

# Семинар #5: Наблюдатель и Команда. Домашнее задание.

## Задача 1. Ещё один наблюдатель

Начальный код для задачи в файле `observer.cpp`. Добавьте ещё один объект, который будет наблюдателем слайдера. При изменении положения слайдера, должно меняться вертикальное положение нового объекта-наблюдателя.

## Задача 2. Два субъекта

Начальный код для задачи в файле `observer.cpp`. Сделайте эту задачу в отдельном файле. Измените код так, чтобы в программе было 2 субъекта-слайдера. При изменении положения одного слайдера все объекты должны вращаться. При изменении положения второго слайдера, все объекты должны менять цвет. Постарайтесь решить задачу таким образом, чтобы было меньше повторений кода.

## Задача 3. Случайные цвета

Начальный код для этой и последующих задач в файле `command.cpp`. Создайте новую команду, которая бы меняла цвета всех объектов на случайные. Присоедините эту команду к одной из кнопок.

## Задача 4. Очистка

Создайте новую команду, которая будет удалять все фигуры. Присоедините эту команду к одной из кнопок.

## Задача 5. Мультикоманда

Создайте новую команду `MultiCommand`, которая должна будет объединять в себе несколько команд. Класс `MultiCommand` должен быть реализацией интерфейса `Command`. Мультикоманда должна конструироваться из вектора команд. При исполнении мультикоманды должны исполняться подряд все входящие в неё команды. Создайте мультикоманду, которая будет одновременно меняет цвета и позиции всех фигур, используя уже написанные команды.

## Задача 6. Отмена команд

Добавьте новую кнопку, которая будет отменять последнюю исполненную команду. При нескольких последовательных нажатиях на эту кнопку, должны отмениться несколько последних команд. Для решения этой задачи измените интерфейс команды на следующий:

```
class Command
{
public:
    virtual void execute() = 0;
    virtual void unexecute() = 0;
    virtual ~Command() {}
};
```

Где виртуальный метод `unexecute` должен отменять действия, которые прошли в методе `execute`. Нужно будет реализовать метод `unexecute` для каждой команды. Также, придётся хранить вектор всех совершённых команд, чтобы можно было их отменить.

## Задача 7. Команда и умные указатели

Перепишите код в `command.cpp` так, чтобы он использовал умные указатели `std::unique_ptr` для указания на объекты в куче вместо обычных указателей.