Домашнее задание

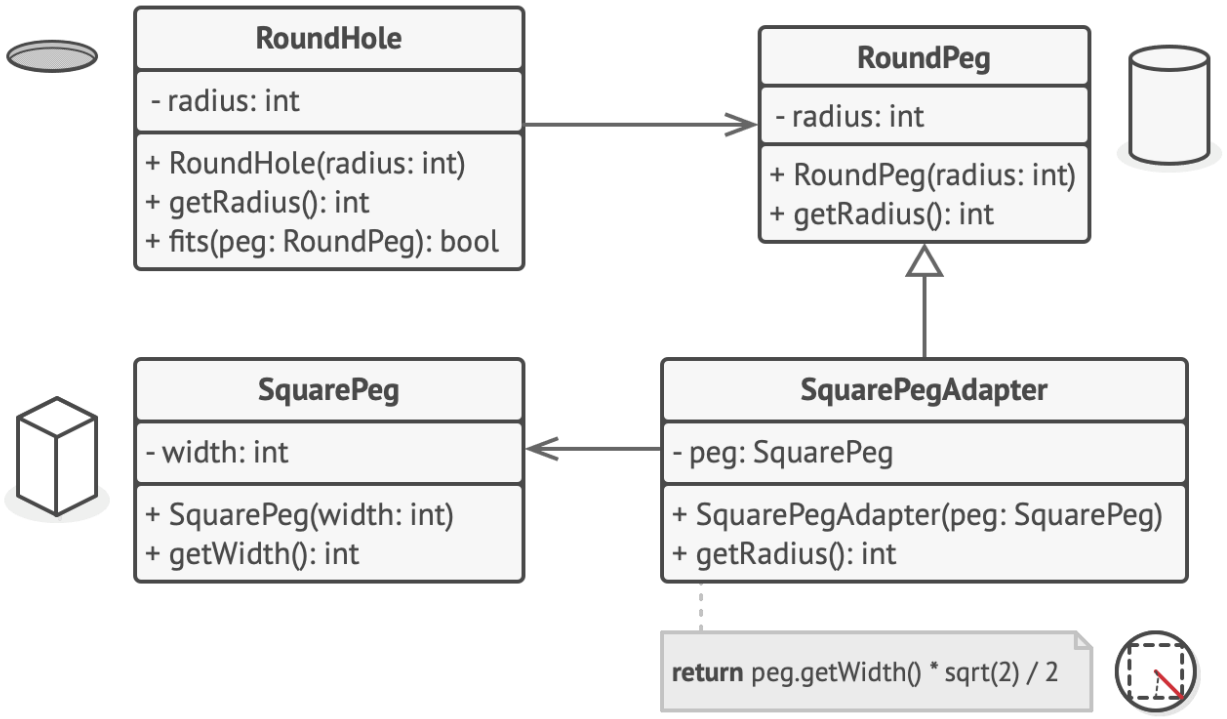
UML/Паттерны проектирования

Задание 1: По диаграмме классов UML, написать программный код, используя паттерн Адаптер.

Задача, описанная в UML диаграмме: Адаптер вычисляет наименьший радиус окружности, в которую можно вписать квадратный колышек, и представляет его как круглый колышек с этим радиусом. **Адаптер** преобразует один интерфейс в другой, позволяя совместить квадратные колышки и круглые отверстия.

Ссылка на паттерн Адаптер.

https://refactoring.guru/ru/design-patterns/adapter



Код паттерна Адаптер с комментариями:

```
//Паттерн Адаптер
/*Целевой класс объявляет интерфейс, с которым может работать
клиентский код*/

class Target{
public:
    virtual ~Target() = default;//virtual ~Target(){}
    virtual string Request() const{
        return "Target: поведение класса по умолчанию\n";
    }
};
/*Адаптируемый класс содержит некоторое полезное поведение, но его
интерфейс несовместим с существующим клиентским кодом. Адаптируемый
класс нуждается в некоторой доработке, прежде чем клиентский код
сможет его использовать*/
class Adaptee{
public:
    string SpecificRequest() const{
        return "Особое поведение Адаптируемого класса";
    }
};

class Adapter: public Target{
private:
    Adaptee *adaptee_;
public:
    Adapter(Adaptee *adaptee): adaptee_(adaptee){}
    string Request() const override{
        string to_reverse = adaptee_->SpecificRequest();
        reverse(to_reverse.begin(), to_reverse.end());
        return "Adapter: перевернул строку " + to_reverse;
    }
};
/*Клиентский код поддерживает все класс, использующие целевой интерфейс
Из класса Adaptee переводим объект в формат класса Target, используя
(с помощью класса) Adapter
*/
void ClientCode1(Target *target ){
    cout<<target->Request();
}
```

```
int main()
{
    cout<<"Client: Я могу работать только с  объектами типа Target\n";
    Target* target = new Target();
    ClientCode1(target);

    Adaptee* adaptee= new Adaptee();
    cout<<"Client: Adaptee имеет незнакомый интерфейс\n";
    cout<<"Adaptee: "<<adaptee->SpecificRequest();
    cout<<"Client: Я могу работать с объектами Adaptee, но через"<<
            " Adapter\n";
    Adapter *adapter = new Adapter(adaptee);
    ClientCode1(adapter);

    delete target;
    delete adaptee;
    delete adapter;
}
```