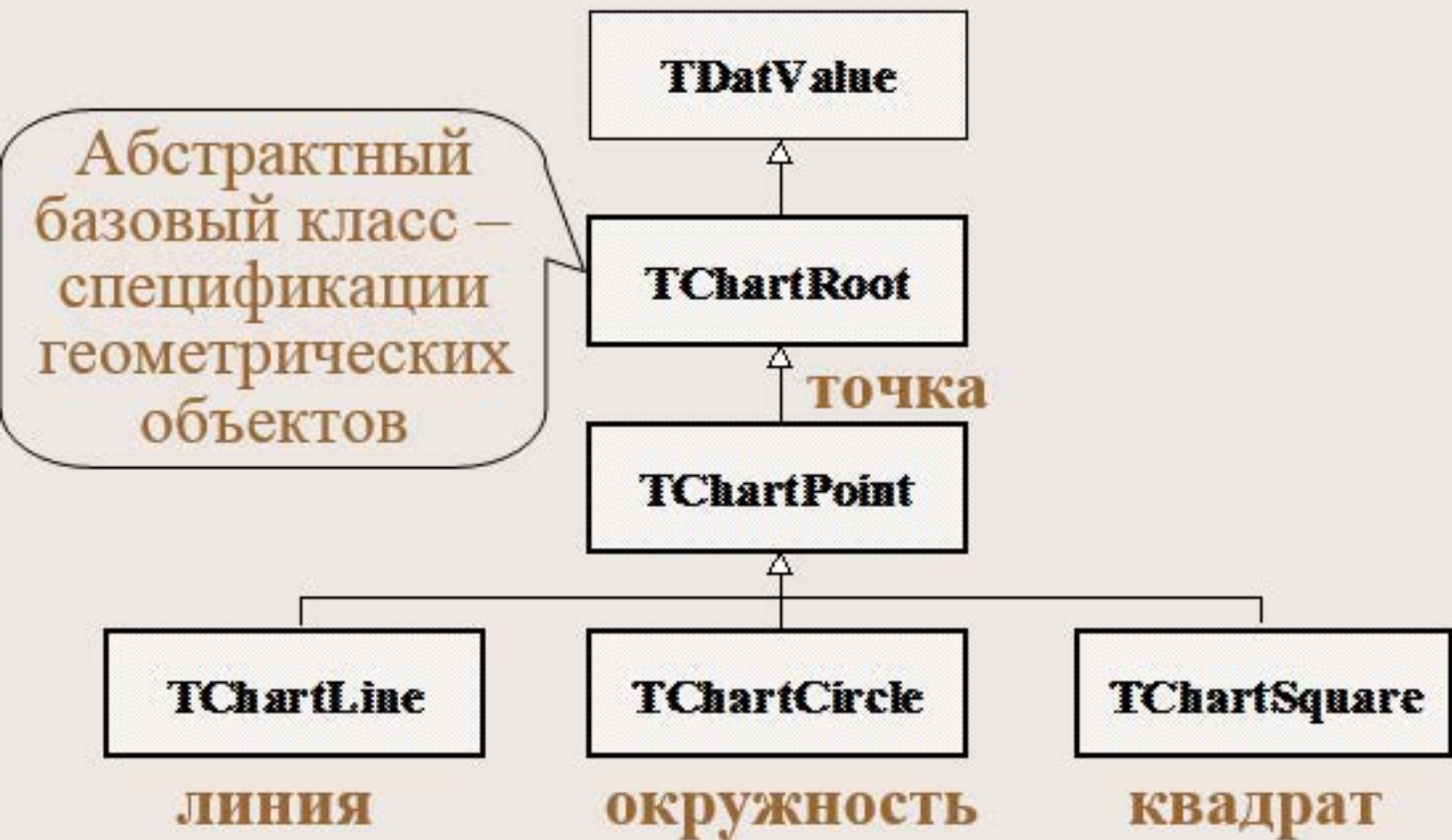


Рассмотрим в качестве базовых объектов **основные геометрические фигуры** – точку, окружность, прямоугольник и т.д.


- **Информационное описание объектов** – параметры фигуры (координаты, размер, радиус и др.). В общем случае, описание фигуры включает значение координат некоторой опорной точки

- **Операции обработки** геометрических объектов включают методы для задания и изменения параметров; расширим набор операций процедурами визуализации (например, на экране дисплея) и скрывтия фигур.

Возможная **схема иерархии** классов для реализации геометрических объектов может состоять в следующем




(!!!) Для обеспечения возможности динамической визуализации геометрических объектов введем тип данных, значения которого вычисляются в соответствии с задаваемым формульным выражением

```
template <class TValue>   
class TFormValue : public TDatValue {  
protected:  
    char    Formula[FormLen]; // формула  
    TValue Value; // значение  
    double Param; // параметр формулы  
private:  
    FTRAN ft; // формульный транслятор  
public:  
    TFormValue(TValue val=0, const char *f="");  
    TValue GetValue(double par); // вычислить  
    TValue GetValue(); // получить значение  
};
```


Абстрактный базовый класс TChartRoot

```
class TChartRoot : public TDatValue {
protected:  // поля
    int Visible;           // ВИДИМОСТЬ
    TFormValue<int> Active; // АКТИВНОСТЬ
public:
    TChartRoot();
    int IsVisible( void ) const; // высвечен ?
    int IsActive ( void ) const; // активен ?
    void SetActiveValue(int val=1, char *f=NULL);
    virtual void Show()=0; // визуализация
    virtual void Hide()=0; // сккрытие объекта
    // перевычисление параметров
    virtual void CalcParams(double t=-1);
    // перевысветка объекта
    virtual void ViewTimeShot(double t=-1);
};
```


Класс TChartPoint для описания точки на плоскости


```
class TChartPoint : public TChartRoot {   
protected: // поля  
    TFormValue<int> X, Y; // координаты точки  
public:  
    TChartPoint (int a=0, int b=0);  
    int  GetValueX(void);  
    int  GetValueY(void);  
    void SetValueX(int val=0, char *f=NULL);  
    void SetValueY(int val=0, char *f=NULL);  
    virtual void Show(); // визуализация точки  
    virtual void Hide(); // скрывание точки  
    // перевычисление параметров  
    virtual void CalcParams(double t=-1);  
};
```

Класс TChartCircle для описания окружности ➡

```
class TChartCircle : public TChartPoint {
protected:  // поля
    TFormValue<int> Radius; // радиус окружности
public:
    TChartCircle (int a=0, int b=0, int rad=1);
    void SetRadiusValue(int val=1, char *f=NULL);
    virtual void Show(); // визуализация
    virtual void Hide(); // скрывание окружности
    // перевычисление параметров
    virtual void CalcParams(double t=-1);
};
```

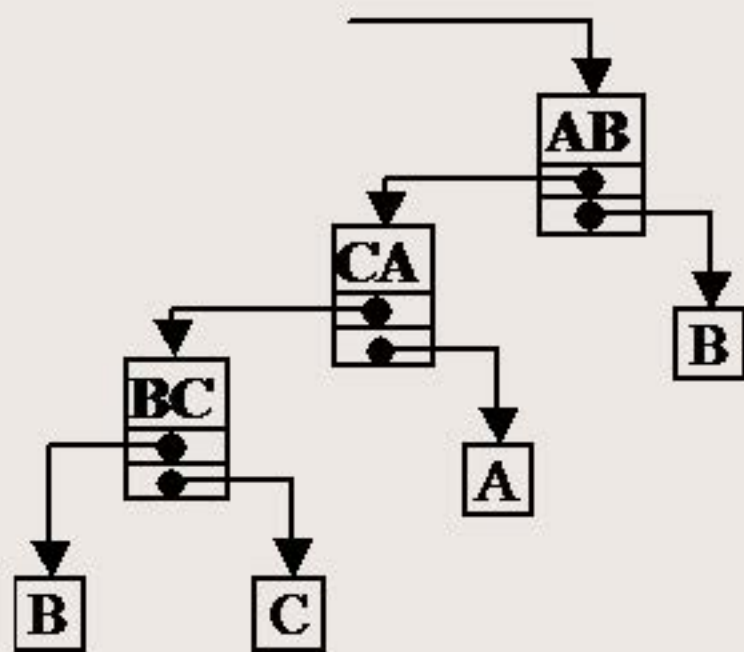

Составной объект – набор геометрических объектов (как базовых, так и составных), рассматриваемых при выполнении операций обработки как единый объект

```
class TChartGroup : public TChartRoot { 
protected:
    TDatList Group; // список объектов
public:
    TChartGroup () {}
    void InsUnit ( TChartRoot *pUnit ); // добавить
    virtual void Show(); // визуализация
    virtual void Hide(); // скрывтие
    // перевычислить параметры
    virtual void CalcParams(double t=-1);
}; программа, приложение
```

Геометрический объект может быть сконструирован с использованием уже существующих объектов (например, ломаная может быть определена через набор конечных точек составляющих отрезков) 

```
class TChartPolyline : public TChartGroup {
public:
    TChartPolyline () {}
    void InsPoint (TChartRoot *pUnit ) : // добавить
    virtual void Show(); // визуализация
    virtual void Hide(); // сккрытие
    // перевычислить параметры
    virtual void CalcParams(double t=-1);
};
```


4. Комбинирование объектов – структура хранения..




- ☑ Узел структуры хранения представляет линию чертежа
- ☑ Указатель на начальную точку линии может также указывать на линию, т.е. конечная точка предыдущей линии является начальной точкой следующей линии

Определение 2.2. Структура хранения данного вида называется *плексом* (содержит элементы разного типа)

☞ Плекс является общей структурой хранения сетевых моделей данных

4. Комбинирование объектов – структура хранения

```
class TChart : public TChartGroup { 
protected:
    stack<TChartLine> St; // стек для обхода
public:
    TChart () {}
    TChartRoot *GetFirstPoint(void); // нач.т.
    TChartRoot *GetLastPoint(void): // кон.т.
    void SetFirstPoint ( TChartRoot *pUnit );
    void SetLastPoint ( TChartRoot *pUnit );
    virtual void Show(); // визуализация рисунка
    virtual void Hide(); // сккрытие рисунка
};
```