|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **по домашнему заданию №** | 1 |

**Название:** Программирование на Object Pascal с использованием классов

**Дисциплина:** Объектно-ориентированное программирование

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-22Б |  |  | Д.С. Твердюк |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | А.А. Веселовский |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Часть 1. Графический редактор**

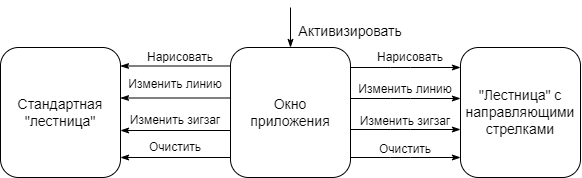
Разработать иерархию классов. Поместить определение классов в отдельном модуле.

Класс, позволяющий рисовать ломаную линию, состоящую из прямых углов (лестницу). При нажатии левой клавиши мыши между текущей и предыдущей точкой рисуется прямой угол. Нажатие правой клавиши заканчивает рисование линии.

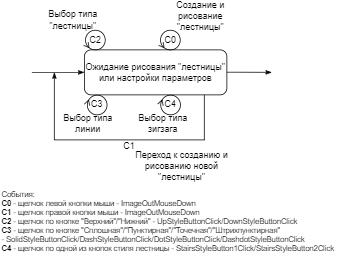
Класс, позволяющий рисовать лестницу по тем же правилам, но каждая «ступень» оканчивается стрелкой.

Тип линии и тип зигзага (верхний или нижний) выбирать с использованием интерфейсных элементов.

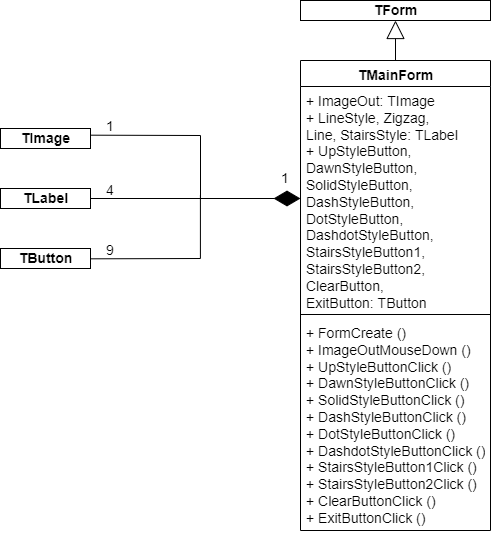
Объектная декомпозиция:

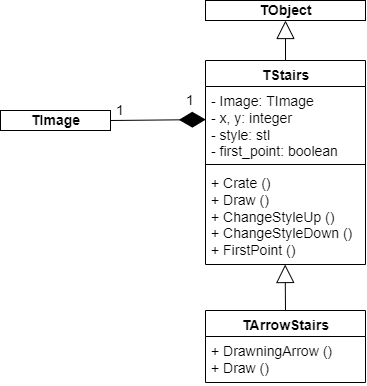


Граф состояний интерфейса:

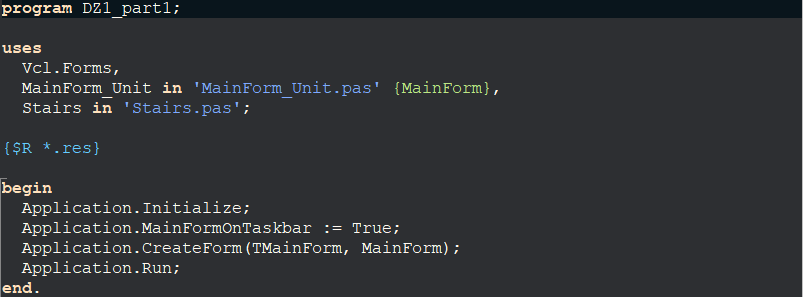


UML-диаграмма классов:

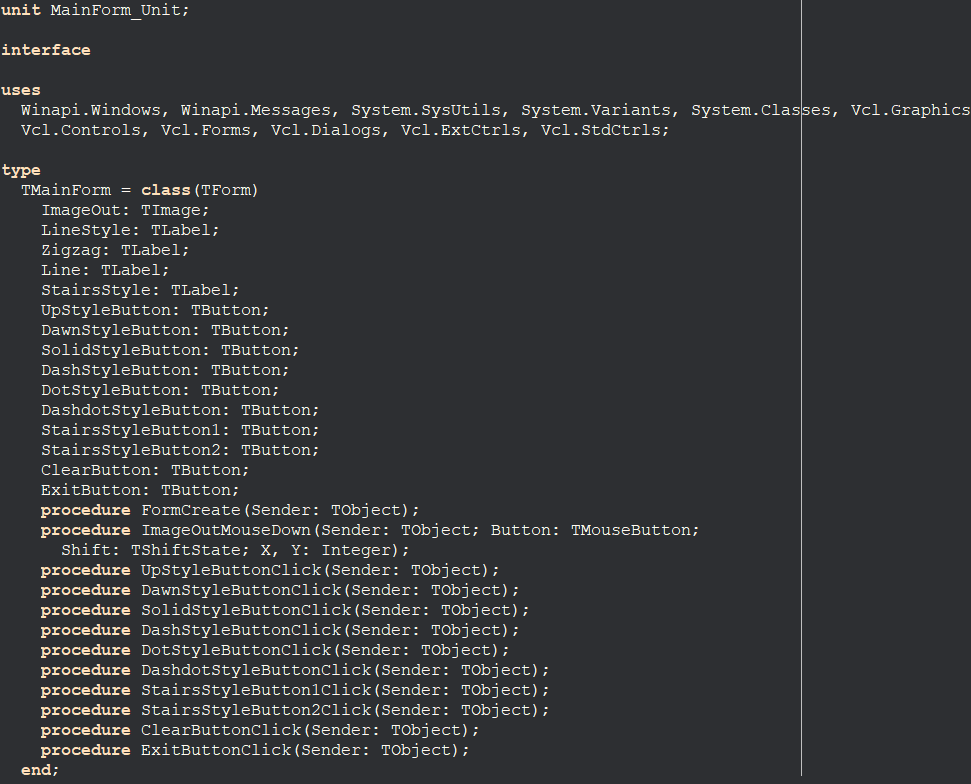


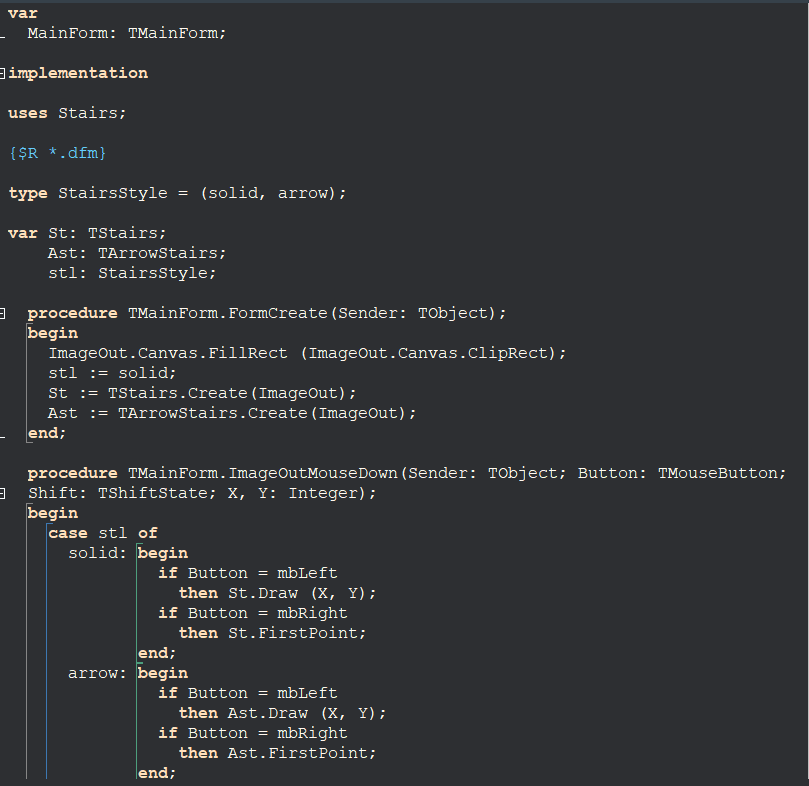


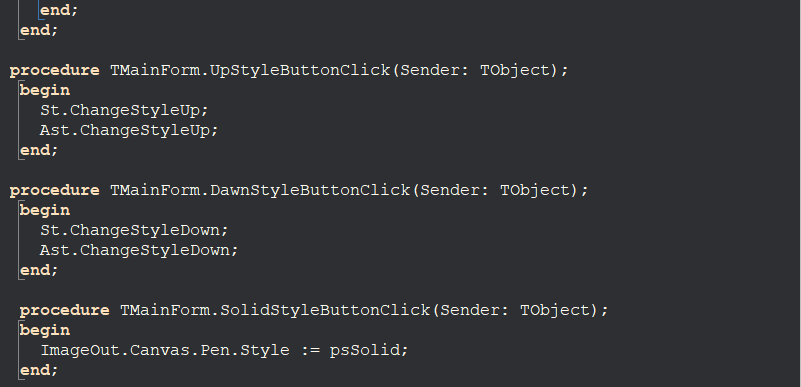
Текст основной программы:

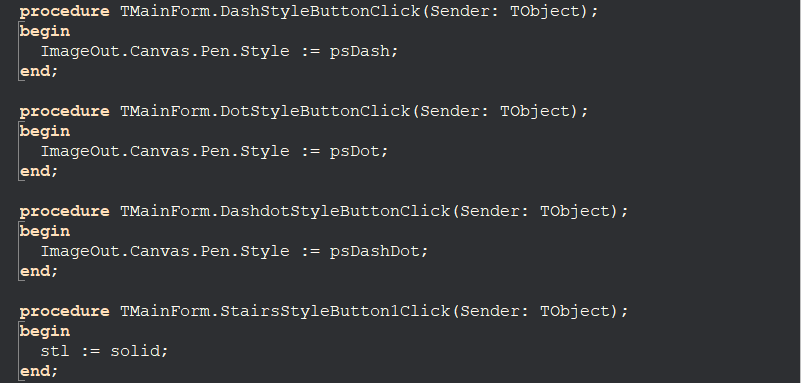


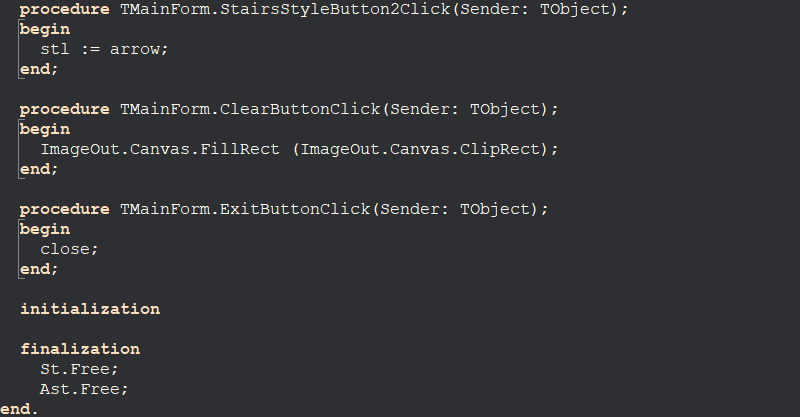
Модуль, реализующий приложение:



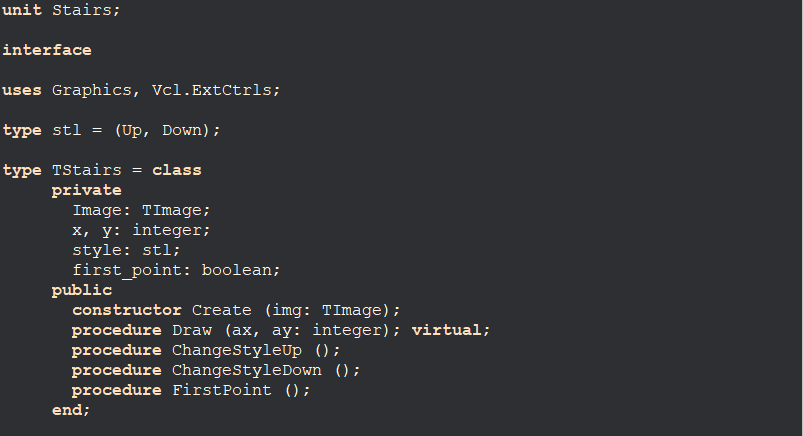


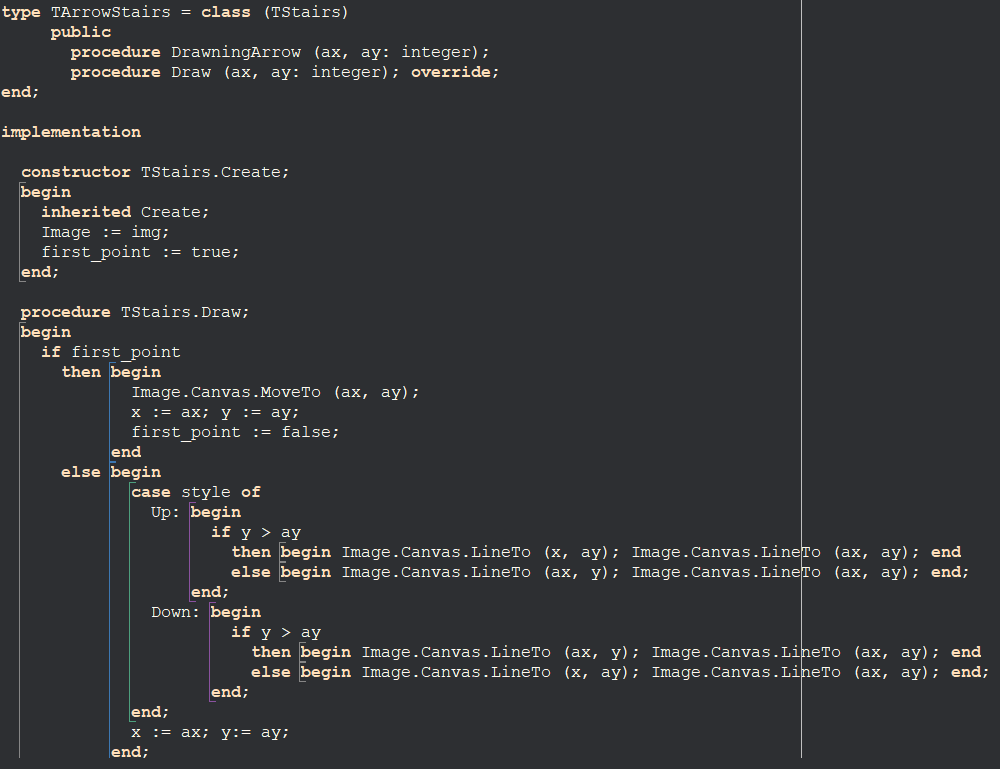


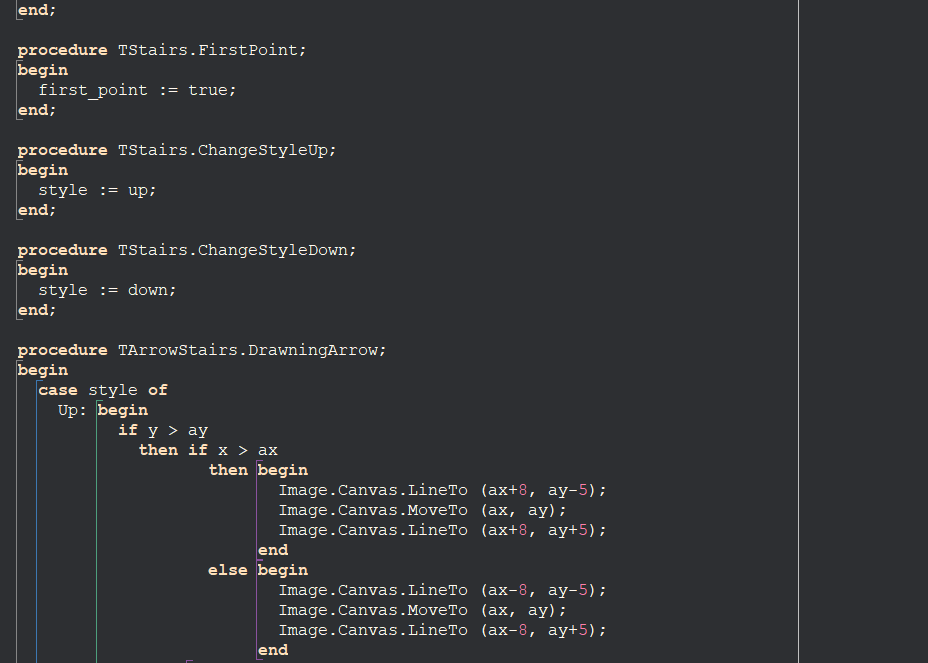


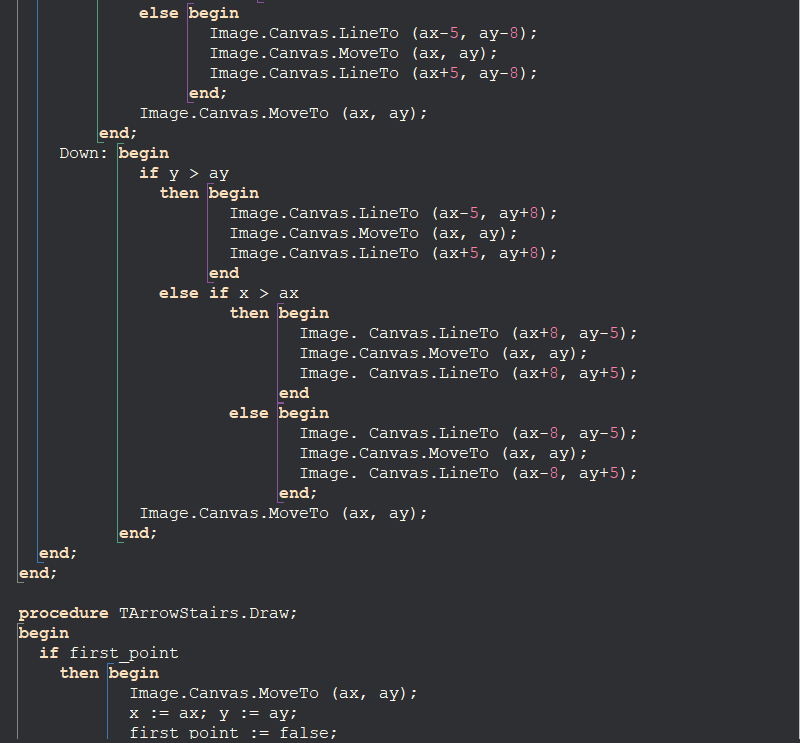


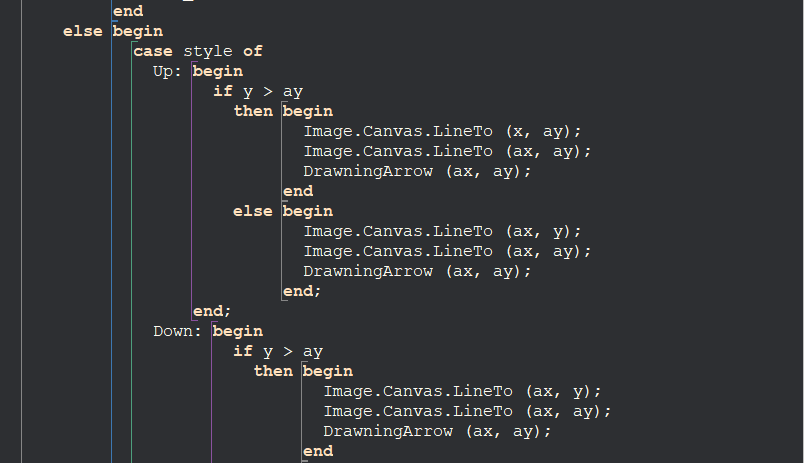
Модуль, реализующий «Лестницы»:

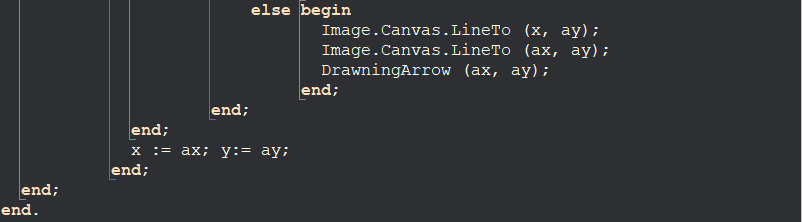




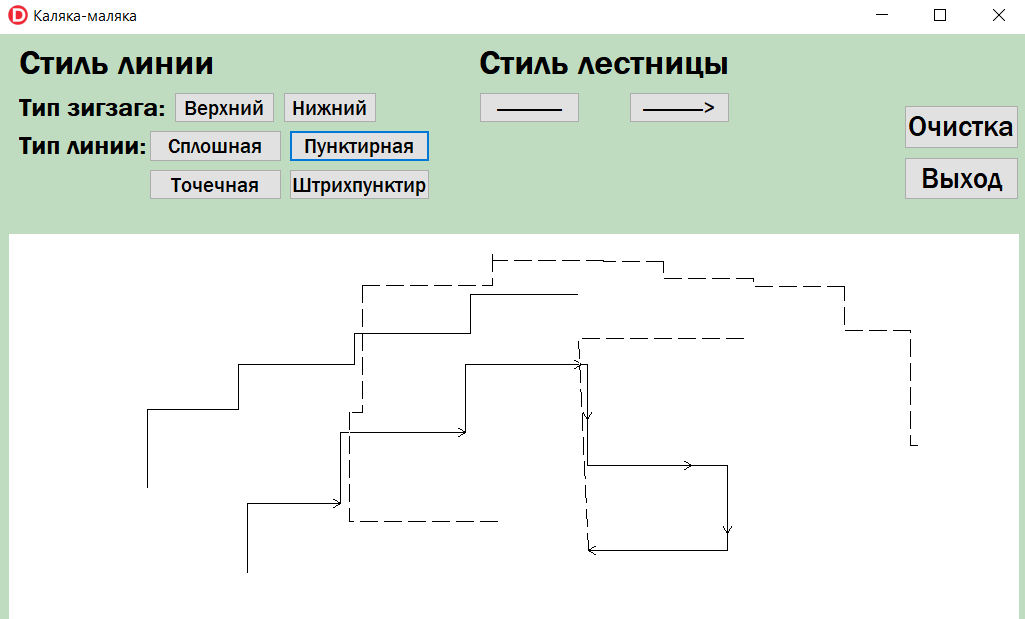








Результат работы:



**Часть 2. Полиморфное наследование**

Разработать программу, содержащую описание трех графических объектов:

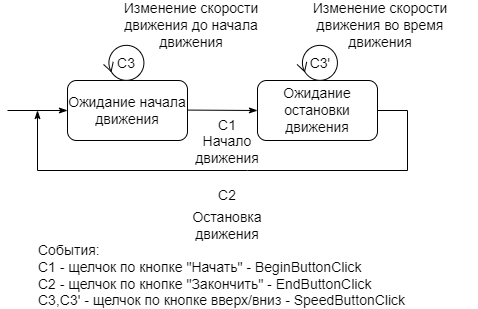
изображения букв: С, О, Q.

Реализуя механизм полиморфизма, привести объекты в вертикальное движение по экрану с различными скоростями с отражением от границ экрана.

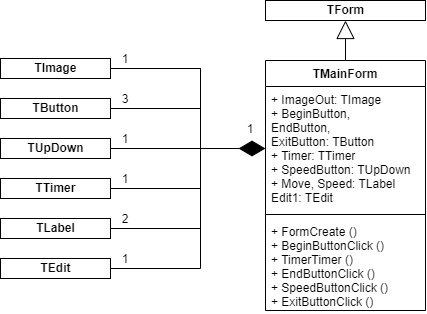
Объектная декомпозиция:

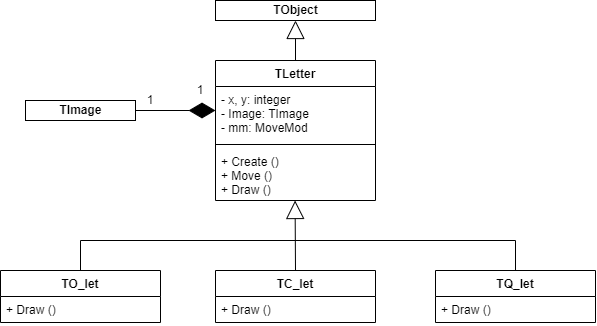


Граф состояний интерфейса:

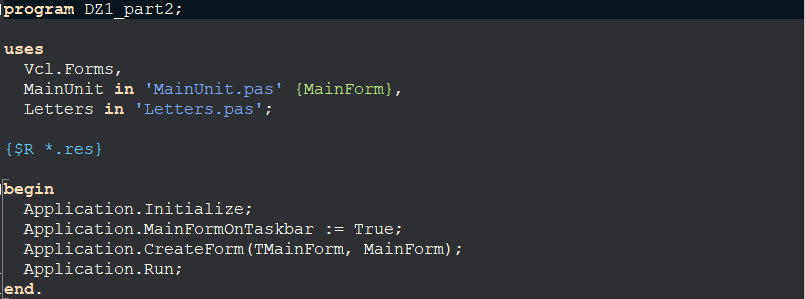


UML-диаграмма классов:

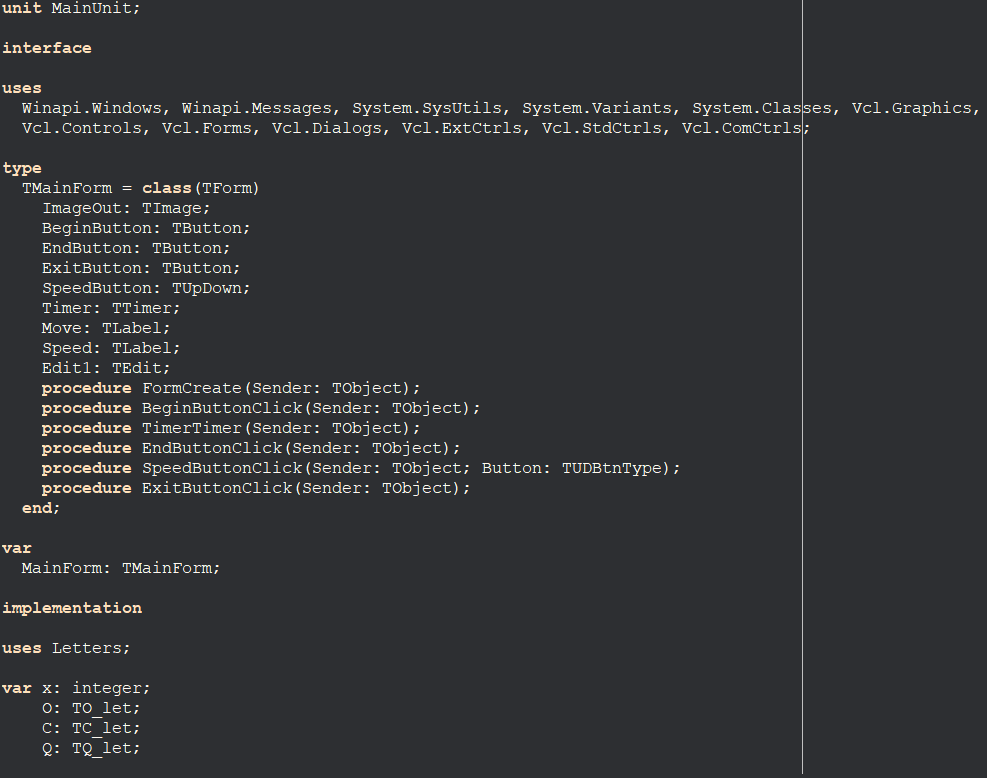


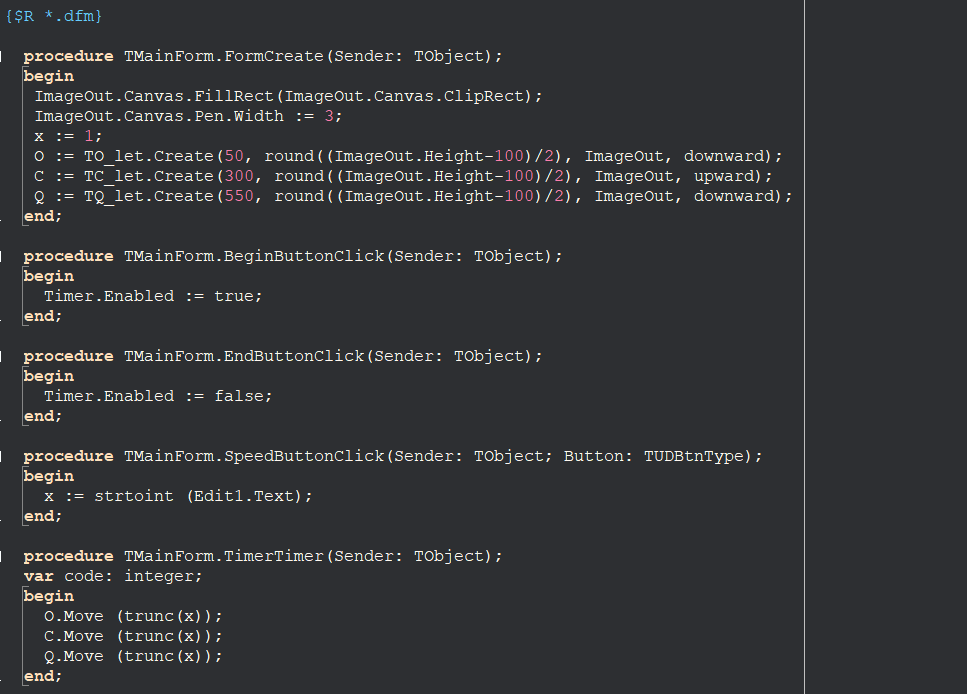


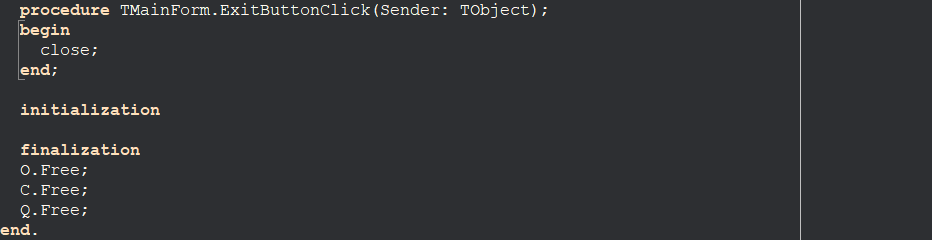
Текст основной программы:



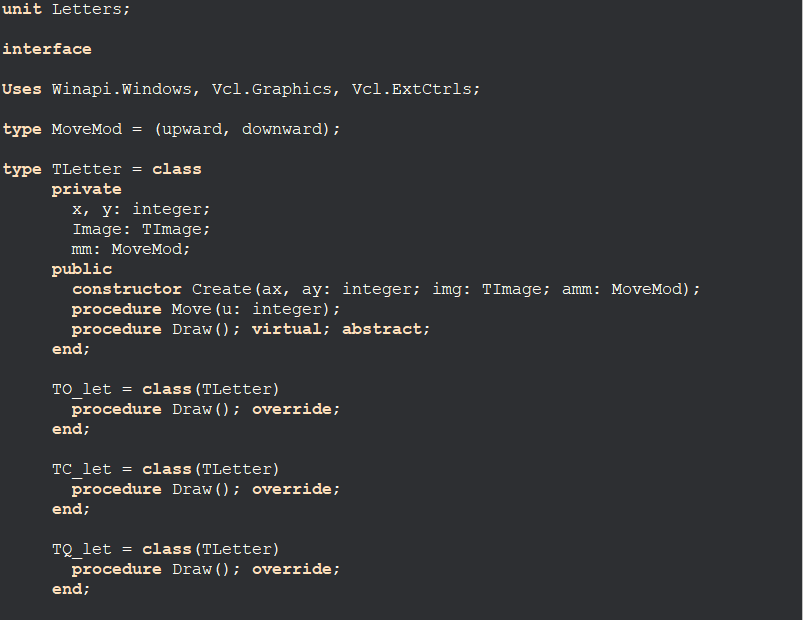
Модуль, реализующий приложение:

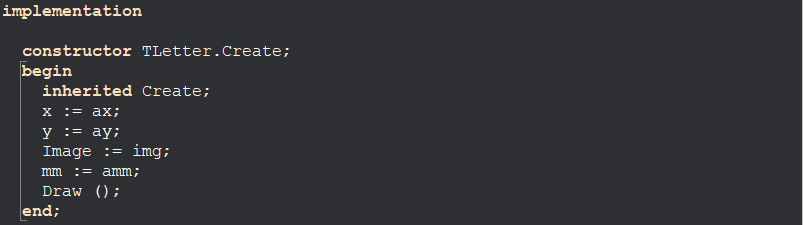


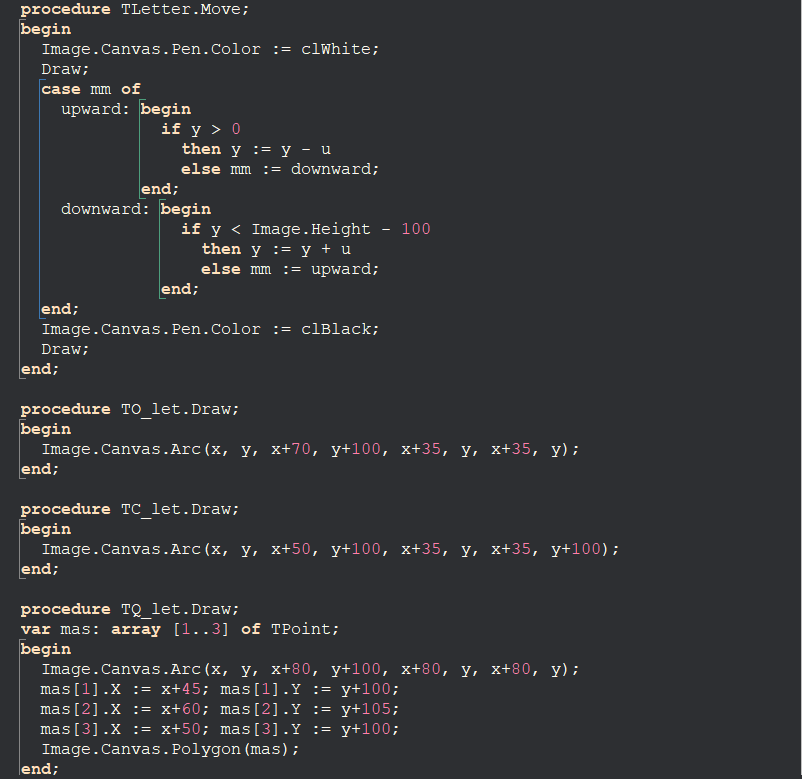




Модуль, реализующий буквы:







Результат работы программы:

