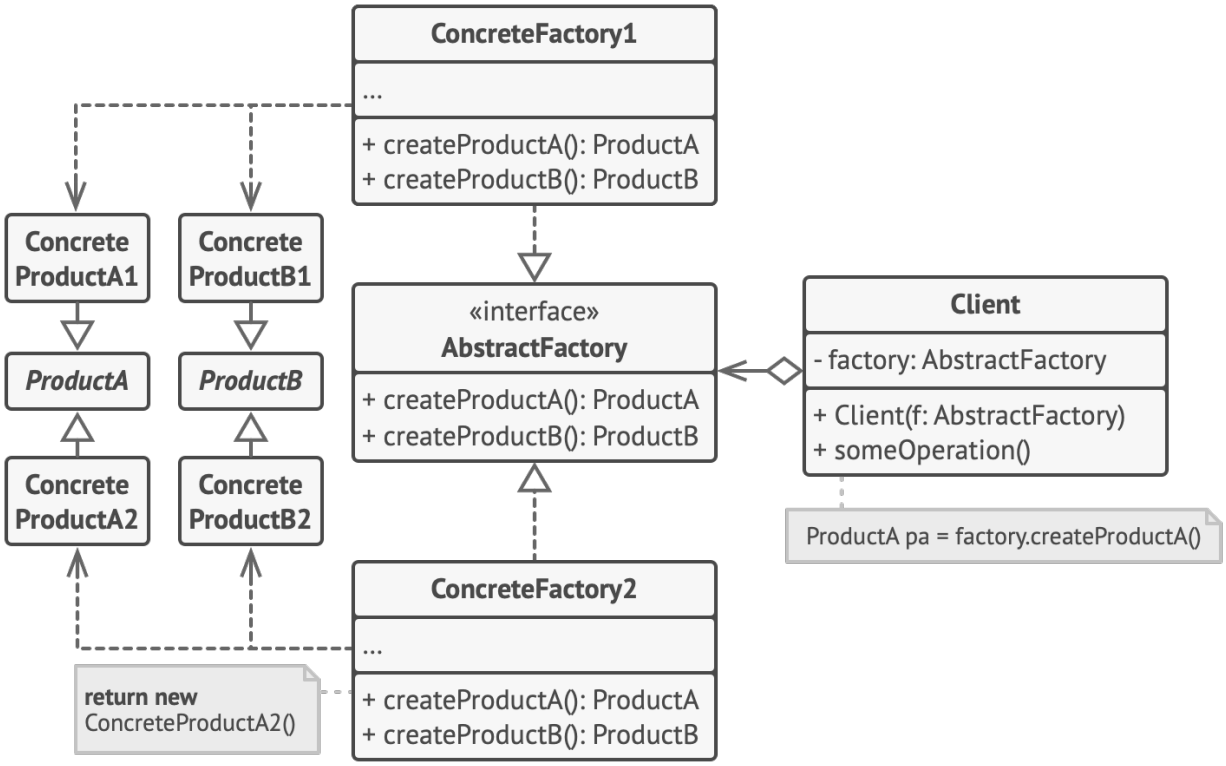
Лабораторная работа №5

Стрекнев Д. СКБ172

Задание:

Используя паттерн Abstract Factory, реализовать представление графического интерфейса приложения, состоящего из нескольких частей (например, текст, баннеры, руководство пользователя) для нескольких языков (например, русского, английского и т.д.).

Паттерн Abstract Factory:



Код:

```
#include <iostream>
using namespace std;

// Абстрактные базовые классы всех возможных частей интерфейса
class Text {
public:
    virtual void text() = 0;
    virtual ~Text(){};
};

class Banner1 {
public:
    virtual void banner1() = 0;
    virtual ~Banner1(){};
};

class Banner2 {
public:
    virtual void banner2() = 0;
    virtual ~Banner2(){};
};

class Manual {
public:
    virtual void manual() = 0;
    virtual ~Manual(){};
};



// Классы всех частей интерфейса на английском языке
class English_Text: public Text {
public:
    void text(){cout << "English_Text" << endl;}
};

class English_Banner1: public Banner1 {
public:
    void banner1(){cout << "English_Banner1" << endl;}
};

class English_Banner2: public Banner2 {
public:
    void banner2(){cout << "English_Banner2" << endl;}
};

class English_Manual: public Manual {
public:
    void manual(){cout << "English_Manual" << endl;}
```

```
};



// Классы всех частей интерфейса на русском языке
class Russian_Text: public Text {
public:
    void text(){cout << "Russian_Text" << endl;}
};

class Russian_Banner1: public Banner1 {
public:
    void banner1(){cout << "Russian_Banner1" << endl;}
};

class Russian_Banner2: public Banner2 {
public:
    void banner2(){cout << "Russian_Banner2" << endl;}
};

class Russian_Manual: public Manual {
public:
    void manual(){cout << "Russian_Manual" << endl;}
};



// Классы всех частей интерфейса на французском языке
class French_Text: public Text {
public:
    void text(){cout << "French_Text" << endl;}
};

class French_Banner1: public Banner1 {
public:
    void banner1(){cout << "French_Banner1" << endl;}
};

class French_Banner2: public Banner2 {
public:
    void banner2(){cout << "French_Banner2" << endl;}
};

class French_Manual: public Manual {
public:
    void manual(){cout << "French_Manual" << endl;}
};
```

```
// Абстрактная фабрика для производства частей интерфейса
class Part_Factory {
public:
    virtual Text* create_Text() = 0;
    virtual Banner1* create_Banner1() = 0;
    virtual Banner2* create_Banner2() = 0;
    virtual Manual* create_Manual() = 0;
    virtual ~Part_Factory(){};
};



// Фабрика для создания частей интерфейса на английском языке
class English_Part_Factory: public Part_Factory {
public:
    Text* create_Text(){return new English_Text;}
    Banner1* create_Banner1(){return new English_Banner1;}
    Banner2* create_Banner2(){return new English_Banner2;}
    Manual* create_Manual(){return new English_Manual;}
};



// Фабрика для создания частей интерфейса на русском языке
class Russian_Part_Factory: public Part_Factory {
public:
    Text* create_Text(){return new Russian_Text;}
    Banner1* create_Banner1(){return new Russian_Banner1;}
    Banner2* create_Banner2(){return new Russian_Banner2;}
    Manual* create_Manual(){return new Russian_Manual;}
};



// Фабрика для создания частей интерфейса на французском языке
class French_Part_Factory: public Part_Factory {
public:
    Text* create_Text(){return new French_Text;}
    Banner1* create_Banner1(){return new French_Banner1;}
    Banner2* create_Banner2(){return new French_Banner2;}
    Manual* create_Manual(){return new French_Manual;}
};



// Класс, содержащий все части интерфейса на том или ином языке
class Parts {
public:
    ~Parts(){}
    void info(){
        t->text();
```

```
        b1->banner1();
        b2->banner2();
        m->manual();
    }
    Text* t;
    Banner1* b1;
    Banner2* b2;
    Manual* m;
};



// Здесь создается интерфейс на том или ином языке
class Interface {
public:
    Parts* createParts(Part_Factory& factory){
        Parts* p = new Parts;
        p->t=factory.create_Text();
        p->b1=factory.create_Banner1();
        p->b2=factory.create_Banner2();
        p->m=factory.create_Manual();
        return p;
    }

};



int main() {
    Interface interface;
    English_Part_Factory IPF;
    Russian_Part_Factory RPF;
    French_Part_Factory FPF;

    Parts* I = interface.createParts(IPF);
    Parts* R = interface.createParts(RPF);
    Parts* F = interface.createParts(FPF);

    cout << "English Interface:" << endl;
    I->info();
    cout << endl << "Russian Interface:" << endl;
    R->info();
    cout << endl << "French Interface:" << endl;
    F->info();
}
```

Результат:

```
English Interface:
English_Text
English_Banner1
English_Banner2
English_Manual

Russian Interface:
Russian_Text
Russian_Banner1
Russian_Banner2
Russian_Manual

French Interface:
French_Text
French_Banner1
French_Banner2
French_Manual
Program ended with exit code: 0
```