**Определения**

Барьер - замкнутая возможно самопересекающаяся ломаная (тип  
"многоугольник") или незакмнутая возможно самопересекающаяся ломаная  
(тип "стена"). Для многоугольника определено понятие внутренности:  
если луч, выходящий из заданной точки не параллельный ни одному ребру  
и не проходящий ни через одну вершину пересекает нечетное число ребер,  
то точка внутри многоугольника, если четное - то снаружи. Границу  
многоугольника можно считать его внутренностью. Многоугольники могут  
быть пересекающимися, тут никаких ограничений нет.

**Оганичения**  
Код должен быть на C++. Никаких дополнительных библиотек, кроме stl и tinyxml использовать  
нельзя, поскольку будут проблемы с портированием под Android.

**Задача**

Разработать классы Barrier, BarrierMap, которые будут позволять решать  
следующие задачи:  
  
1) Парсить xml-файлы config.xml (извлечение идентификаторов локаций и  
подлокаций, размеров подлокаций) и barriers.xml, используя библиотеку  
tinyxml (<http://www.grinninglizard.com/tinyxmldocs/index.html>)  
  
static BarrierMap parse(const std::string& configXml, const  
std::string& barriersXml)  
  
2) Находить барьер по паре (subLoc, barrierId)  
Barrier getBarrier(int subLoc, int id);  
  
3) Находить все барьеры для подлокации  
std::vector<Barrier> getBarriers(int subLoc);  
  
4) Находить все барьеры, содержащие заданную точку:  
std::vector<Barrier> getBarriers(int subLoc, double x, double y);  
  
5) Определять пересекает ли заданный отрезок какой-нибудь барьер и  
находить ближайшую точку пересечения:  
  
double intersects(int subLoc, double x1, double y1, double x2, double y2);  
  
Возвращаемое значение:  
NAN - ошибка (например, subLoc не найден)  
0    - точка (x1,y1) лежит внутри барьера;  
1    - отрезок (x1,y1) - (x2,y2) не пересекает барьер;  
k \in (0,1) - отрезок (x1,y1) - (x2,y2) пересекает барьер в точке (x1  
+ k\*(x2-x1), y1 + k\*(y2-y1)).  
  
6) Метод toString(), сериализующий BarrierMap обратно в XML-формат barriers.xml:  
  
std::string toString()const;  
  
В качестве образца оформления кода можно взять интерфейс класса  
ZoneMap (файл приложен).  
Примеры barriers.xml и config.xml приложены.  
  
Barrier может быть двух типов:  
- тип "многоугольник" - замкнутая ломаная, последняя точка соединяется  
с первой, внутренность многоугольника - принадлежит барьеру (при  
самопересечении ломаной внутренность определяется используя  
"шахматный" принцип)  
- тип "стена" - незамкнутая ломаная, внутренности нет, но для метода  
intersects стены надо учитывать.