

# Программирование на языке C++

## Лекция 4

Виртуальные методы

Александр Смаль

## Переопределение методов (overriding)

```
→ struct Person {  
    → string name() const { return name_; }  
    ...  
};  
→ struct Professor : Person {  
    → string name() const {  
        return "Prof. " + Person::name();  
    }  
    ...  
};
```

```
→ Professor pr("Stroustrup");  
→ cout << pr.name() << endl; // Prof. Stroustrup  
→ Person * p = &pr;  
→ cout << p->name() << endl; // Stroustrup
```

# Виртуальные методы

```
struct Person {  
    - virtual string name() const { return name_; }  
    ...  
};  
- struct Professor : Person {  
    - string name() const {  
        return "Prof. " + Person::name();  
    }  
    ...  
};
```

```
Professor pr("Stroustrup");  
cout << pr.name() << endl; // Prof. Stroustrup  
- Person * p = &pr;  
cout << p->name() << endl; // Prof. Stroustrup
```

## Чистые виртуальные (абстрактные) методы

~~Person p;  
p.occupation();~~

```
struct Person {  
    virtual string ocupation() const = 0;  
    ...  
};  
struct Student : Person {  
    string ocupation() const {return "student";}  
    ...  
};  
struct Professor : Person {  
    string ocupation() const {return "professor";}  
    ...  
};
```

```
Person * p = next_person();  
cout << p->ocupation();
```

# Виртуальный деструктор

К чему приведёт такой код?

```
• struct Person {  
    ...  
};  
• struct Student : Person {  
    ...  
private:  
    string uni_; * Oxford  
};  
  
int main() {  
    Person * p = new Student("Alex", 21, "Oxford");  
    ...  
    delete p; ~Person()  
}
```

# Виртуальный деструктор

Правильная реализация:

```
struct Person {  
    ...  
    virtual ~Person() {}  
};  
struct Student : Person {  
    ...  
private:  
    string uni_;  
};  
  
int main() {  
    Person * p = new Student("Alex", 21, "Oxford");  
    ...  
    delete p;      ~Student()  
}
```

# Полиморфизм

## Полиморфизм

Возможность единообразно обрабатывать разные типы данных.

- Перегрузка функций

Выбор функции происходит в момент компиляции на основе типов аргументов функции, статический полиморфизм.

- Виртуальные методы

Выбор метода происходит в момент выполнения на основе типа объекта, у которого вызывается виртуальный метод, динамический полиморфизм.