

Лабораторная работа №1

«Разработка, отладка и выполнение программы с использованием процедур и записей»

Цель работы: Разработка, отладка и выполнение программы с использованием процедур и записей. Дополнительно изучить основы объектно-ориентированного программирования, а также ознакомиться с библиотеками для создания оконных приложений в QT C++.

Программа должна обеспечивать ввод/вывод данных с помощью визуальных компонентов, контроль ошибок при вводе, обработку исключительных ситуаций. Каждый класс необходимо помещать в отдельный модуль, т.е. разбить программу по файлам.

Создать оконное приложение, которое позволит пользователю создавать фигуры и производить с ними указанные в заданиях операции.

Задание 1.

Описать класс-родитель и класс-потомок, имеющие методы, указанные в соответствующем варианте задания (потомок наследует или переопределяет методы родителя и приобретает новые). Предусмотреть необходимое количество кнопок для демонстрации каждого из методов объектов.

Данные задания выполнять в соответствии с выданным вариантом:

1. Нарисовать вращающееся колесо. Родительский класс – перемещающийся круг.
2. Нарисовать повозку (прямоугольник на 2 колесах). Родительский класс – перемещающийся прямоугольник.
3. Нарисовать ракету с пламенем из сопла. Родительский класс – перемещающийся отрезок.
4. Нарисовать лицо двигающую глазами и открывающую рот. Родительский класс – перемещающийся эллипс.
5. Нарисовать солдата, перемещающегося и отдающего честь. Родительский класс – перемещающийся прямоугольник.
6. Нарисовать корабль, который может поднимать флаг. Родительский класс – перемещающийся прямоугольник.
7. Нарисовать автомобиль с открывающимися дверями и включающимися фарами. Родительский класс – перемещающийся прямоугольник.
8. Нарисовать сигнальщика, подающего различные сигналы. Родительский класс – перемещающийся прямоугольник.
9. Нарисовать самосвал, который может поднимать кузов. Родительский класс – перемещающийся прямоугольник.
10. Нарисовать самолет, который может при посадке выпускает шасси. Родительский класс – перемещающийся прямоугольник.
11. Нарисовать домик, в котором открываются двери и окна. Родительский класс – перемещающийся прямоугольник.
12. Нарисовать паровоз, который выпускает дым. Родительский класс – перемещающийся прямоугольник.

13. Нарисовать воздушный шарик, который может лопнуть. Родительский класс – перемещающийся эллипс

14. Нарисовать лифт, который доставляет людей на нужный этаж. Родительский класс – перемещающийся прямоугольник.

15. Нарисовать тележку, на которой перевозят различные грузы. Родительский класс –
перемещающийся
прямоугольник.

Задание 2.

Реализовать иерархию классов, представляющих плоские геометрические фигуры. Необходимые фигуры: треугольник, круг, ромб, квадрат (отдельный класс), прямоугольник, звезда (пятиконечная, шестиконечная, восьмиконечная), шестиугольник и любая другая фигура не основанная на вышеперечисленных. Каждая фигура должна позволять производить следующие операции (при этом операции в общем виде для произвольной геометрической фигуры можно не реализовывать, реализовать их нужно для каждого конкретного типа фигур):

- нахождение площади и периметра, центра масс (для многоугольников можно воспользоваться триангуляцией);
- изменение координат центра масс (при этом вся фигура сдвигается на столько же, на сколько сдвинут ее центр);
- управление параметрами, специфическими для конкретных типов фигур (координаты вершин многоугольника, радиусы эллипса и т. п.);
- осуществление базовых преобразований: перемещение, поворот относительно точки, масштабирование относительно точки; при этом фигура должна изменяться плавно в течение заданного промежутка времени и на каждый такт изменения генерировать событие (можно воспользоваться Timer);
- возможность рисования фигуры на заданной канве.

Обязательной является реализация многоуровневой иерархии, т.е. когда одни фигуры являются разновидностями других.

Стандартный **плохой** пример: прямоугольник – разновидность многоугольника, а квадрат – разновидность прямоугольника. Пример **плохой** т.к. принцип инвариантности не соблюден.