

Семинар #11: Наследование. Домашнее задание.

Задача 1. Фигуры

В файле `shape.cpp` написаны простейшие классы `Circle`, `Rectangle` и `Triangle`, описывающие геометрические фигуры: круг, прямоугольник и треугольник. При написании этих классов наследование не было использовано, в следствии чего некоторые поля и методы одинаковы у всех классов (поле `mPosition` и методы `getPosition` и `setPosition`).

- Создайте класс `Shape`, который бы описывал абстрактную фигуру и содержал бы поля и методы, общие для всех фигур. Измените код классов `Circle`, `Rectangle` и `Triangle` так, чтобы они наследовались от класса `Shape`. Общие для всех фигур поля и методы должны содержаться только в классе `Shape`, но не в классах-наследниках.
- Добавьте метод `void move(Vector2f change)` в класс `Shape`. Этот метод должен изменять поле `mPosition` на значение `change`. Так как остальные классы наследуются от `Shape`, то этот метод можно будет вызвать у всех объектов дочерних классов. Протестируйте этот метод, изменив положения объектов дочерних классов.

Задача 2. Изменение цвета

В файле `draggable.cpp` написан класс `Draggable`, который описывает передвигаемый курсором мыши прямоугольник. Ваша задача – написать класс `DraggableWithColorChange` – наследник класса `Draggable`. Новый класс должен также описывать передвигаемый прямоугольник, но, во время передвижения прямоугольника, его цвет должен меняться на другой. Конструктор нового класса будет иметь вид:

```
DraggableWithColorChange(Vector2f position, Vector2f size, Color baseColor, Color dragColor)
```

Где `baseColor` – это основной цвет прямоугольника, а `dragColor` – цвет прямоугольника при перетаскивании. Протестируйте этот класс в функции `main`.

Задача 3. Окна

В файле `window.cpp` написан класс `BaseWindow`, описывающий простейшее окно. Это окно состоит из двух прямоугольников. Первый прямоугольник определяет границы отрисовки окна, а второй определяет границы области, за которую прямоугольник можно перетаскивать.

- Создай класс `QuestionWindow`, наследник `BaseWindow`. У объектов этого класса, помимо функционала `BaseWindow` должен быть текст, в котором отображается некоторая строка. А также две кнопки внизу: `Ok` и `Cancel`. При нажатии на кнопку `Ok` в консоль должно выводиться строка `Ok`, а при нажатии на кнопку `Cancel` – строка `Cancel`. Используйте класс `Button` из файла `button.hpp` для создания кнопок.
- Создай класс `MessageWindow`, наследник `BaseWindow`. У объектов этого класса, помимо функционала `BaseWindow` должен быть текст, в котором отображается некоторая строка. Также такое окно должно быть одним из 2-х типов: `Error`, `Done`. Окно типа `Error` должно всегда рисоваться оттенком красного цвета, а окно типа `Done` – оттенком зелёного цвета.
- В данной реализации класс `BaseWindow` использует класс `Draggable` через композицию (то есть объект класса `Draggable` является полем класса `BaseWindow`). Измените класс `BaseWindow` так, чтобы он получал все возможности класса `Draggable` через наследование. То есть нужно переписать класс `BaseWindow`, сделав его наследником класса `Draggable`.