**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Информационных систем**

**ОТЧЕТ**

**по практической работе №8**

**по дисциплине «Объектно-ориентированное программирование»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 8374 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Пихтовников К.С. |
| Студент гр. 8374 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Подсекин Г.С. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Егоров С.С. |

Санкт-Петербург

2021

**Задание на практическую работу**

Заданы следующие понятия: «Треугольник», «Окружность», «Геометрическая фигура», «Эллипс», «Четырехугольник», «Ромб», «Квадрат», «Параллелограмм», «Прямоугольник», «Равносторонний треугольник», «Равнобедренный треугольник». Установить иерархию этих понятий и определить соответствующие ей классы.

Создать консольное приложение, содержащее меню команд создания объектов этих классов с вводимыми пользователем параметрами, для которых рассчитываются и выводятся значения их площади и периметра.

Представить диаграмму классов объектной модели разработанного приложения.

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

**Спецификация классов**

Класс TApplication

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод TApplication(); | Область видимости public.  Конструктор класса |
| Метод void exec(); | Формальных параметров нет, тип void, область видимости-public.  В этом методе идет вызов функции menu(), задается конкретное действие, которое пожелал сделать пользователь и результат выводится на экран. |

*Таблица 1. Класс TApplication*

Класс Quadrangle

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод:  explicit Quadrangle (const double &a, const double &b, const double &c, const double &alpha, const double &beta); | Тип формальных параметров - const double &, const double &, const double &, const double&, const double&.  Область видимости — public.  Метод позволяет записать введенные значения и посчитать площадь и периметр, наследуя функции класса Polygon |

*Таблица 2. Класс Quadrangle*

Класс Polygon

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Аттрибуты:  QList<double> \_edge;  QList<double> \_angle;  QList<double> \_iEdge;  QList<double> \_iAngle; | Область видимости protected. Переменные для подсчета площади и периметра фигуры |
| Методы Poligon(); ~Polygon(); | Формальных параметров нет, область видимости public.  Конструктор и деструктор класса |
| Метод double area(); | Формальных параметров нет, область видимости public.  Метод позволяет посчитать площадь фигуры |
| Метод double perimetr(); | Формальных параметров нет, область видимости public.  Метод позволяет посчитать периметр фигуры |
| Метод void calcInternal(); | Формальных параметров нет, область видимости protected.  Метод позволяет посчитать площадь фигуры через угол |

*Таблица 3. Класс Polygon*

Класс Parallelogram

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод:  Parallelogram (const double &a, const double &b, const double &alpha); | Тип формальных параметров - const double &, const double &, const double&.  Область видимости — public.  Метод позволяет записать введенные значения и посчитать площадь и периметр, наследуя функции класса Quadrangle |

*Таблица 4. Класс Parallelogram*

Класс Rhombus

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод:  Rhombus (const double &a, const double &alpha); | Тип формальных параметров - const double &, const double &.  Область видимости — public.  Метод позволяет записать введенные значения и посчитать площадь и периметр, наследуя функции класса Parallelogram |

*Таблица 5. Класс Rhombus*

Класс Square

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод:  Square (const double &a); | Тип формальных параметров - const double &.  Область видимости — public.  Метод позволяет записать введенные значения и посчитать площадь и периметр, наследуя функции классов Rectangle и Rhombus |

*Таблица 6. Класс Square*

Класс Triangle

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод:  Triangle (const double &a, const double &b, const double &angle); | Тип формальных параметров - const double &, const double &, const double &.  Область видимости — public.  Метод позволяет записать введенные значения и посчитать площадь и периметр, наследуя функции класса Polygon |

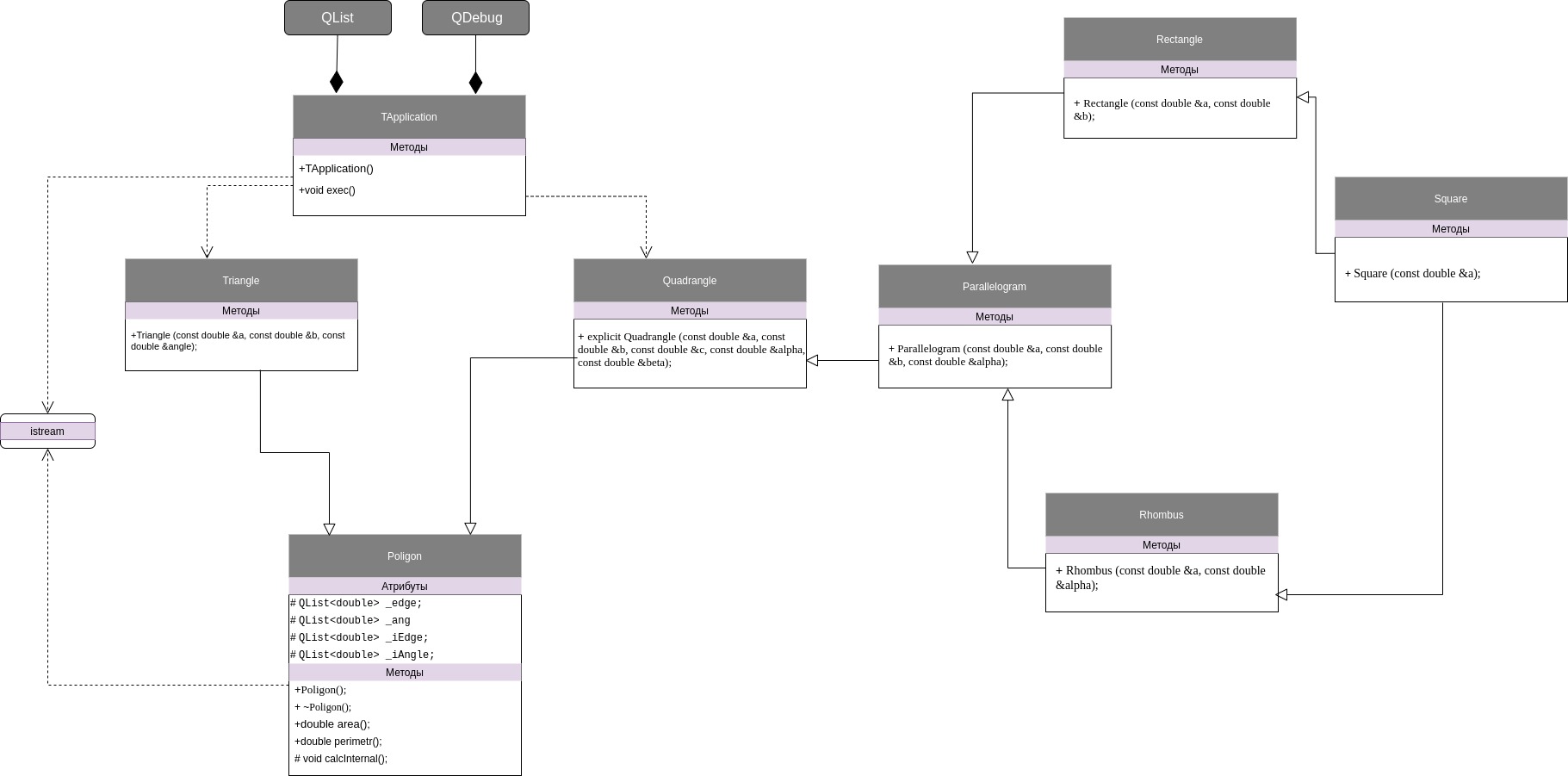
*Таблица 7. Класс Triangle*

Класс Rectangle

|  |  |
| --- | --- |
| Метод/атрибут | Описание |
| Метод:  Rectangle (const double &a, const double &b); | Тип формальных параметров - const double &, const double.  Область видимости — public.  Метод позволяет записать введенные значения и посчитать площадь и периметр, наследуя функции класса Parallelogram |

*Таблица 8. Класс Rectangle*

**Диаграмма классов**

**

*Рис.1. Реализация диаграммы классов клиентской части*

|  |  |
| --- | --- |
| Символ | Значение |
| + | public - открытый доступ |
| - | private - только из операций того же класса |
| # | protected - только из операций этого же класса и классов, создаваемых на его основе |

*Таблица 9. Обозначение аттрибутов и методов класса*

**Описание контрольного примера с исходными и ожидаемыми (расчетными) данными**

**Пример 1 (треугольник):**

Исходные данные:

a = 4

b = 3

угол = 90

Ожидаемые данные:

Площадь - 6

Периметр - 12

**Пример 2 (ромб):**

Исходные данные:

a = 3

угол = 80

Ожидаемые данные:

Площадь - 8.9

Периметр - 12

**Пример 3 (прямоугольник):**

Исходные данные:

a = 5

b = 6

Ожидаемые данные:

Площадь - 30

Периметр - 22

**Пример 4 (квадрат):**

Исходные данные:

a = 10

Ожидаемые данные:

Площадь - 100

Периметр - 40

**Пример 5 (четырехугольник):**

Исходные данные:

a = 7

b = 9

с = 5

угол 1 = 180

угол 2 = 30

Ожидаемые данные:

Площадь - 20

Периметр - 33

**Пример 6 (параллелограм):**

Исходные данные:

a = 50

b = 10

угол = 30

Ожидаемые данные:

Площадь - 1082.5

Периметр - 186.6

**Пример 7 (многоугольник):**

Исходные данные:

a = 1

b = 2

c = 4

d = 5

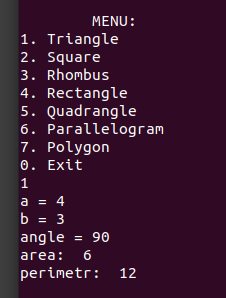
Ожидаемые данные:

Площадь - 0.07

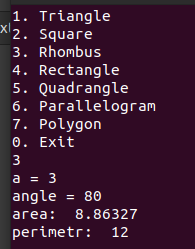
Периметр - 4

**Скриншоты программы на контрольных примерах**

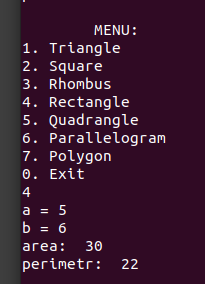
**Пример 1:**

****

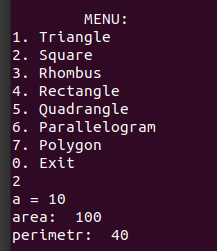
**Пример 2:**



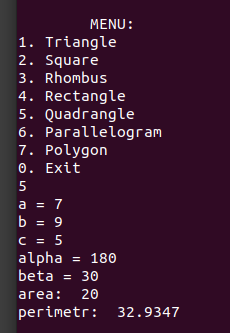
**Пример 3:**

****

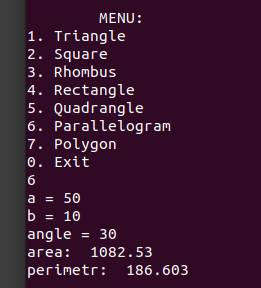
**Пример 4:**

****

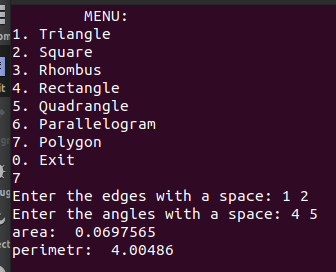
**Пример 5:**

****

**Пример 6:**

****

**Пример 7:**

****

**Вывод**

В ходе данной лабораторной работы было создано консольное приложение, содержащее меню команд создания объектов этих классов с вводимыми пользователем параметрами, для которых рассчитываются и выводятся значения их площади и периметра. На рис.1 представлен макет диаграммы классов приложения, который требуется реализовать в приложении.

Помимо этого, была произведена отладка работы программы. Разработаны контрольные примеры с исходными и ожидаемыми данными, которые затем были протестированы в созданном консольном приложении. Все результаты совпали.