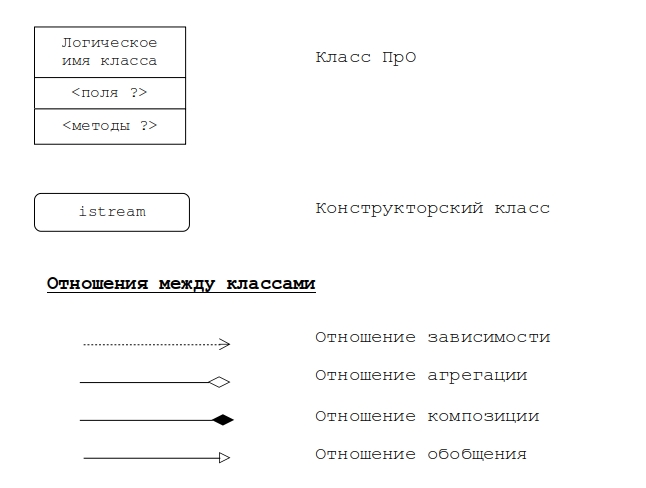
**Методические указания к выполнению лабораторных работ по курсу «Объектно-ориентированное программирование»**

Цикл состоит из 8 лабораторных работ, охватывающий все разделы курса «Объектно-ориентированного программирования».

Для каждой работы приводится содержательное описание предметной области. Формулируются цель работы и ограничения, в рамках которых эта цель должна быть реализована.

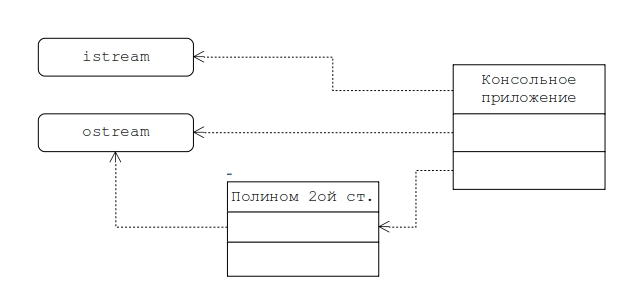
В качестве исходных данных для программирования в объектно-ориентированное парадигме (ООП) выступает объектная модель (ОМ) предметной области (ПрО), дополненная объектной библиотекой (ООБ) конструкторских классов (КК). ОМ в данном лабораторном цикле представлен в виде диаграммы классов (ДК), которую следует реализовать, заполнив классы ПрО необходимыми полями и методами, обоснованно распределенными по существующим областям видимости. В качестве КК на ДК используются стандартные классы библиотеки С++ и классы из ООБ Qt. На ДК используются следующие обозначения:



На диаграммах классов могут быть не прорисованы некоторые используемые в качестве источников зависимости конструкторские классы из используемых ООБ.

Для выполнения лабораторных работ нет ограничений для выбора языка, среды и объектной библиотеки. Однако ОМ должна быть реализована в точном соответствии с приведенной ДК. Допускаются расширение состава или замена используемых КК с сохранением их функционального назначения.

**Лабораторная работа №1**

****

Создать консольное приложение для вычисления корней и для заданного аргумента значения полинома 2-ой степени на множестве вещественных чисел. Приложение должно включать основной модуль (функция main), модуль «application» и модуль «polinom».

Основной модуль main.cpp консольного приложения может иметь вид:

#include "application.h"

int **main**(int argc, char \*argv[])

{

TApplication a(argc, argv);

return a.exec();

}

Модуль «application» содержит описание и реализацию членов класса ***TApplication***, который предназначен для выполнения функций ввода коэффициентов полинома, значения аргумента, инициализации процесса вычисления корней, инициализации процесса вычисления значения и вывода результатов.

Метод ***exec()*** класса ***TApplication*** управляет выводом в консоль меню команд и завершением приложения и можетиметь примерно следующую реализацию

int TApplication::**exec**()

{

int ch;

while (true)

{

ch = menu();

if (ch == 0) break;

switch (ch) {

case 1:

// ... to do

break;

default:

break;

}

}

return 0;

}

Модуль «polinom» содержит описание и реализацию членов класса ***TPolinom***, необходимых для выполнения цели разрабатываемого приложения. Описание класса должно использовать вместо double (вещественное число, заданное в условии) абстрактный тип ***number***, описание которого должно задаваться в отдельном заголовочном файле number.h с помощью оператора

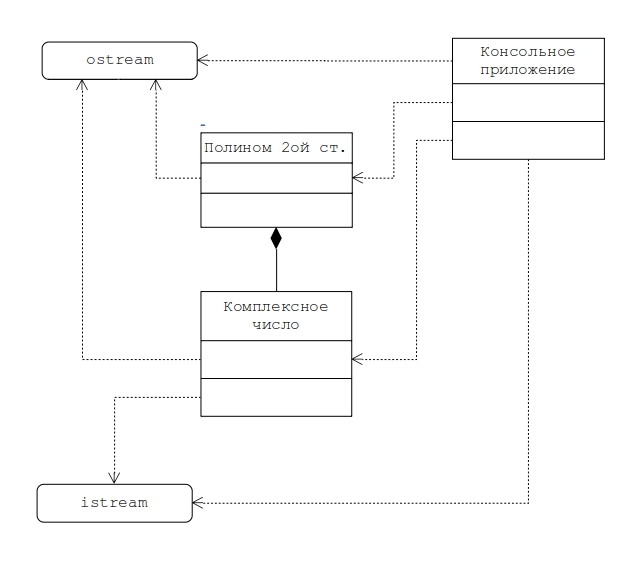
**typedef double number;**

Требуется реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям.

Разработать контрольные примеры и оттестировать на них программу.

Оформить отчет, сделать выводы по работе.

**Лабораторная работа №2**

****

Создать консольное приложение для вычисления корней и значения полинома 2-ой степени на множестве комплексных чисел. Приложение должно включать основной модуль, модуль «application», модуль «polinom» и модуль «complex».

Для этого в проект лабораторной работы №1 следует добавить модуль с описанием и реализацией собственного класса комплексных чисел TComplex. При этом основной модуль, модуль «application» и модуль «polinom» не должны изменяться. Изменения вносятся лишь в заголовочный файл number.h, где

**typedef double number;**

следует заменить на

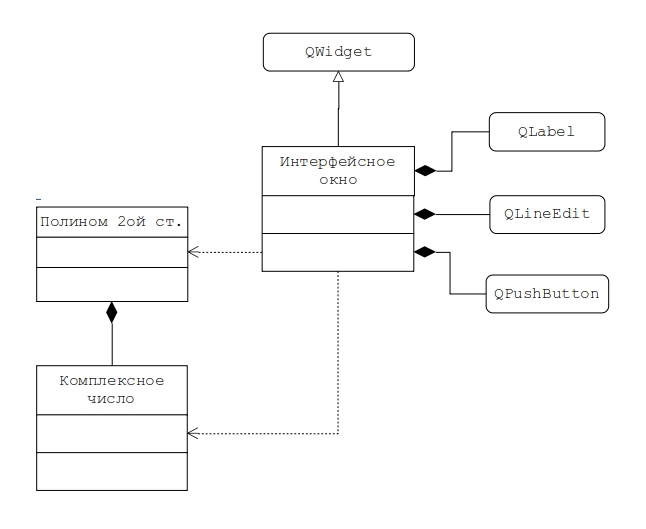
**#include «TComplex»;**

**typedef TComplex number;**

В классе **TComplex** следует определить только те члены класса и декларации, которые необходимы для совместимости модулей проекта и реализации отношений, приведенных в ДК объектной модели.

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленной цели. Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

**Лабораторная работа №3**

****

Создать GUI приложение для вычисления корней и значения полинома 2-ой степени на множестве комплексных чисел. Приложение должно включать основной модуль, модуль «interface», модуль «polinom» и модуль «complex».

Основной модуль main.cpp GUI приложения может иметь вид:

#include <QApplication.h>

int **main**(int argc, char \*argv[])

{

QApplication a(argc, argv);

TInterface interface;

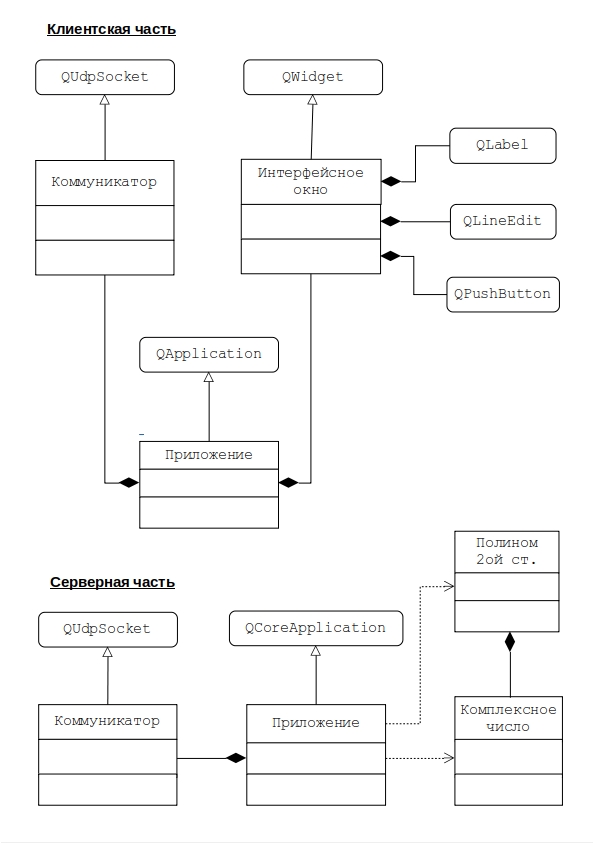
interface.show();

return a.exec();

}

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

**Лабораторная работа №4**

****

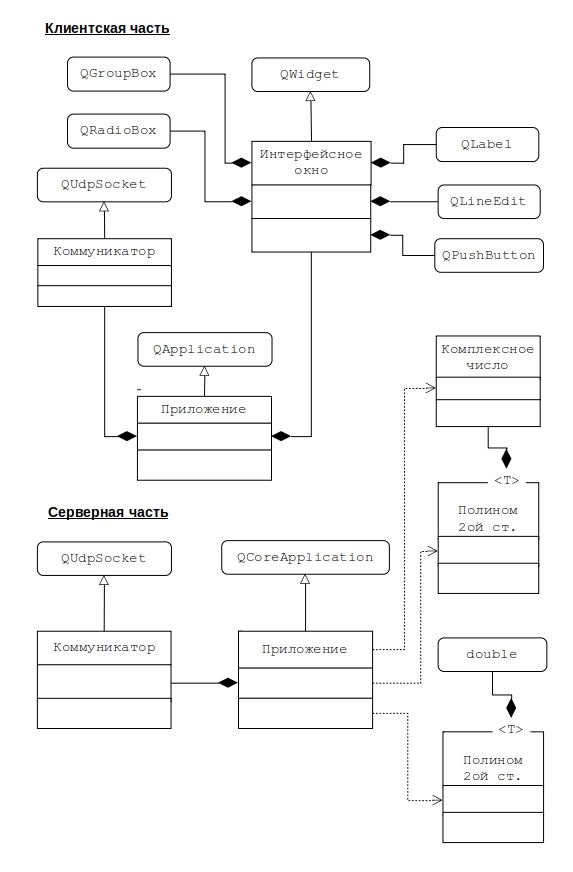
Создать распределенное приложение, включающее клиентскую и серверную части, взаимодействующие посредством сетевого обмена сообщениями.

Клиентская часть представляет собой GUI приложение, реализующее интерфейс аналогичный работе №4.

Серверная часть представляет собой консольное приложение, предназначенное для вычисления корней и значения полинома 2-ой степени на множестве комплексных чисел.

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

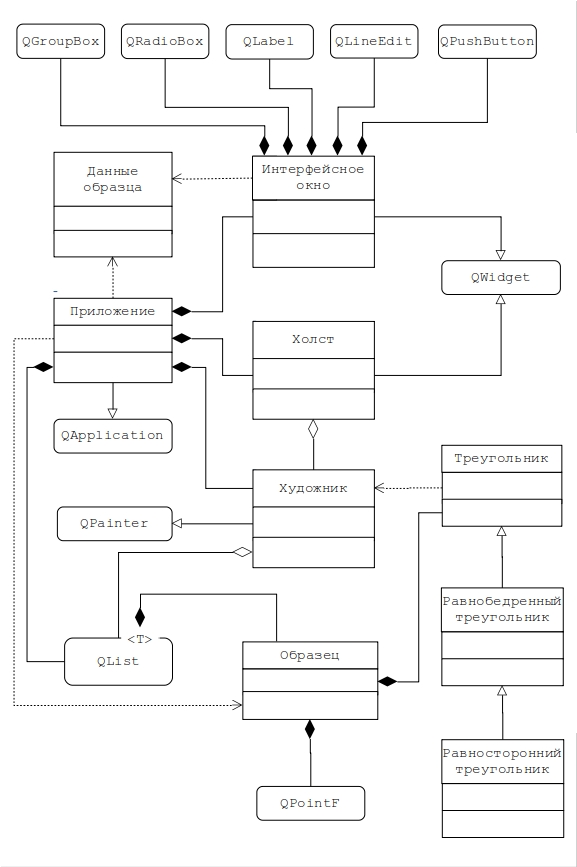
**Лабораторная работа №5**

****

Создать консольное приложение для вычисления корней и значения полинома 2-ой степени на множестве как вещественных так и комплексных чисел. Для этого преобразовать класс **TPolinom** в параметризованный класс, в котором абстрактный тип **number** использовать как параметр класса.

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

**Лабораторная работа №6**

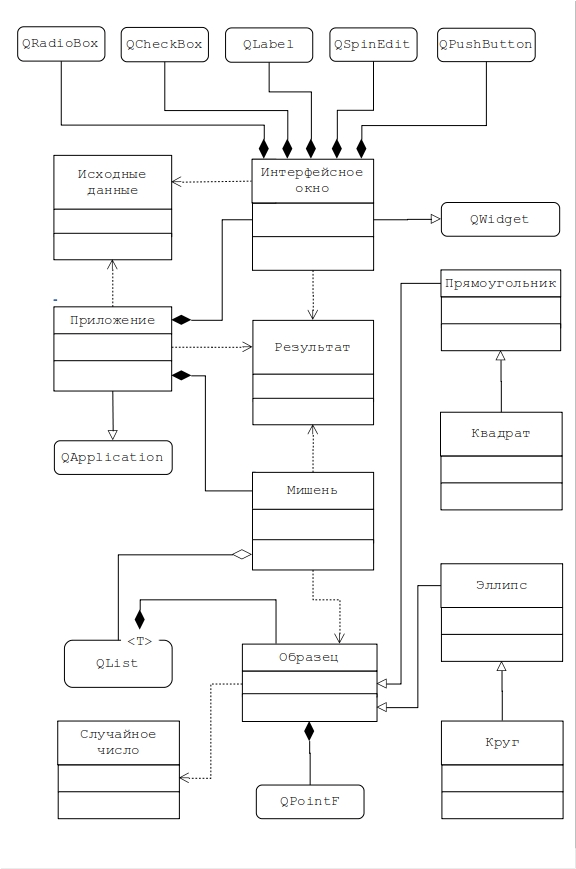


На холсте фиксированного размера художник рисует контуры следующих геометрических фигур (детали рисунка): «Равнобедренный треугольник», «Равносторонний треугольник» и «Треугольник». Эти детали отрисовываются последовательно в произвольном порядке и в любой последовательности. Для каждой из них задается точка привязки к холсту, которая однозначно связана с одной из вершин (повороты деталей отсутствуют).

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям.

Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.

**Лабораторная работа №7**



Задана прямоугольная мишень со сторонами w и h. Заданы 4 вида фигур, обладающих свойствами осевой симметрии: квадрат, эллипс, круг и прямоугольник. Пользователем выбираются фигуры с задаваемыми размерами. Эти фигуры набрасываются на мишень так, что их точка симметрии попадает в задаваемую пользователем точку мишени, а ориентация осей симметрии при этом не нарушается (без вращения).

Требуется написать программу, которая определит оставшуюся свободную долю мишени двумя способами: регулярным и случайным.

Суть регулярного способа заключается в наложении на мишень равномерной сетки с i узлов по горизонтали и j узлов по вертикали.

Суть случайного способа заключается в набрасывании на мишень i\*j равномерно распределенных на контуре мишени узлов.

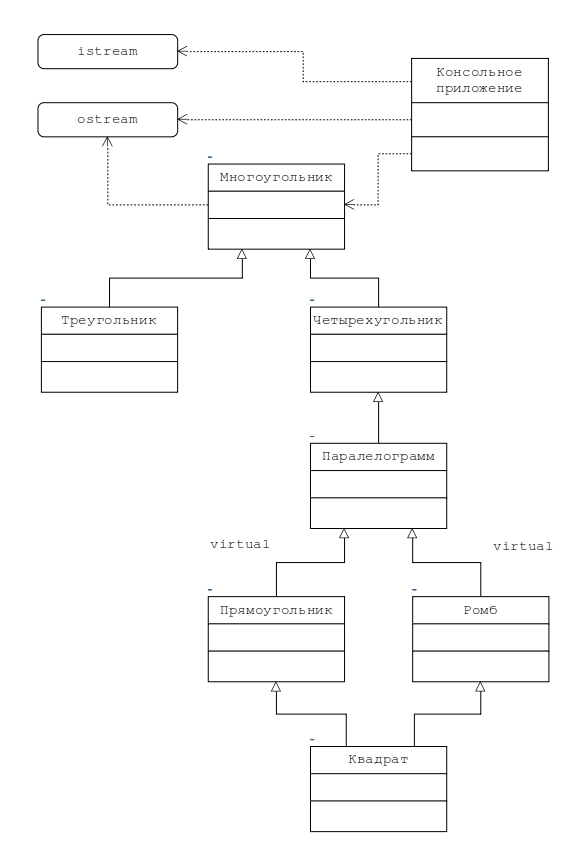
Для обоих способов доля узлов не попадающих к контур всех фигур определяет свободную долю мишени, точность которой зависит от количества тестовых узлов.

Обязательным условием является отсутствие отрисовки мишени и самих фигур с последующим подсчетом закрашенных пикселей изображения.

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям.

Разработать контрольные примеры и оттестировать на них программу. На примерах с возможным аналитическим решением сравнить точности обоих способов. Оформить отчет, сделать выводы по работе.

**Лабораторная работа №8**



Заданы следующие понятия «Четырехугольник», «Ромб», «Квадрат», «Параллелограмм» и «Прямоугольник». Установить иерархию этих понятий и определить соответствующие ей классы. Создать объекты этих классов, рассчитать их площади и периметры. Представить полную объектную модель разработанной программы.

Реализовать и отладить программу, удовлетворяющую сформулированным требованиям и заявленным целям. Разработать контрольные примеры и оттестировать на них программу. Оформить отчет, сделать выводы по работе.