федеральное ГОСУДАРСТВЕННОЕ Бюджетное ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

**"САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ "ЛЭТИ" им.В.И.Ульянова (Ленина)" (СПбГЭТУ)**

Кафедра МОЭВМ

**Отчет по лабораторной работе №2**

«Основные проектные решения»

Выполнил: Тюлин Р.Д.

Факультет КТИ

Группа № 0304

Преподаватель: Смольянинов А.В.

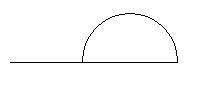
Санкт-Петербург

2013

## Формулирование основной идеи построения программы

Основным объектом задания является схематичный набор изображений «бейсболка мелкая», где каждое изображение образовано из следующих частей:

1. «Козырек» – представляется “линией”, задаётся с помощью двух точек;
2. «Шапка» – представляется ”сегментом круга ”, задаётся с помощью двух точек и угла разворота (центр и точка, относительно которой разворачивается дуга сегмента).

Набор изображений «Бейсболок» должен быть представлен в виде списка, разработанного в работе №1.  
 

## Построение системы объектов

Для построения объекта «Шапка» определим класс «Сегмент круга», а для объекта «Козырек» – класс «Линия».

### Объект «Точка»

|  |  |
| --- | --- |
| Декартовы координаты точки на плоскости. | float x, y; |
| Создаёт новую точку с координатами . | Point(); |
| Создаёт новую точку с координатами в прямоугольной системе координат. | Point(float x, float y); |
| Создаёт копию точки p. | Point(const Point &p); |
| Удаляет экземпляр точки. | ~Point(); |
| Создает точку, сдвинутую относительно данной на указанные сдвиги. | Point MoveBy(Point dxdy); |

### Объект «Линия»

|  |  |
| --- | --- |
| Начало и конец линии. | Point start, end; |
| Создает новую линию из начала и конца. | Line(Point start, Point end); |
| Создает копию линии. | Line(const Line &obj); |
| Удаляет экземпляр линии. | ~Line(); |
| Возвращает начало линии. | Point GetStart(); |
| Возвращает конец линии. | Point GetEnd(); |
| Двигает линию на указанный сдвиг. | void MoveBy(Point dxdy); |

### Объект «Сегмент круга»

|  |  |
| --- | --- |
| Начало сегмента. | Point start; |
| Ширина и высота сегмента. | double width, height; |
| Создает новый сегмент круга из указанных центра, ширины и высоты. | Segment(Point start, double width, double height); |
| Создает копию сегмента круга. | Segment(const Segment &obj); |
| Удаляет сегмент круга. | ~Segment(); |
| Двигает сегмент круга на указанный сдвиг. | void MoveBy(Point dxdy); |

### Объект «Шапка»

|  |  |
| --- | --- |
| Сегмент, описывающий шапку. | Segment seg; |
| Создает новую шапку из указанных положения., ширины и высоты. | Hat(Point position, float len, float height); |
| Создаёт копию шапки. | Hat(const Hat &obj); |
| Удаляет экземпляр шапки. | ~Hat(); |
| Получает набор характерных точек. | CycleList<Point> GetPoints(); |
| Двигает шапку на указанный сдвиг. | void MoveBy(Point dxdy); |

### Объект «Козырек»

|  |  |
| --- | --- |
| Линия, представляющая козырек. | Line line; |
| Создает новую шапку в указанной позиции с указанными высотой и шириной. | Visor(Point position, double dx, double dy); |
| Создаёт копию шапки. | Visor(const Visor &obj); |
| Удаляет экземпляр шапки. | ~Visor(); |
| Получает набор характерных точек. | CycleList<Point> GetPoints(); |
| Двигает козырек на указанный сдвиг. | void MoveBy(Point dxdy); |

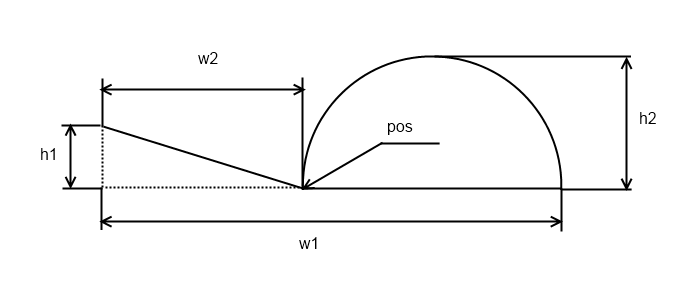
### Объект «Кепка»

|  |  |
| --- | --- |
| Шапка для описания кепки. | Hat hat; |
| Козырек для описания кепки. | Visor visor; |
| Создает новую кепку с указанными параметрами. | Cap(Point pos, double w1, double w2, double h1, double h2); |
| Создаёт копию кепки. | Cap(const Cap &obj); |
| Удаляет экземпляр кепки. | ~Cap(); |
| Получает набор характерных точек. | CycleList<Point> GetPoints(); |
| Двигает козырек на указанный сдвиг. | void MoveBy(Point dxdy); |

### Объект «Кольцевой список», «Линейный однонаправленный список» и элементы для них

Были спроектированы и реализованы в лабораторной работе №1.

## Математические модели объектов

  
Для построения используется декартовая система координат.

Pos – координаты всей кепки, шапки и козырька.

W1 – ширина всей кепки  
w2 – ширина козврька  
h1 – высота козырька  
h2 – высота шапки

## Проектирование классов объектов на языке C++ на основе контрактов

В графах “Предусловия” и “Постусловия” и “Инвариант” предполагается, что значения типа данных float не содержат значений NaN, ±∞ и соответствуют множеству , а значения Point соответствуют множеству .

### Объект «Шапка»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Имя метода | Предусловия | Постусловия |
| 1 | Hat(Point position, float len, float height); | len > 0 height > 0 | seg.start = position  seg.width = len  seg.height = height |
| 2 | Hat(const Hat &obj); | - | \*this = obj |
| 3 | ~Hat(); | - | - |
| 4 | CycleList<Point> GetPoints(); | - | - |
| 5 | void MoveBy(Point dxdy); | - | seg.start = seg.start + dxdy |

### Объект «Козырек»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Имя метода | Предусловия | Постусловия |
| 1 | Visor(Point position, double dx, double dy); | - | line.start = position  line.end = Point(position.x + dx, position.y + dy) |
| 2 | Visor(const Visor &obj); | - | \*this = obj |
| 3 | ~Visor(); | - | - |
| 4 | CycleList<Point> GetPoints(); | - | - |
| 5 | void MoveBy(Point dxdy); | - | line.start = line.start + dxdy  line.end = line.end + dxdy |

### Объект «Кепка»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Имя метода | Предусловия | Постусловия |
| 1 | Cap(Point pos, double w1, double w2, double h1, double h2); | w1 > w2  w1 > 0  w2 > 0  h2 > 0 | hat.seg.start = pos  hat.seg.width = w1 – w2  hat.seg.height = h2  visor.line.start = pos  visor.line.end = Point(pos.x – w2, h1) |
| 2 | Cap(const Cap &obj); | - | \*this = obj |
| 3 | ~Cap(); | - | - |
| 4 | CycleList<Point> GetPoints(); | - | - |
| 5 | void MoveBy(Point dxdy); | - | hat.start = hat.start + dxdy  visor.start = visor.start + dxdy |

**Инвариант**:

### Объект «Кольцевой список» и «Элемент списка»

Описания приведены в отчёте к лабораторной работе №1.