### Задание 1 для первого семестра 2 курса

Требуется разработать и реализовать несколько простейших классов на C++ и написать тесты для проверки работы этих классов.

**Требования к реализации**

0. Реализация должны состоять из нескольктх файлов. Нужно уметь использовать собственные заголовочные файлы, составлять простейшие make-файлы и использовать их для сборки, а также уметь собирать и отлаживать программу из командной строки (в том числе с использованием gdb).

1. Все реализации должны удовлетворять согласованным неформальным интерфейсам и иметь заголовочный .h файл с прокомментированным описанием методов класса. Разработка формального описания классов предоставляется студенту, при этом должны быть предусмотрены способы реакции и оповещения о некорректных или ошибочных ситуациях.

2. К заданию должны быть написаны тесты, т.е. отдельные программы которые вызывают требуемые функции с разнообразными наборами данных и (если возможно, то автоматически) проверяют правильность результата.

3. Поскольку детальная реализация предлагаемых задач может оказаться весьма объемной, то не нужно пытаться реализовать все возможные операции. Но при этом реализованная часть все же должна поддерживать некоторые вполне содержательные задачи (тесты) и должно быть ясно как при необходимости можно расширить реализацию до необходимого уровня доопределением дополнительных функций и операций.

**Возможные варианты задания** Данные формулировки задают только направление работы. Конкретизация постановок, уточнение и т.д. проводятся в рабочем порядке с преподавателем.

Цель задания — знакомство с формализмом определения классов C++, перегрузка опреаторов.

Предлагается разработать систему классов для работы с геометрическими объектами: точка, вектор, отрезок, прямая, луч, угол, плоскость, многоугольник и т.д.

Особенностью постановки является “защита координат объектов”, т.е. координаты помещаются в private секции каждого объекта, инициализируются только в конструкторах и могут модифицироваться только в результате естественных операций с объектами. Возможен также вариант, когда координаты запрещается изменять, а необходимые модификации по смыслу решаемых задач реализуются созданием новых объектов.

Под естественными операциями понимаются линейные комбинации объектов, их традиционные преобразования (сдвиг, поворот, растяжение), проверка совпадения, принадлежности, пересечения, вычисление расстояний и других характеристик и т.п. Эти операции должны быть реализованы перегрузкой подходящих по символике операторов.

Кроме этого каждый реализованный геометрический класс дожен иметь метод, позволяющий его отрисовать, например, в смысле использования утилиты gnuplot. В этом случае, можно будет визуализировать последовательность и результат построений требуемых геометрических объектов.

В качестве задач можно взять подходящие задачи по геометрии для 1 курса с соответствующими модификациями — на входе алгоритма (в параметрах функции) задаются исходные объекты, на выходе тоже получается геометрический объект.

Например, задача “найти угол, под которым данный многоугольник виден и данной точки” может реализовываться функцией с таким заголовком:

```
Angle GetViewAngle (const Point &pt, const Polygon &poly);
```

Отдельный вопрос, на который следует обратить внимание, это существование результата геометрической операции. Например, точка пересечения отрезков может не существовать. Что в таком случае должна возвращать операция пересечения отрезков?

Один из вариантов — добавить в каждый объект булевский флаг корректности его состояния. Другой вариант — определить специальные значения (например, координат), которые будут считаться некорректными, и уже по ним устанавливать и проверять коррекность состояния.

В примерах, которые будут постепенно дополняться и корректироваться, проверка корректности возлагается на функцию IsDef() от слова definite.

Попытка выполнить содержательную операцию с несуществующим объектом должна вызывать исключение.

В реализации следует добиваться максимального использования перегруженных операций с объектами, а не прямого вычисления ответа по исходным координатам.

На сайте выложены примеры построения геометрических классов. Их не надо воспринимать как догму, а скорее как иллюстрацию возможных решений, поскольку для конкретной студенческой задачи многое из

этих примеров и не потребуется. Поэтому при работе над задачей имеет смысл разобраться в примерах и оставить их них только те функции, которые будут использоваться в вашем конкретном алгоритме.

Итак, вот примерный список задач для данного задания. Конечно, можно придумывать и добавлять аналогичные по использованию классов и по трудоемкости задачики. Некоторые задачи могу оказаться достаточно сложными. В таком случае нет нужды добиваться полного решения. Нужно чтобы какие-то содержательные алгоритмы выражались через методы и операции с элементаными геометрическими объектами.

**Задание 1.1.** Определить угол, под которым данный многоугольник виден из заданной точки.

**Задание 1.2.** Дан выпуклый многоугольник. Найти его минимальный и максимальный диаметры (т.е. размер "щели", в которую он еще может протиснуться (min) и которую он еще будет касаться при протискивании (max)).

**Задание 1.3.** Дан выпуклый многоугольник. Найти прямоугольник минимальной площади, который охватывает данный многоугольник (стороны многоугольника не обязательно параллельны осям).

**Задание 1.4.** Определить расстояние между двумя многоугольниками общего вида.

**Задание 1.5.** Определить расстояние между двумя выпуклыми многоугольниками с линейной трудоемкостью по числу сторон.

**Задание 1.6.** Построить пересечение двух выпуклых многоугольников.

**Задание 1.7.** Дано множество точек на плоскости. Построить минимальное покрывающее дерево для этого множества.

**Задание 1.8.** Дан выпуклыйм ногоугольник. Построить многоугольник, который получится, если линию, задающую каждую сторону, отодвинуть в перпендикулярном ей направлении на величину $h$ (можно двигать как "внутрь" так и "наружу").

**Задание 1.9.** Дан многоугольник общего вида. Построить многоугольники, полученные сечением данного многоугольника заданной прямой.

**Задание 1.10.** Дан многоугольник. Разбить его на треугольники (не вводя дополнительных точек, если это возможно).

**Задание 1.11.** Прямая определеляет полуплоскость отрицательным направлением своей нормали. Построить многоугольник пересечения нескольких полуплоскостей (для заданного множества прямых).

**Задание 1.12.** Найти наилучшее совмещение двух треугольников (т.е. подвинуть один из них так, чтобы площадь их пересечения стала максимальной).

**Задание 1.13.** Дано множество прямоугольников со сторонами параллельными осям. Построить их объединение (т.е. в общем случае несколько многоугольников со сторонами параллельными осям).

**Задание 1.14.** Дано множество отрезков на плоскости. Образуют ли некоторые из них многоугольник? Если да, то построить эти многоугольники.

**Задание 1.15.** Дано множество точек на плоскости. Построить многоугольник их выпуклой оболочки

**Задание 1.16.** Дано множество многоугольников. Пересечь их с данным прямоугольником (стороны прямоугоььника параллельны осям), т.е. обрезать по границам прямоугольника.