

==== Условия задачи ====

Есть файл, в котором записаны геометрические объекты в бинарном виде. Интерфейс для чтения этого файла -- класс DataProvider. Общий формат файла известен:

```
---начало файла---  
[число объектов в файле = N]  
[объект1]  
[объект2]  
...  
[объектN]  
---конец файла---
```

Объекты записаны в следующем виде ([объект] =):

```
---начало объекта---  
[тип объекта]  
[размер данных объекта]  
[данные объекта]  
---конец объекта---
```

Известен также формат для некоторых типов объектов: квадрат, окружность, многоугольник, ломаная, дуга окружности (см. DataProvider.cpp). Но общее количество типов объектов и их формат неизвестны на данный момент. Они станут известны в будущем в процессе разработки продукта.

Есть интерфейс для отрисовки объектов -- WDraw. Основной метод отрисовки -- drawSegment.

==== В рамках реализации задачи необходимо: ====

Построить расширяемый Framework для объектов, которые читаются из DataProvider-а и рисуются в WDraw. В результате должно быть также реализовано следующее:

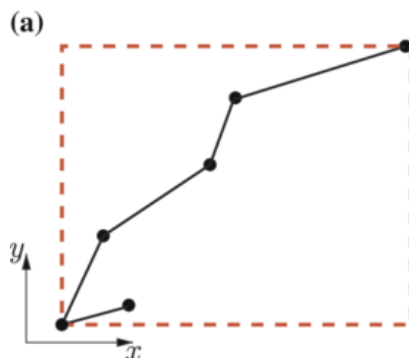
1. Отрисовка вызывается для каждого объекта во входном файле.
2. Для каждого объекта во входном файле определить boundingBox --(см. рисунок AABV ниже).
3. Для каждого объекта во входном файле посчитать общую длину линейаризованного представления = сумма длин сегментов, которыми отрисовывается объект (см. WDraw::drawSegment)

Примечания*:

- Нужно учесть, что фреймворк будет расширяться и дополняться как объектами, так и функциональностью.
- Для отрисовки объектов нужно использовать методы WDraw.
- В процессе работы можно менять существующий код, если необходимо.
- Желательно наличие unit-тестов.

На выходе нужно предоставить *.exe-файл, при запуске которого демонстрируется работа фреймворка. Основной метод просмотра результата -- debugger в Visual Studio.

Исполнение задачи не ограничено жесткими временными рамками, однако сроки должны быть рационально соответствовать количеству и качеству кода. Больше внимание при анализе результатов будет отдаваться логичности, оправданности и чистоте кода, расширяемости системы, доступности кода для понимания другими разработчиками, соответствию решения поставленным задачам и т.д.



*Пояснение некоторых моментов.

В первом задании не будет реальной отрисовки объектов в графическом виде. На данный момент «отрисовка» - это вывод в консоль координат объектов.

По поводу расширяемости программы. Программа должна быть написана так, чтобы при необходимости изменить формат графического вывода (например, рисовать в окно OpenGL или SFML) нужно было добавить только новые классы (файлы), описывающие работу с конкретной графической библиотекой, а в основном проекте достаточно было изменить только одну строку.

Чтение также должно быть расширяемым. Например, на данном этапе вместо файла с данными используется массив. Нужно сделать так, чтобы при необходимости можно было читать данные из файла или интернета – то есть разработать интерфейсы, которые не надо будет переписывать, а только добавлять новые файлы с требуемой реализацией.

Рекомендуется перед началом выполнения задания изучить существующие паттерны проектирования: <https://refactoring.guru/ru/design-patterns/cpp>