

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра МО ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №4**  
**по дисциплине «Объектно-ориентированное программирование»**  
**Тема: shared\_ptr**

Студентка гр. 7303

\_\_\_\_\_

Аплачкина Е.А.

Преподаватель

\_\_\_\_\_

Размочаева Н.В.

Санкт-Петербург

2019

### **Цель работы:**

Реализовать умный указатель разделяемого владения объектом (shared\_ptr). Поведение реализованных функций должно быть аналогично функциям std::shared\_ptr.

### **Ход работы:**

Был реализован класс shared\_ptr, содержащий в себе два приватных поля: T \*ptr – указатель на объект типа T и long \*count – счетчик, который хранит в себе количество указателей на данный объект. Были написаны следующие методы для класса shared\_ptr:

- 1) Конструктор explicit shared\_ptr(T \*ptr = 0);
- 2) Конструктор копирования shared\_ptr(const shared\_ptr & other);
- 3) Конструктор копирования shared\_ptr (constshared\_ptr<Sad> & other):pObj (other. pObj) , count (other.count);
- 4) Оператор присваивания shared\_ptr& operator=(const shared\_ptr<Sad> & other);
- 5) Оператор присваивания shared\_ptr& operator=(const shared\_ptr & other);
- 6) Деструктор ~ shared\_ptr();
- 7) Оператор explicit operator bool() const;
- 8) Оператор T& operator\*() const;
- 9) Оператор T\* operator->() const;
- 10) Метод T\* get() const;
- 11) Метод void swap(shared\_ptr& x) noexcept;
- 12) Метод void reset(T \*ptr = 0);

## **Выводы**

В ходе выполнения лабораторной работы были изучены и реализованы умные указатели. Были реализованы основные функции для работы с ними. Поведение реализованных функций соответствует классам `std::shared_ptr`.