



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(национальный исследовательский университет)»

Институт №3 «Системы управления, информатика и
электроэнергетика»

Кафедра 304 «Вычислительные машины, системы и сети»

Пояснительная записка к курсовой работе

по дисциплине: «Программирование»

на тему: «Игровая программа "Фонарик"»

Выполнил:

студент группы МЗО-219БВ-24

Нарзиев А.Т.

Принял:

Ассистент каф.304

Лебедев И.И.

Москва, 2025

Задание

Разработайте интерактивное графическое приложение на языке C++ с применением объектно-ориентированного подхода для визуализации основной фигуры «Фонарик». Иерархия классов должна быть реализована в соответствии с предоставленной в Приложении А UML-диаграммой, включающей как вертикальное, так и веерное наследование.

От базового класса «Фонарик» (BaseFlashlight) должны быть реализованы различные производные классы через вертикальное и веерное наследование, с добавлением и изменением поведения функций базового класса. Для этого используется интерфейс IFlashlight, методы которого изменяются в производных классах, что позволяет отображать фонарик в различных состояниях. Интерфейс базовой фигуры должен содержать набор чисто виртуальных функций, определяющих составные части (корпус, кнопка, головка, свет, трещины) фигуры.

Класс "Конфликтный объект" (Conflict) служит базовым для объектов, взаимодействующих с основной фигурой, при этом результат взаимодействия должен изменять состояние и поведение фонарика. Управление движением основной фигуры и её производных осуществляется через клавиатуру, обеспечивая интерактивность приложения.

Должно быть реализовано как минимум три статических конфликтных объекта с уникальным графическим представлением. От базовой фигуры должны быть созданы веерные потомки, каждый из которых переопределяет одну или несколько виртуальных функций интерфейса, наследуя остальное поведение от базовой иерархии.

Дополнительно необходимо реализовать конфликтные объекты через веерное наследование от базового класса "Конфликтный объект", включая объект, движущийся по кругу (Fly), и объект, автоматически преследующий основную фигуру (Ghost). Траектория преследующего объекта должна рассчитываться с учётом избегания определённых зон, что добавляет сложность взаимодействию.

Программа должна быть написана без дублирования кода, обеспечивая лёгкость сопровождения и модификации, а также возможность быстрого добавления новых классов как в иерархию фигур, так и в иерархию конфликтных объектов. Иерархия должна строиться с использованием механизмов виртуальных функций, динамического полиморфизма, веерного

и вертикального наследования, множественного наследования, интерфейсов и абстрактных базовых классов.

Также должны быть реализованы дополнительные конфликтные объекты, которые обеспечивают развитие сценария и видоизменяют функционал основного объекта, например, батарейки (Battery). Динамический полиморфизм для основной фигуры и конфликтных объектов должен обеспечивать управление поведением объекта на базовом уровне, при этом все производные объекты разных типов должны быть преобразованы через указатели к базовому классу, а обращение к конкретному потомку должно осуществляться через механизм виртуальных функций.

На уровне конфликтных объектов должна быть объявлена виртуальная функция Move, которая будет переопределяться потомками, позволяя каждому потомку реализовывать собственный алгоритм движения, обеспечивая единство управления конфликтными объектами.

Дополнительное задание:

1. Реализовать светящиеся версии фонариков с таймером (LitRectFlashlightWithTimer, LitRoundFlashlightWithTimer)
2. Реализовать механизм постепенного уменьшения таймера свечения
3. Обеспечить переход между обычными и светящимися состояниями фонарика при сборе батареек

Срок окончания: _____ декабря 2025 г.

Руководитель:
Ассистент каф.304 _____ Лебедев И.И.

Исполнитель:
Студент группы М30-219БВ-24 _____ Нарзиев А.Т.

Приложение А

