

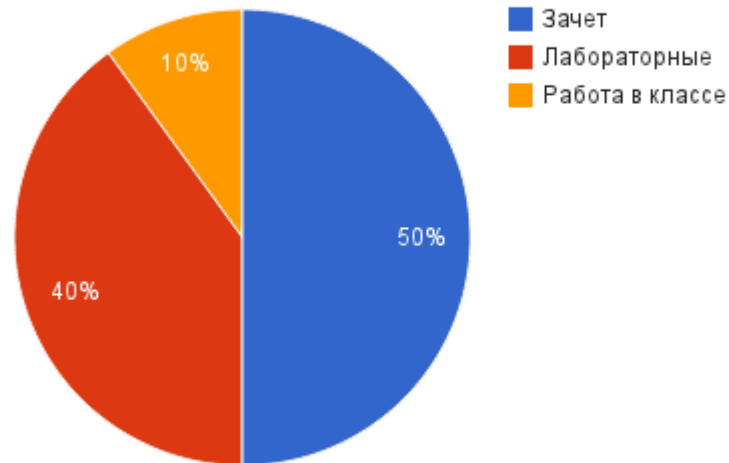
ООП

Введение

7 сентября, 2012
Алексей Лебедев

Как мы будем работать

1. 4 лекции.
2. 2 лабораторных.
3. В конце семестра — зачет.



О чем курс

1. На самом деле он о C++ :-).
2. Инкапсуляция (классы).
3. Полиморфизм и наследование.
4. Динамическое связывание (виртуальные функции).
5. Обобщенное программирование.
6. Обзор c++11.

План на сегодня

1. Что не так с языком Си.
2. Как решает эти проблемы C++.
3. И как C++ меняет подход к программированию.

Проблемы языка Си

1. Методы не привязаны к структурам.
2. `int_list, double_list, char_list, ...`
3. Интерфейс смешан с реализацией.
4. Ручное освобождение ресурсов (`free, fclose, pthread_destroy, ...`).

Пример: списки

```
typedef struct elem elem;  
typedef elem *list;  
  
extern list nil;  
  
bool empty(list);  
list cons(int, list);  
int head(list);  
list tail(list);  
list settail(list l, list tl);  
list sethead(list l, int x);  
void print(list);
```

Добавляем немножко ООП

1. Инкапсуляция: объединяем структуру и методы.
2. Деструкторы.
3. Шаблоны.
4. Выделяем интерфейс.

Что получилось

```
template<class T>
class list {
    struct elem {
        T val;
        elem *next;
        elem(const T &v, elem *n): val(v), next(n) {}
    };
    elem *first;
public:
    list(): first(0) {}
    ~list();

    bool empty() const { return first == 0; }
    elem &head() { return first->val; }
    const elem &head() const { return first->val; }
    void push_front(const T &v) { first = new elem(v,
first); }
```


Выводы

1. `std::vector<std::shared_ptr<Data>>::const_iterator ...`
2. Но C++ мощнее и универсальнее.
3. Без потерь производительности.