Реализация умных указателей

Гусев Илья, Булгаков Илья

Московский физико-технический институт

Москва, 2019

Содержание

ScopedPtr

- Простой умный указатель: для хранения на стеке или в классе.
- Единственные владелец.
- Нельзя копировать и присваивать.
- Нельзя вернуть владение объектом.

ScopedPtr. Основные методы

- get — возвращает указатель, сохраненный внутри ScopedPtr (например, чтобы передать его в какую-то функцию)
- release — забирает указатель у ScopedPtr и возвращает значение этого указателя, после вызова release ScopedPtr не должен освобождать память (например, чтобы вернуть этот указатель из функции)
- reset — метод заставляет ScopedPtr освободить старый указатель, а вместо него захватить новый (например, чтобы переиспользовать ScopedPtr, так как оператор присваивания запрещен)

ScopedPtr. Задача

Напишите реализацию методов для ScopedPtr

```
struct Expression;

struct ScopedPtr
{
    // TODO
    //
    //explicit ScopedPtr(Expression *ptr = 0);
    //~ScopedPtr();
    //Expression* get() const;
    //Expression* release();
    //void reset(Expression *ptr = 0);
    //Expression& operator*() const;
    //Expression* operator->() const;
private:
    ScopedPtr(const ScopedPtr&);
    ScopedPtr& operator=(const ScopedPtr&);

    Expression *ptr_;
};
```

ScopedPtr. Реализация

```
struct Expression;

struct ScopedPtr
{
    explicit ScopedPtr(Expression *ptr = 0) { ptr_ = ptr; }
    ~ScopedPtr() { delete ptr_; }
    Expression* get() const { return ptr_; }
    Expression* release() { Expression *temp = ptr_; ptr_ = 0;
        return temp; }
    void reset(Expression *ptr = 0) { if(ptr == ptr_) { return; }
        else { delete ptr_; ptr_ = ptr; } }
    Expression& operator*() const { return *ptr_; }
    Expression* operator->() const { return ptr_; }
private:
    ScopedPtr(const ScopedPtr&);
    ScopedPtr& operator=(const ScopedPtr&);

    Expression *ptr_;
};
```

SharedPtr

Умный указатель с подсчетом ссылок

```
{
    SharedPtr sp1( new Expression() );
    SharedPtr sp2(sp1);
}
```

Особенности:

- Для разделяемых объектов.
- Ведётся подсчёт ссылок.
- Нельзя вернуть владение объектом.

SharedPtr. Задание

Напишите реализацию методов для SharedPtr

```
struct Expression;
struct Number;
struct BinaryOperation;

struct SharedPtr
{
    // TODO
    //
    // explicit SharedPtr(Expression *ptr = 0)
    // ~SharedPtr()
    // SharedPtr(const SharedPtr &)
    // SharedPtr& operator=(const SharedPtr &)
    // Expression* get() const
    // void reset(Expression *ptr = 0)
    // Expression& operator*() const
    // Expression* operator->() const
};
```