

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования «Алтайский государственный технический
университет им. И. И. Ползунова»

Факультет информационных технологий Кафедра прикладной математики

Отчет защищен с оценкой _____

Преподаватель _____ (подпись)

«___» _____ 2023 г.

Отчет
по лабораторной работе № 6

« Структурные паттерны 1 - проектирование
(Adapter, Decorator)

по дисциплине «Объектно-ориентированное программирование»

Студент гр. ПИ-02
Чередов Р.А.

Ассистент кафедры ПМ,
Рахманин Д. С.

Барнаул 2023

Тема: Структурные паттерны 1 - реализация
(Adapter, Decorator , Composite, Iterator)

Отчет:

Логи работы программы Архив с кодом программы

125	Чередов Роман Алексеевич	Системы анализа больших данных	Проектирование системы обработки потоков данных о местонахождении абонентов сети сотовой связи
-----	-----------------------------	-----------------------------------	--

Adapter

```
#include <iostream>
#include <map>
#include <random>
#include <time.h>
using namespace std;
class ILineUP { // Интерфейс линии связи
private:
    ILineUP* line;
public:
    void setline(ILineUP* c) { line = c; }
    virtual void MakePacket() = 0;
    virtual void GivePacket() = 0;
};
// Реализация - линии связи
class ClassPacketLine : public ILineUP {
public:
    void MakePacket() { cout << " Make Packet " << endl; }
    void GivePacket() { cout << " Give Packet " << endl; }
};

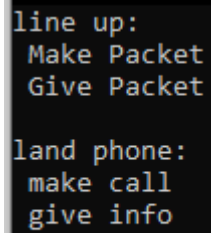
class LandPhone { // адаптируемый класс, не реализующий ILineUP
public:
    void MakeCall() { cout << " make call " << endl; }
    void GiveInfo() { cout << " give info " << endl; }
};

class Adapter : public ILineUP { // адаптер, реализующий интерфейс ILineUP
private:
    LandPhone* adp;
public:
    void MakePacket() { adp->MakeCall(); }
    void GivePacket() { adp->GiveInfo(); }
    Adapter() { adp = new LandPhone(); }
    ~Adapter() { delete adp; }
};

int main() {
    //Создаем линию связи
    ILineUP* car = new ClassPacketLine();
    //Создаем лодку
    Adapter* LandPhone = new Adapter();
    //Проверка автомобиля
    cout << "line up: " << endl;
    car->MakePacket();
    car->GivePacket();
    cout << endl;
    //Проверка лодки
    cout << "land phone: " << endl;
    LandPhone->MakePacket();
    LandPhone->GivePacket();
    cout << endl;
    delete LandPhone, car;
```

}

Интерфейсы и объекты:



```
line up:
  Make Packet
  Give Packet

land phone:
  make call
  give info
```

Рис. 1. Логи работы программы (Adapter)

Decorator

```
using namespace std;

class Ibasestation {
public:
    virtual string getDescription() = 0;
    virtual float getCost() = 0;
};

class AdditionalOption : public Ibasestation {
protected:
    Ibasestation* item;
public:
    virtual string getDescription() = 0;
    virtual float getCost() = 0;
    AdditionalOption(Ibasestation* inItem) { item = inItem; }
    ~AdditionalOption() {
        if (item) delete item;
    }
};

class Beeline : public Ibasestation { // конкретный объект
public:
    virtual string getDescription() {
        return string("you using Beeline base station"); // название услуги
    }
    virtual float getCost() {
        return 2; // стоимость
    }
};

class MTS : public Ibasestation { // конкретный объект
public:
    virtual string getDescription() {
        return string("you using MTS base station"); // название услуги
    }
    virtual float getCost() {
        return 1; // стоимость
    }
};

class Rouming : public AdditionalOption { // конкретный декоратор
public:
    virtual string getDescription() {
```

```

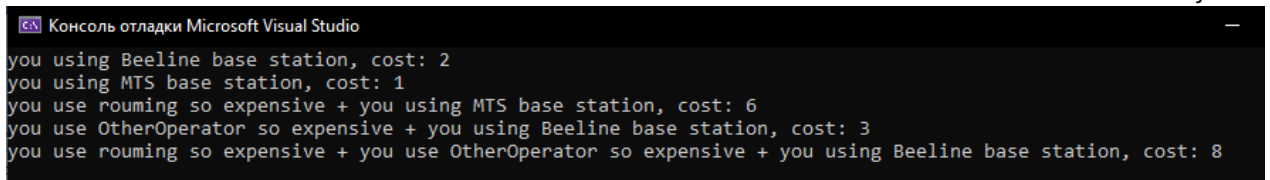
        return string("you use rouming so expensive + ") + item->getDescription();
// Составная услуга
    }
    virtual float getCost() {
        return (5 + item->getCost()) * 1; //Цена в роуминге
    }
    Rouming(Ibasestation* inItem) : AdditionalOption(inItem) {}
};
class OtherOperator : public AdditionalOption { // конкретный декоратор
public:
    virtual string getDescription() {
        return string("you use OtherOperator so expensive + ") + item-
>getDescription();
    }
    virtual float getCost() {
        return (1 + item->getCost()) ; //Составная цена со скидкой
    }
    OtherOperator(Ibasestation* inItem):AdditionalOption(inItem) {}
};
int main() {
    Beeline* bee = new Beeline();
    cout << bee->getDescription() << ", cost: " << bee->getCost() <<endl;
    MTS* mst = new MTS();
    cout << mst->getDescription() << ", cost: " << mst->getCost()
        << endl;

    Rouming* roum = new Rouming(mst);
    cout << roum->getDescription() << ", cost: " << roum->getCost() << endl;

    OtherOperator* other = new OtherOperator(bee);
    cout << other->getDescription() << ", cost: " << other->getCost() << endl;

    Rouming* otherroum = new Rouming(other);
    cout << otherroum->getDescription() << ", cost: " << otherroum->getCost() <<endl;
    delete bee, mst;
    delete roum, other, otherroum;
    return 0;
}

```



Консоль отладки Microsoft Visual Studio

```

you using Beeline base station, cost: 2
you using MTS base station, cost: 1
you use rouming so expensive + you using MTS base station, cost: 6
you use OtherOperator so expensive + you using Beeline base station, cost: 3
you use rouming so expensive + you use OtherOperator so expensive + you using Beeline base station, cost: 8

```

Рис. 2. Лог работы программы (Decorator)