1. Класс-группа – это объект, в который включены другие объекты (элементы). Элементы также могут быть группами. Примеры:

Военная структура (Отделение, взвод, полк);

Вещество, состоящее из молекул, а те в свою очередь – из атомов

class List  
{  
  Node \*head;    // Корень списка  
  int count = 0; // Количество узлов списка  
  Node\* Prev(Node \*); // Переход к предыдущему узлу  
public:  
  List()

{

head = NULL;  // Инициализация списка

}    
  int getCount() { return count; } // Получение количества узлов списка  
  bool isEmpty() { return head == NULL; }  // Проверка, пуст ли список  
  int getValue(Node\* p) { return p->field; } // Получение значения узла списка  
  void setValue(Node \*p, int val) { p->field = val; } // Установка значения узла списка  
  Node\* getFirst() { return head; } // Получение корневого узла списка  
  Node\* getLast();      // Получение последнего узла списка  
  void Clear();        // Очистка списка  
  Node\* Next(Node \*);      // Переход к следующему узлу  
  Node\* Add(int, Node\*);    // Добавление узла списка  
  Node\* Delete(Node\*);    // Удаление узла списка  
  void Print();        // Вывод значений узлов списка  
  void Swap(Node\*, Node\*);  // Взаимообмен двух узлов  
};

1. public:

List()

{

}

public:

List(integer knot1, integer knot2)

{

knot1 = 5;

knot2 = 6;

}

List(const List &list) :

head(list.head);

1. f
2. d
3. Группа дает иерархию объектов (целое/часть). Она построена на основе агрегации. Первый вид – иерархия классов, построенная на основе исследования
4. В
5. Событие можно представить как пакет информации, которым обмениваются объекты. Нажатие на кнопку клавиатуры или мыши создает событие, которое передается по цепочке объектов
6. Для него задаются два параметра: command – вид команды, которую необходимо выполнить при появлении данного события;

передаваемая с событием информация

1. Class TEvent

{

int what;

union

{

Int command;

Class

{

Int message;

Int a;

};

};

};

1. What может принимать значения:

evNothing – пустое событие, означающее, что ничего делать не нужно. Это значение присваивается, когда событие обработано каким-либо объектом.

evMouseEventType – событие происходит в результате взаимодействия с мышью.

evKeyDownEvent – событие наступает при нахождении компонента в фокусе, при нажатии пользователем любой клавиши.

evMessageEvent – событие исходящего вызова от объекта к объекту.

1. Command может принимать значения:

evNothing – пустое событие, означающее, что ничего делать не нужно. Это значение присваивается, когда событие обработано каким-либо объектом.

evMessage – событие-сообщение от объекта. Для него задаются два параметра:

command – код команды, необходимую для выполнения при появлении данного события

передаваемая с событием информация (сообщение)

1. Int message – это само сообщение (его текст)

Int a – это параметр команды

1. Пустое событие, событие-сообщение от объекта
2. evMessage