Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет» (ПНИПУ)

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

**Лабораторная работа по классам № 8**

**по дисциплине**

**«Основы алгоритмизации и программирования»**

**семестр** 2

Выполнил студент гр. ИВТ-21-1б

Балдин Алексей Игоревич

Проверил:

Яруллин Денис Владимирович

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_

(оценка) (подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

г. Пермь – 2022

**Цель**

Разработка программы, управляемой событиями.

**Постановка задачи**

* + 1. Определить иерархию пользовательских классов (см. лабораторную работу №5). Во главе иерархии должен стоять абстрактный класс с чисто виртуальными методами для ввода и вывода информации об атрибутах объектов.
    2. Реализовать конструкторы, деструктор, операцию присваивания, селекторы и модификаторы.
    3. Определить класс-группу на основе структуры, указанной в варианте.
    4. Для группы реализовать конструкторы, деструктор, методы для добавления и удаления элементов в группу, метод для просмотра группы, перегрузить операцию для получения информации о размере группы.
    5. Определить класс Диалог – наследника группы, в котором реализовать методы для обработки событий.
    6. Добавить методы для обработки событий группой и объектами пользовательских классов.
    7. Написать тестирующую программу.
    8. Нарисовать диаграмму классов и диаграмму объектов.

**Исходные данные**

Вариант 5:

Базовый класс: ПЕЧАТНОЕ\_ИЗДАНИЕ(PRINT)

Название– string Автор – string Производный класс

ЖУРНАЛ (MAGAZIN)

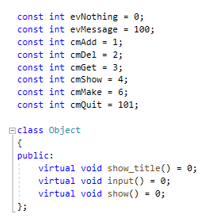
Количество страниц - int Группа – Вектор (Vector). Команды:

* Создать группу (формат команды: m количество элементов группы).
* Добавить элемент в группу (формат команды: +)
* Удалить элемент из группы (формат команды -)
* Вывести информацию об элементах группы (формат команды: s)

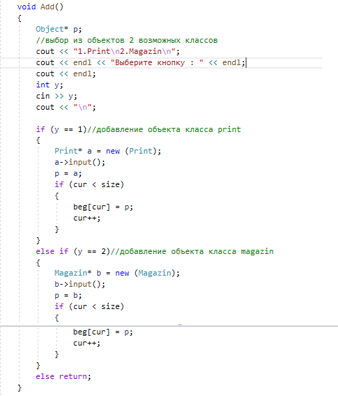
Вывести информацию о названиях всех элементов группы (формат команды: z)

Конец работы (формат команды: q)

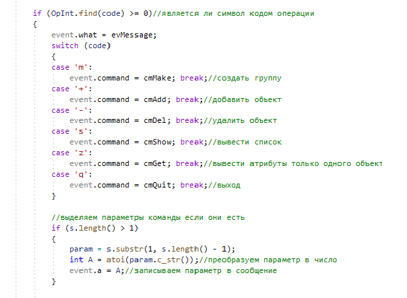
Здесь прописаны базовые команды – в виде констант. Далее создаю класс, в котором 3 – чисто виртуальных функции, значит этот класс является абстрактным.



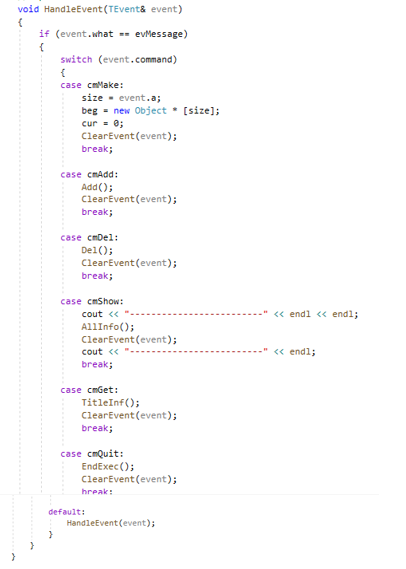
В классе Vector – есть метод – Add, он как раз и даёт выбрать кнопку (1 или 2):



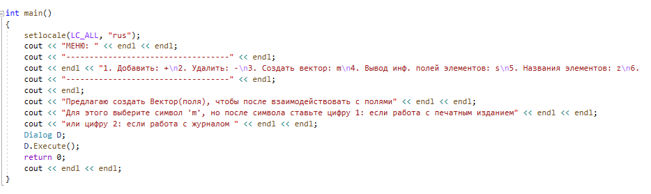
В классе Dialog – через проверку на нужный символ, выполнятся операции сравнения введённого символа с теми, которое даны по условию.



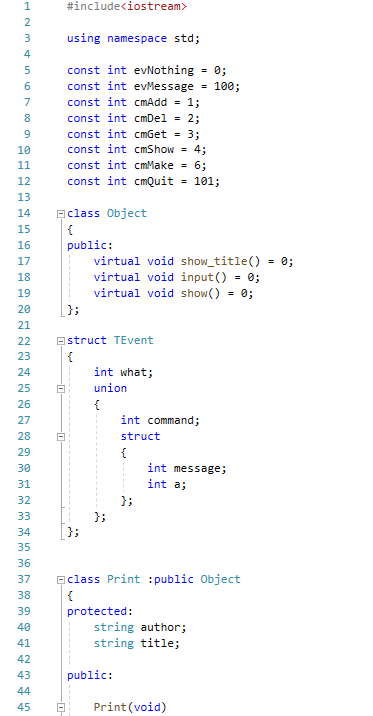
Обработчик событий – выводит соответствующее сообщение под каждый определенный символ:

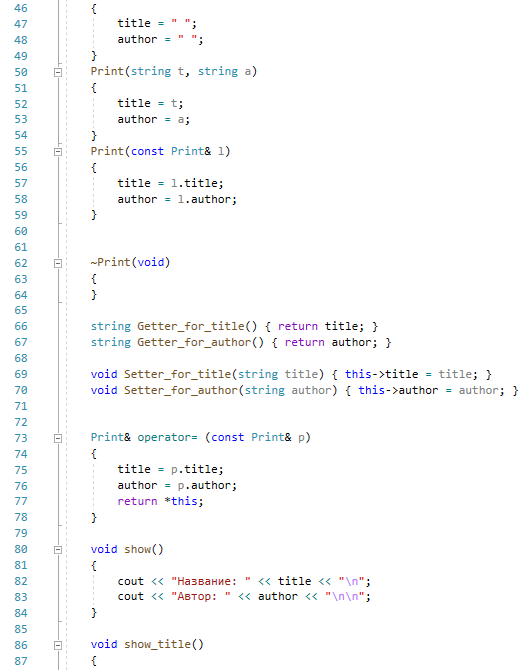


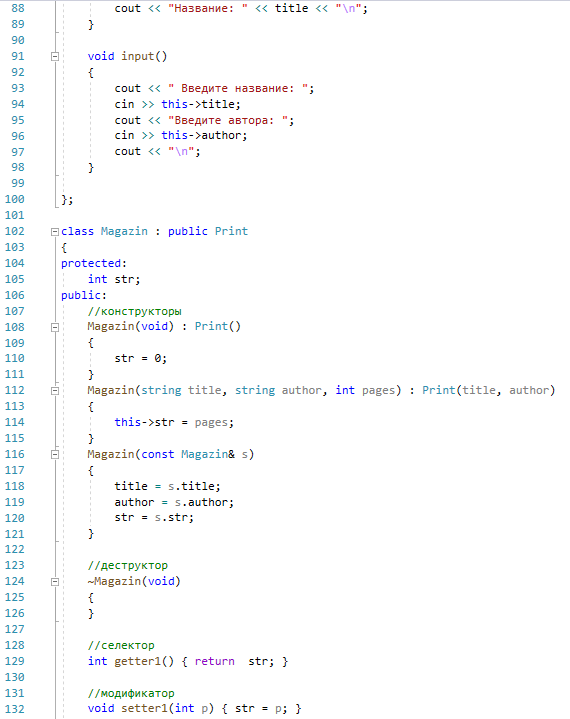
В main () – сделано на подобии (меню), расписаны команды под номерами, и небольшая инструкция, что сделать, чтобы создать поля Вектора:

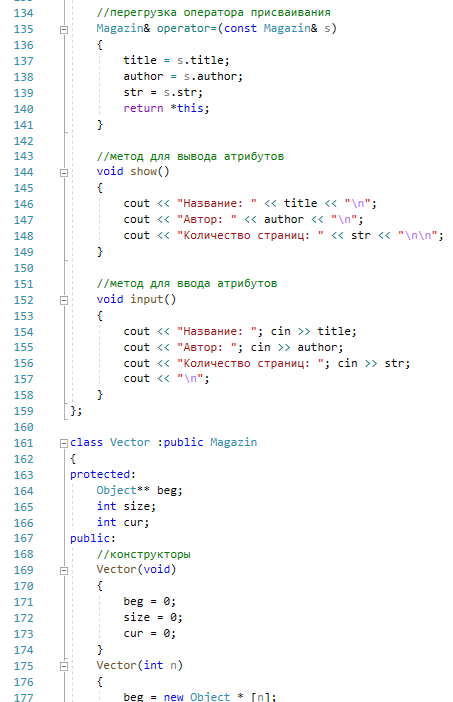


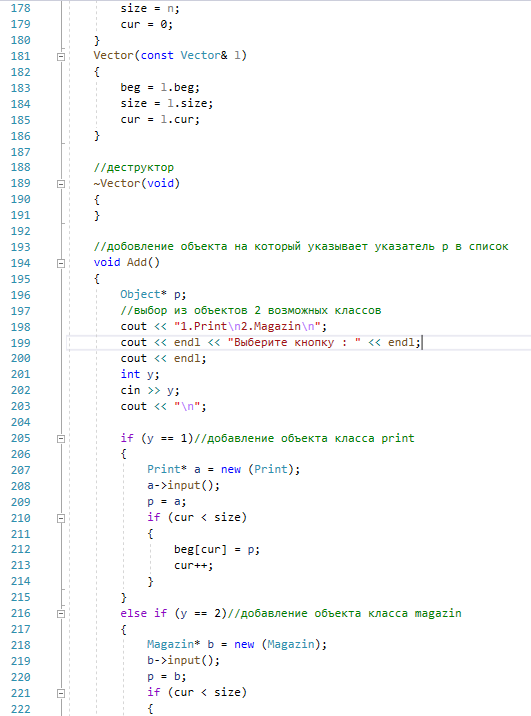
Код программы:

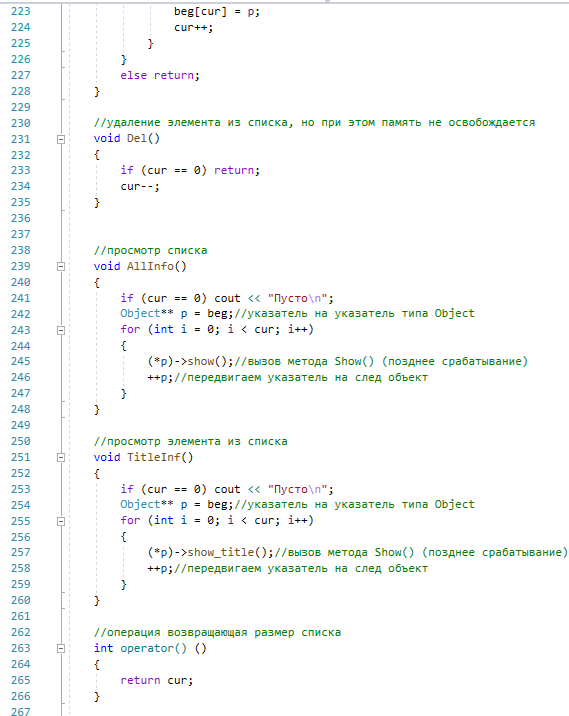


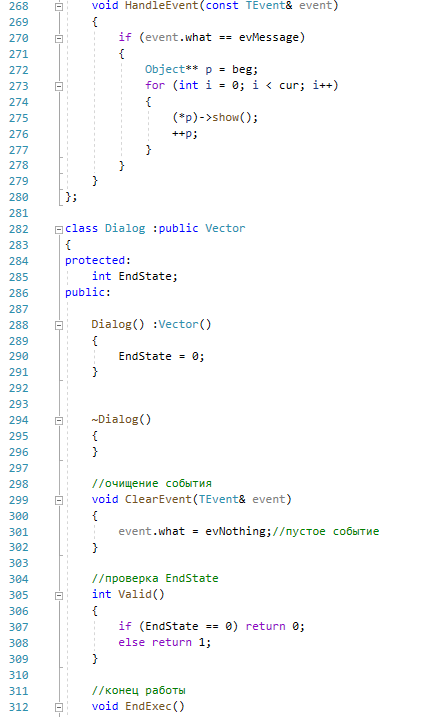


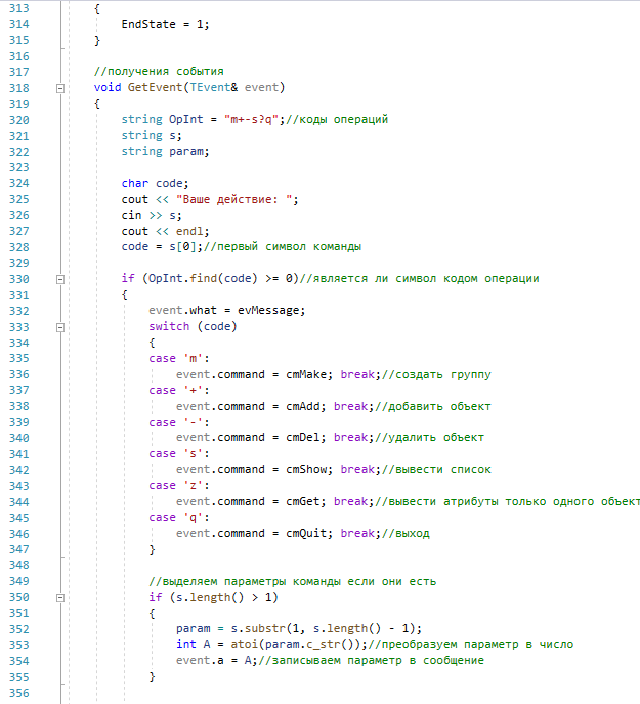


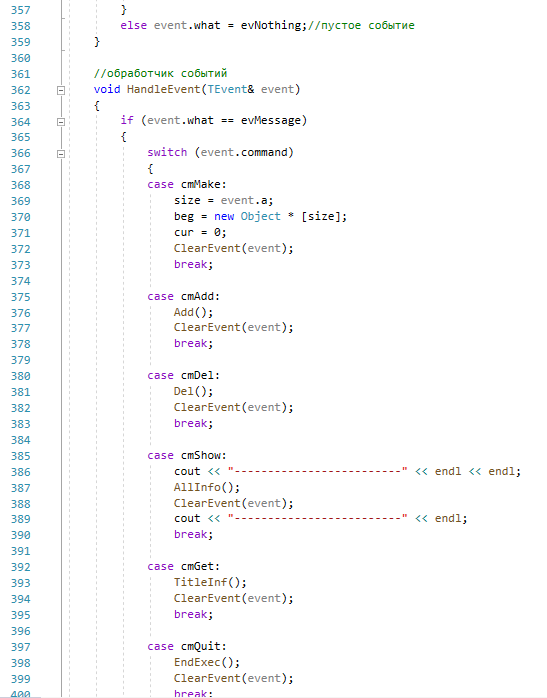


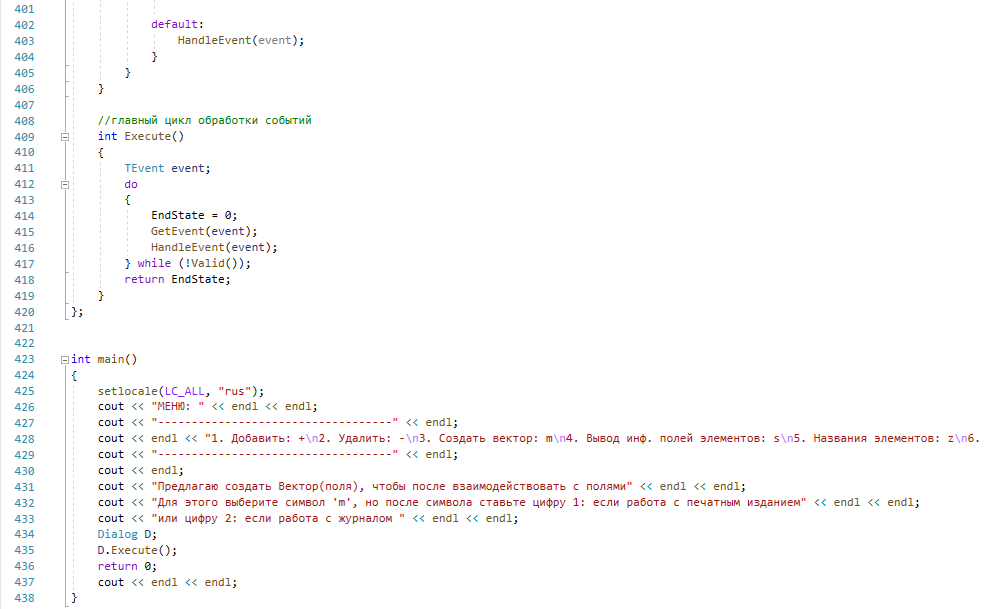




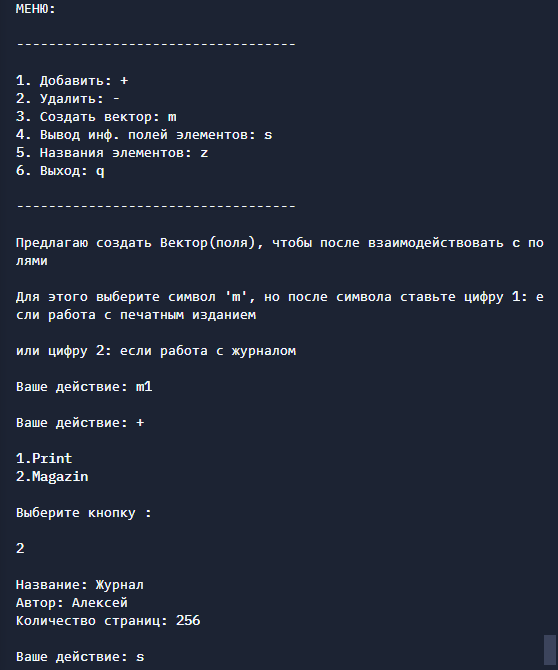








Результат программы:



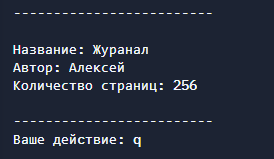


Диаграмма к задаче:

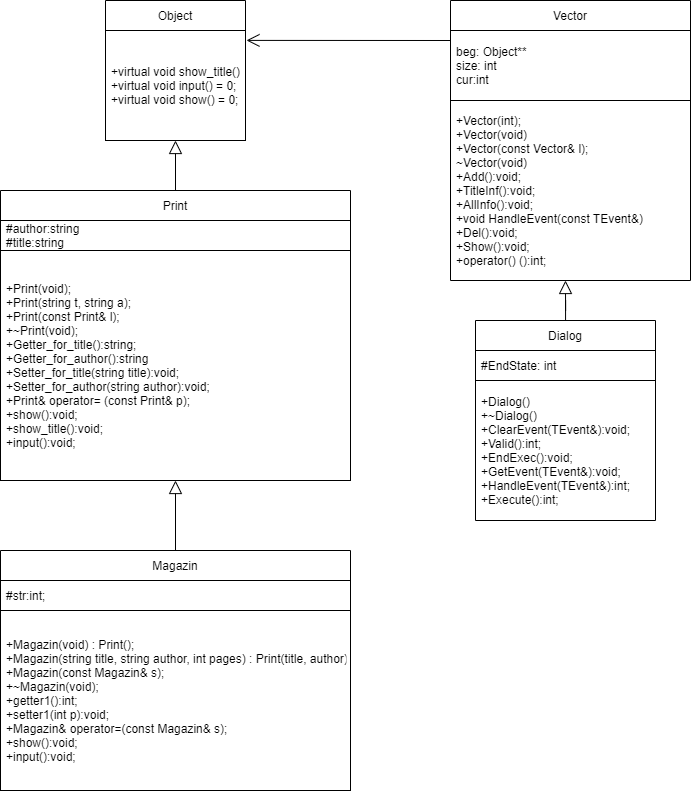


Рисунок 1

**Контрольные вопросы:**

* + 1. Что такое класс-группа? Привести примеры таких классов.
    2. Привести пример описания класса-группы Список (List).
    3. Привести пример конструктора (с параметром, без параметров, копирования) для класса-группы Список.
    4. Привести пример деструктора для класса-группы Список.
    5. Привести пример метода для просмотра элементов для класса-группы Список.
    6. Какой вид иерархии дает группа?
    7. Почему во главе иерархии классов, содержащихся в группе объектов должен находиться абстрактный класс?
    8. Что такое событие? Для чего используются события?
    9. Какие характеристики должно иметь событие-сообщение?
    10. Привести пример структуры, описывающей событие.
    11. Какие методы необходимы для организации обработки сообщений?
    12. Какой вид имеет главный цикл обработки событий-сообщений?
    13. Какую функцию выполняет метод ClearEvent()? Каким образом?
    14. Какую функцию выполняет метод HandleEvent ()?Каким образом?
    15. Какую функцию выполняет метод GetEvent ()?
    16. Для чего используется поле EndState? Какой класс (объект) содержит это поле?
    17. Для чего используется функция Valid()?

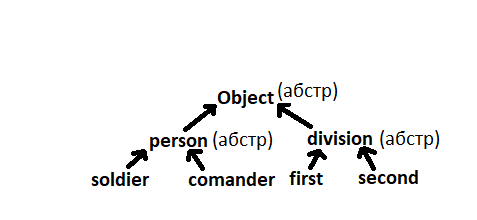
**Ответы на вопросы**

**1)** Понятия "группа" от "контейнер" отличаются. Контейнер используется для хранения других данных. Пример контейнеров: объекты контейнерных классов библиотеки STL в C++ (массивы, списки, очереди).

В отличие от контейнера группа есть класс, который не только хранит объекты других классов, но и обладает собственными свойствами, не вытекающими из свойств его элементов.

Группа дает второй вид иерархии (первый вид - иерархия классов, построенная на основе наследования) - иерархию объектов (иерархию типа целое/часть), построенную на основе агрегации.

Вот иерархия классов: Division, first, second - "классы-группы"



2)

|  |
| --- |
| #include <list> |

Далее используем конструкцию ниже:

|  |  |
| --- | --- |
| 1 | list < тип данных > <имя контейнера>; |

* < тип данных > — сюда мы должны указать тип, который хотим использовать.
* <имя контейнера> — это будет нашим именем контейнера. Лучше указывать такое имя, которое будет *говорить*, за что этот контейнер отвечает.

Вот пример создания списка с типом string:

|  |  |
| --- | --- |
| 1 | list <string> listok; |

5) Осуществляется через итератор

6)Группа дает второй вид иерархии (первый вид - иерархия классов, построенная на основе наследования) - иерархию объектов (иерархию типа целое/часть), построенную на основе агрегации.

7)  Во главе иерархии стоит абстрактный базовый класс. Он используется для наследования интерфейса. Производные классы будут конкретизировать и реализовать этот интерфейс

8)Событие в объектно-ориентированном программировании — это сообщение, которое возникает в различных точках исполняемого кода при выполнении определённых условий. События предназначены для того, чтобы иметь возможность предусмотреть реакцию программного обеспечения.

События применяются довольно широко. Примером могут служить всевозможные библиотеки, реализующие графический интерфес пользователя. Но события при правильном применении могут оказаться ДЕЙСТВИТЕЛЬНО ПОЛЕЗНОЙ ВЕЩЬЮ.

9) Объект инициирует событие и вызываются все объекты-обработчики. Т.е. от одного объекта к нескольким. Причем объект инициатор события может ничего не «знать» об его обработчиках, поэтому событие называют исходящим вызовом. Раз уж в С++ события на уровне языка не поддерживаются, значит стоит организовать их на уровне библиотеки.

10) struct TEvent

{

int what;

union{ MouseEventType mouse; KeyDownEvent keyDown;}

};

11) Функция обработки, сообщения опознает, какое именно сообщение поступило для обработки и выполняет соответствующие действия. Сообщения распознаются по его коду. Обычно функция окна содержит оператор switch, который служит для определения кода сообщений. Вот пример типичной функции окна ... Чтобы класс смог обрабатывать сообщения, необходимо, чтобы он имел таблицу сообщений класса. В этой таблице для каждого сообщения указан метод класса, предназначенный для его обработки.

12) Execute реализует главный цикл обработки событий. Он постоянно получает событие путем вызова GeEvent и обрабатывает их с помощью HandleEvent. Этот цикл завершается, когда поступит событие «конец». HandleEvent – обработчик событий. Обрабатывает каждое событие нужным для него образом. (Показано в решении лаб.р.)

13)Метод ClearEvent − очистка события. ClearEvent очищает событие, присваивая полю Event.What значение evNothing. + + Главный цикл обработки событий (метод Execute).

14)HandleEvent – обработчик событий. Обрабатывает каждое событие нужным для него образом. Если объект должен обрабатывать определенное событие (сообщение), то его метод HandleEvent должен распознавать это событие и реагировать на него должным образом. Событие может распознаваться, например, по коду команды (поле command).

15)Метод GetEvent наследуется всеми видимыми элементами от TView и является основным источником событий. Этот метод вначале проверяет, не подготовил ли событие метод PutEvent и, если это так, возвращает его.

16) Показано в решении лаб.р.