**Объектно-ориентированное программирование**

**Домашнее задание №1, модуль 3**

Написать класс ***CMy2DPoint***, описывающий точку на плоскости. Написать класс ***CMy2DObjectНомерВарианта***, описывающий геометрическую фигуру, соответствующую варианту задания. Продемонстрировать работоспособность разработанных классов с помощью небольшой консольной программы.

Для ***CMy2DPoint*** необходимо перегрузить операторы ‘+’, ’-‘, ’\*’, ’/’, ’==’, ‘!=’, ‘<<’, ‘>>’. Оператор ‘\*’ должен работать для следующих пар операндов: (***CMy2DPoint, CMy2DPoint***), (***double, CMy2DPoint***), (***CMy2DPoint, double***). В первом случае результатом операции является скалярное произведение, в остальных – произведение вектора на число. Операторы ввода/вывода проверить, записывая и считывая данные в текстовый файл и на консоль.

Во всех вариантах задания фигура описывается четырьмя размерами (A, A1, A2, A3), координатами центра (m\_centre) и углом поворота вокруг центра (m\_dAngle). Координатами фигуры считаются координаты центра квадрата, в который она вписана, эта же точка является центром вращения (см. рис. 1). За координаты фигуры должно отвечать поле ***m\_center*** типа ***CMy2DPoint*** . Все переменные класса должны быть закрытыми. Класс должен иметь конструктор с параметрами, позволяющий инициализировать все закрытые поля. Разработанный класс должен иметь следующие методы:

1. Методы доступа (Get и Set) ко всем закрытым полям, методы Set должны осуществлять проверку корректности данных (граница фигуры не должна самопересекаться).
2. Метод ***void Move(double dX, double dY, double dAngle = 0)*** – Смещение объекта на вектор dX, dY и поворот на dAngle;
3. Метод ***void Move(const CMy2dPoint &delta, double dAngle = 0)*** – Смещение объекта на вектор дельта и поворот на dAngle;
4. Метод ***void Rotate(double dAngle)*** – поворот фигуры на угол dAngle;
5. Метод ***bool IsInside(const CMy2dPoint &point)*** – возвращает true если точка лежит внутри фигуры и false в противном случае.
6. Метод ***bool IsInside(double x, double y)*** – возвращает true если точка с координатами (x,y) лежит внутри фигуры и false в противном случае.

Методы не меняющие данные класса должны быть константными.

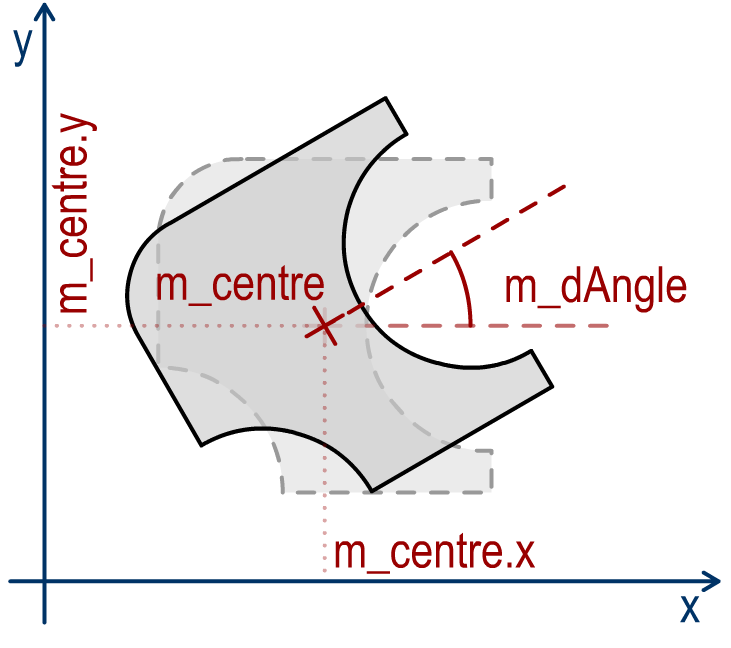


Рис. 1. Пояснение к задаче (фигура соответствует варианту A5 => имя класса: ***CMy2DObjectA5***)

Варианты:

|  | A | B |
| --- | --- | --- |
| 1. | D:\Dropbox\Aksenov_Bober\Задачи по C++\111.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\411.JPG |
| 2. | D:\Dropbox\Aksenov_Bober\Задачи по C++\112.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\412.JPG |
| 3. | D:\Dropbox\Aksenov_Bober\Задачи по C++\113.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\413.JPG |
| 4. | D:\Dropbox\Aksenov_Bober\Задачи по C++\114.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\414.JPG |
| 5. | D:\Dropbox\Aksenov_Bober\Задачи по C++\121.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\421.JPG |
| 6. | D:\Dropbox\Aksenov_Bober\Задачи по C++\122.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\422.JPG |
| 7. | D:\Dropbox\Aksenov_Bober\Задачи по C++\123.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\423.JPG |
| 8. | D:\Dropbox\Aksenov_Bober\Задачи по C++\124.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\424.JPG |
| 9. | D:\Dropbox\Aksenov_Bober\Задачи по C++\131.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\431.JPG |
| 10. | D:\Dropbox\Aksenov_Bober\Задачи по C++\132.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\432.JPG |
| 11. | D:\Dropbox\Aksenov_Bober\Задачи по C++\133.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\433.JPG |
| 12. | D:\Dropbox\Aksenov_Bober\Задачи по C++\134.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\434.JPG |
| 13. | D:\Dropbox\Aksenov_Bober\Задачи по C++\141.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\441.JPG |
| 14. | D:\Dropbox\Aksenov_Bober\Задачи по C++\142.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\442.JPG |
| 15. | D:\Dropbox\Aksenov_Bober\Задачи по C++\143.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\443.JPG |
| 16. | D:\Dropbox\Aksenov_Bober\Задачи по C++\144.JPG | D:\Dropbox\Aksenov_Bober\Задачи по C++\444.JPG |