

КУРСОВАЯ РАБОТА
по дисциплине “Объектно-ориентированное программирование”

ОГЛАВЛЕНИЕ

1. Постановка задачи (вариант №8)	1
а. Описание задания	а
б. Вариант выполнения задания	б

с. Описание иерархии объектов и методов объектов с

2. Описание алгоритма программы 2

Подпункт 1 d

Подпункт 2 e

Подпункт 3 p

3. Скриншоты работы программы (9 файлов) 3

Подпункт 1 g

Подпункт 2 h

Подпункт 3 o

1. Постановка задачи

а. Описание задания.

Круговое движение с вращением составного графического объекта.

б. Вариант выполнения задания.

Обозначим объект, который будет двигаться и вращаться. Для реализации кругового движения (движения по окружности) используем математические (физические) формулы движения точки по окружности $x = x_0 + r\cos(t)$, $y = y_0 + r\sin(t)$, где x_0 , y_0 – центральная точка окружности. Используем для реализации программы библиотеку SFML (<https://en.sfml-dev.org/>), которая предоставляет пользователю удобный интерфейс для работы с окном приложения и простейшими геометрическими фигурами.

Т. к. математические формулы движения по окружности подразумевают точку, а не объект с заданным размером, то сведем задачу к работе с точкой. Но для начала определим размер и форму фигуры. Фигура составная, следовательно, строится путем комбинации простейших геометрических фигур. В этом варианте задания я реализую нечто похожее на космический корабль, как показано в примере. И оно будет двигаться и вращаться. Для двигателей используем прямоугольники, для “”тела” корабля – круг, для “носа” – треугольник. Для удобства при проверке выхода за границы видимой области обозначим форму фигуры как прямоугольник наименьшей площади, такой, что в него помещается вся наша составная фигура. Обозначим точку, которая находится на пересечении левой и верхней сторон нашего прямоугольника. Она и будет двигаться по окружности.

Для того, чтобы наша фигура не уходила за границы окна, зададим рабочую область. Т. к. мы работаем с точкой на левом верхнем углу фигуры, импровизированного прямоугольника, то ограничим левую и нижнюю части окна в использовании. Вычтем из реальных высоты и длины окна высоту и длину заданной фигуры, и получим размер области, с которой будем работать. Кстати, можно урезать рабочую область видимости до квадрата, у которого стороны будут равны длине наименьшей стороны ещё неурезанной рабочей области. Квадрат будет наименьшей площади, такой, что в него помещается вся окружность. И для более приятного глазу вида можно центрировать урезанную рабочую область относительно неурезанной. Таким образом, картинка будет находится по центру, а сама окружность будет смещена чуть левее и выше, как раз на размер фигуры. Для реализации работы с урезанной рабочей областью, меньшей, чем размер окна, создадим отдельный класс.

с. Описание иерархии объектов и методов объектов.

Ссылка на ресурс, где размещена иерархия объектов и методов объектов:

<https://achor.com/>. Также можно посмотреть на картинке:

(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)
(КАРТИНКА С ИЕРАРХИЕЙ КЛАССОВ И МЕТОДОВ КЛАССОВ)

А ЗДЕСЬ ДЛИННОЕ ОПИСАНИЕ ИЕРАРХИИ ТЕКСТОМ.