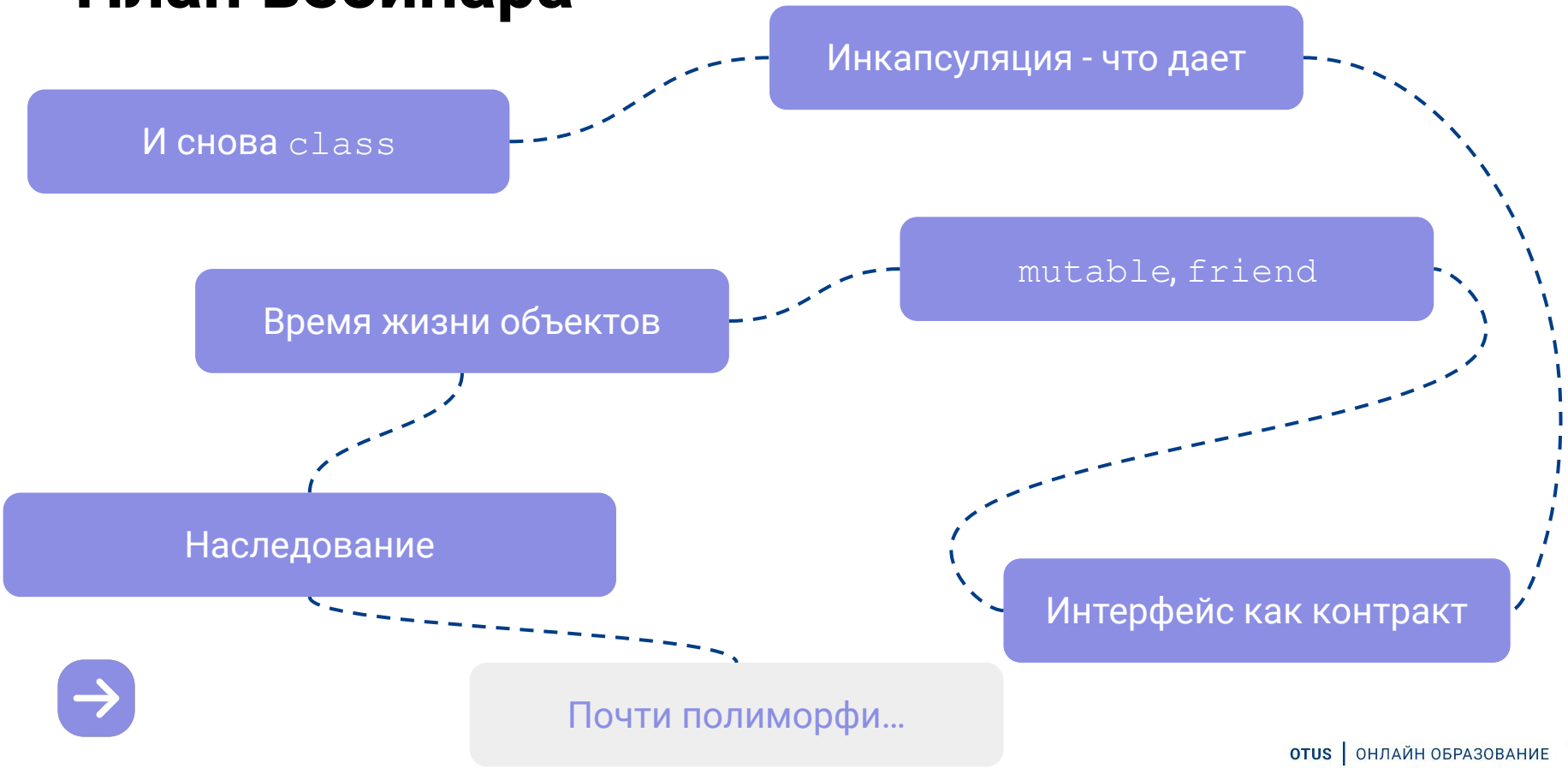
научный сотрудник НИИ обработки аэрокосмических изображений

Почти 10 лет опыта разработки на C++ (обработка изображений)

Отвечаю за модули автоматического уточнения геопривязки снимков от отечественных спутников.

@ryzhikovas

План вебинара



И снова `class`

- Класс - тип данных
- Объект - экземпляр класса
- Ключевые составляющие класса - поля и методы

  ...могут быть `public`, `private` и ...

Кстати, в стандарте нет понятия метод. Есть member function

Что дает?

- Сложные абстракции из простых
- `public` методы - мостики для взаимодействия объектов

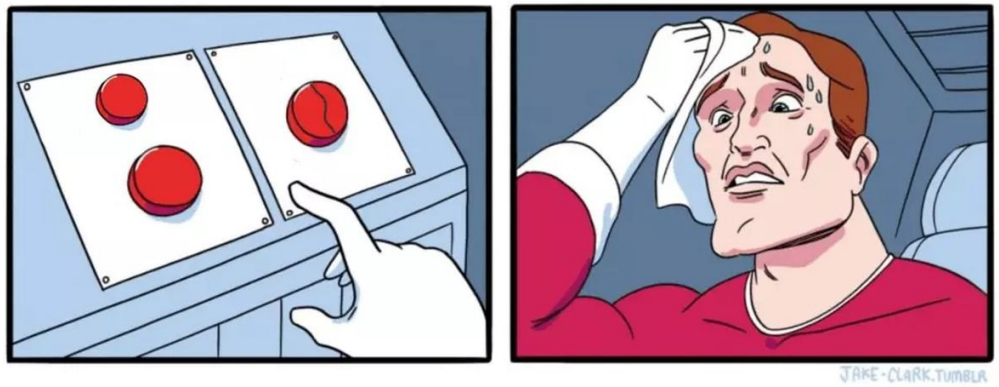
Инкапсуляция - что дает

- Гарантия инвариантов
- Контроль сложности ПО
- Возможность измениться в будущем

```
1 class Spaceflight {
2 public:
3  Spaceflight(Payload payload, Fuel fuel, ...);
4  bool setTarget(const Location& target);
5  bool launch();
6 private:
7    //...
8 };
```

Интерфейс

- В нашем контексте это `public` методы. Т.е. “ручки”, за которые может дергать пользователь вашего класса



- Контракт с пользователем
- Документируйте в первую очередь
- Начинайте с него чтение незнакомой программы

const

- Используйте, гласит CppCoreGuidelines
- Важная часть контракта
- Компилятор проверит
- Часть сигнатуры
- Логическая константность

```
1 class String {
2 public:
3  //…
4  std::size_t length() const;
5 };
```

const

```
 1 class Text {
 2 public:
 3  void render(Surface& target);
 4  void set(std::string text);
 5  std::string get();
 6 };
 7 //…
 8 void show(Text& text, Surface& s) {
 9   if (text.get().empty()) {
10     text.set("<none>");
11   } else
12     text.render(s);
13 }
```

const


```
 1 class Text {
 2 public:
 3  void render(Surface& target) const;
 4  void set(const std::string& text);
 5  std::string get() const;
 6 };
 7 //возможная реализация
 8 void show(const Text& text, Surface& s) {
 9   if (text.get().empty()) {
10     Text tmp; tmp.set("<none>"); tmp.render(s);
11   } else
12     text.render(s);
13 }
```

`mutable`

- В CppCoreGuidelines есть все.
- “Отменяет” `const`

mutable

```
 1 class String { // Warning. Not owning
 2 public:
 3  std::size_t length() const {
 4    if (!cacheIsActual) {
 5      cachedLength = strlen(sStyleStr); cacheIsActual = true;
 6    }
 7    return cachedLength;
 8  }
 9 private:
10  const char* sStyleStr = nullptr;
11  std::size_t cachedLength = 0;
12  bool cacheIsActual = true;
13 };
```

mutable

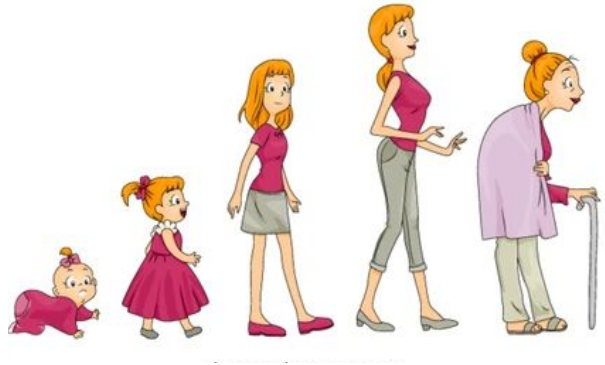
```
 1 class String { // Warning. Not owning
 2 public:
 3  std::size_t length() const {
 4    if (!cacheIsActual) {
 5      cachedLength = strlen(sStyleStr); cacheIsActual = true;
 6    }
 7    return cachedLength;
 8  }
 9 private:
10  const char* sStyleStr = nullptr;
11  mutable std::size_t cachedLength = 0;
12  mutable bool cacheIsActual = true;
13 };
```

`friend`

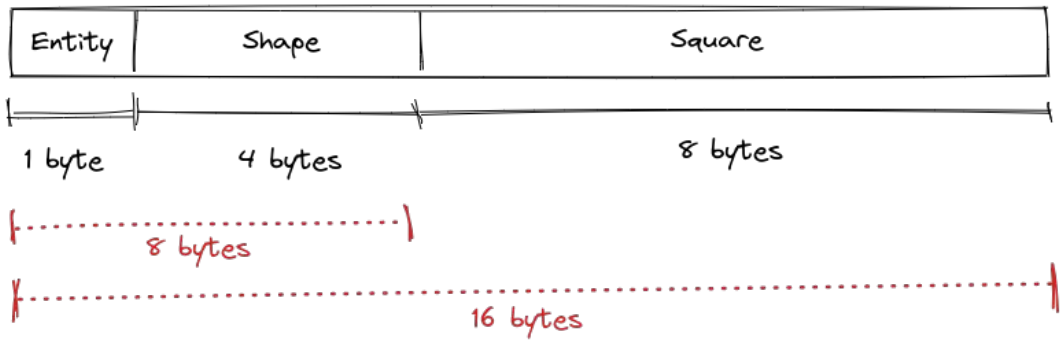
- Ничего не скрываем от друзей
- "Отменяет" `private`
- Друг - функция или класс

Время жизни

- Автоматические объекты.
  … все локальные, если не `static`, `extern` или `thread_local`
- Объекты в динамической памяти
  … `new`, `std::vector`, `std::map` …
- … `//depeer`



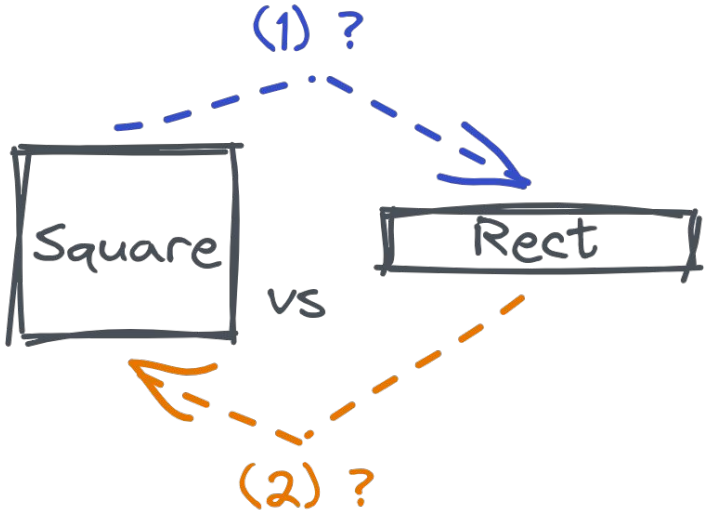
Наследование




```
 1 struct Entity {
 2   uint8_t id{255};
 3 };
 4
 5 using Color = uint32_t;
 6 struct Shape: public Entity {
 7   Color color{};
 8 };
 9
10 struct Square: public Shape {
11   double length{};
12 };
```

Наследование

```
1 class Rect: public Square{
2   //…
3 };
4 //или
5 class Square: public Rect {
6   //…
7 };
```



Заполните, пожалуйста, опрос о занятии по ссылке в чате

Вопросы?

Слушаю/читаю в чате