

### Интерфейс стека

Item& Top() const – возвращает элемент, находящийся на вершине стека.  
 void Push(const Item&) – «заталкивает» элемент в стек.  
 Item& Pop() – «вытаскивает» элемент с вершины стека.

## C.2. Iterator

Iterator – это абстрактный класс, который определяет интерфейс обхода агрегата:

```
template <class Item>
class Iterator {
public:
    virtual void First() = 0;
    virtual void Next() = 0;
    virtual bool IsDone() const = 0;
    virtual Item CurrentItem() const = 0;
protected:
    Iterator();
};
```

Операции делают следующее:

virtual void First() – позиционирует итератор на первый объект в агрегате.

virtual void Next() – позиционирует итератор на следующий по порядку объект.

virtual bool IsDone() const – возвращает true, если больше не осталось объектов.

virtual Item CurrentItem() const – возвращает объект, находящийся в текущей позиции.

## C.3. ListIterator

ListIterator реализует интерфейс класса Iterator для обхода списка List. Его конструктор принимает в качестве аргумента список, который нужно обойти:

```
template <class Item>
class ListIterator : public Iterator<Item> {
public:
    ListIterator(const List<Item>* aList);

    virtual void First();
    virtual void Next();
    virtual bool IsDone() const;
    virtual Item CurrentItem() const;
};
```

## C.4. Point

Класс Point представляет точку на плоскости с помощью декартовых координат, поддерживает минимальный набор арифметических операций над векторами. Координаты точки определяются так:

```
typedef float Coord;
```

Операции класса Point не нуждаются в пояснениях:

```
class Point {
public:
    static const Point Zero;

    Point(Coord x = 0.0, Coord y = 0.0);

    Coord X() const; void X(Coord x);
    Coord Y() const; void Y(Coord y);

    friend Point operator+(const Point&, const Point&);
    friend Point operator-(const Point&, const Point&);
    friend Point operator*(const Point&, const Point&);
    friend Point operator/(const Point&, const Point&);

    Point& operator+=(const Point&);
    Point& operator-=(const Point&);
    Point& operator*=(const Point&);
    Point& operator/=(const Point&);

    Point operator-();

    friend bool operator==(const Point&, const Point&);
    friend bool operator!=(const Point&, const Point&);

    friend ostream& operator<<(ostream&, const Point&);
    friend istream& operator>>(istream&, Point&);
};
```

Статический член Zero представляет начало координат Point (0, 0).

## C.5. Rect

Класс Rect представляет прямоугольник, стороны которого параллельны осям координат. Прямоугольник определяется начальной вершиной и размерами то есть шириной и высотой. Операции класса Rect не нуждаются в пояснениях:

```
class Rect {
public:
    static const Rect Zero;
```