МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Факультет №3. Системы управления, информатика и электроэнергетика.

Кафедра 302

Лабораторная работа по дисциплине «ООП»

Выполнили студенты группы: 3О-208Б:

Чернышевич Е.А.

Макурин В.М.

Принял:

Фетисов А.А.

Москва 2014 год

**Содержание**

1. Формализованная постановка
2. Описание основных алгоритмов
3. UML-диаграммы классов
4. Сведения о реализации
5. Руководство пользователя
6. Тесты
7. Исходный код
8. **Формализованная постановка**

Написать программу, реализующую игру «три в ряд», которая должна соответствовать требованиям реализации в ООП и неким правилам игры.

*Требования реализации в ООП:*

В предметной области должны быть корректно выделены объекты, определены и реализованы их поля и методы.

 Обязательно наличие абстрактных классов.

*Правила игры:*

Уничтожая строчки из 3-х или более одинаковых элементов, включая радужный квадрат, набрать как можно больше очков.

Если ходов уже нет, можно воспользоваться дополнительными элементами (молнией или бомбой).

Молния перемешивает все элементы.

Бомба уничтожает вокруг себя область, размером 3х3.

1. **Описание основных алгоритмов**

*Метод Generate*

// генерация элементов игрового поля

Начало

Создаем матрицу размером 10 на 8 из объектов (цветная ячейка, молния, бомба и радужный квадрат)

Присваиваем ячейке матрицы случайный объект (цветная ячейка, молния, бомба или радужный квадрат)

Конец

*Метод Draw*

// рисование игрового поля

Начало

Рисуем на форме объекты матрицы

Конец

*Метод Scoring*

// подсчитывание количества набранных очков

Начало

Запоминаем текущее количество очков

Подсчитываем очки по горизонтали

Проверяем ячейки справа и слева от текущей ячейки на одинаковость (или на радужный квадрат)

Если проверяемые ячейки равны между собой,

то заменяем найденные одинаковые ячейки другими случайными объектами

и увеличиваем очки на одну единицу

Подсчитываем очки по вертикали

Проверяем ячейки снизу и сверху от текущей ячейки на одинаковость (или на радужный квадрат)

Если проверяемые ячейки равны между собой,

то заменяем найденные одинаковые ячейки другими случайными объектами

и увеличиваем очки на одну единицу

Конец

*Метод FirstClick*

// клик по первому элементу игрового поля

Начало

Проверяем возможность активации при клике на ячейку

Если выбранная ячейка не является бомбой или молнией,

то выделяем выбранную ячейку(затемняем)

иначе подсчитываем очки после активации

Конец

*Метод SecondClick*

// клик по второму элементу игрового поля

Начало

Проверяем корректность второго клика

Если выбранная ячейка является цветной ячейкой (желтой, голубой, зеленой или красной) и выбранная ячейка находится в той же строке или столбце, что и ячейка, которую выбрали при первом клике

то проверяем возможность образования цепочки в результате перемены мест

Если цепочки не образовалось, то снимаем выделение с ячейки из первого клика

Конец

*Метод Chain*

// удаление цепочки из трех или более одинаковых по цвету (в том числе радужного квадрата)

Начало

Запоминаем текущее количество очков

Меняем местами две ячейки с координатами первого и второго клика

Считаем количество очков после перемены мест

Если количество очков не изменилось,

то меняем выбранные ячейки обратно и снимаем выделение с первой ячейки

иначе сохраняем перемену мест и снимаем выделение со второй ячейки

Подсчитываем все очки

Конец

*Метод Swap*

// перестановка местами двух элементов игрового поля

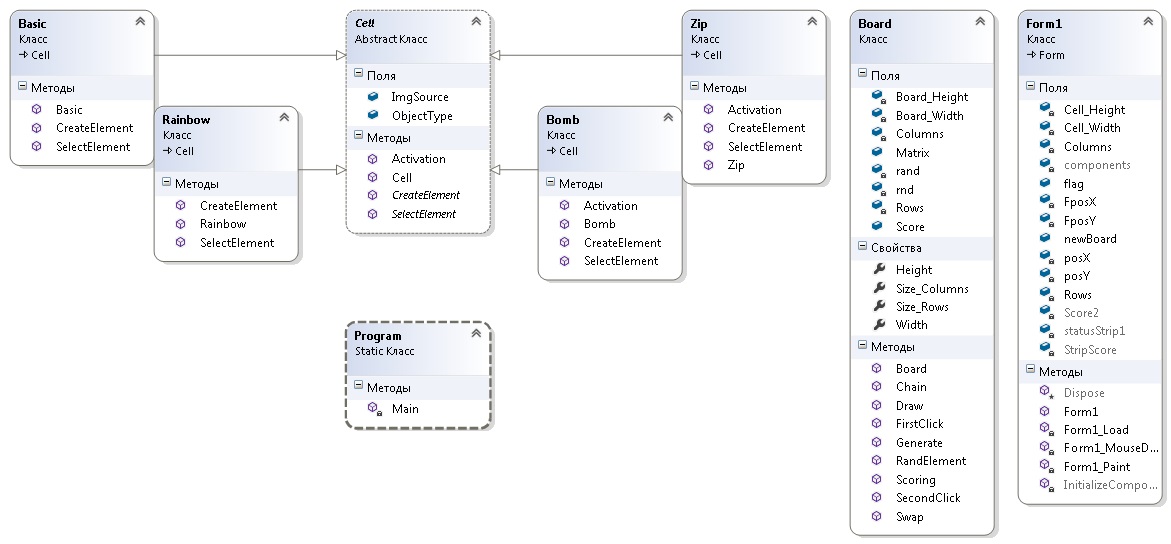
Начало

Сохраняем в буфер первую ячейку

Меняем первую ячейку на вторую

Меняем вторую ячейку на первую (из буфера)

Конец

1. **UML-диаграммы классов**
2. **Сведения о реализации**

*Информация о языке*

Программная реализация выполнена на языке C#

Система программирования: Microsoft Visual Studio 2013

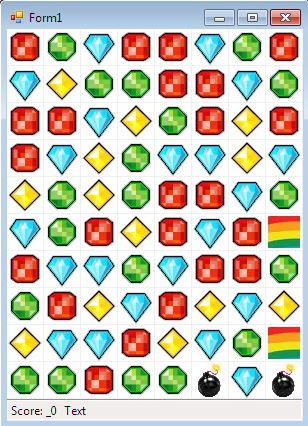
*Системные требования*

* [Операционная система](http://ru.wikipedia.org/wiki/%D0%9E%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0) — Windows XP / Vista / 7 / 8
* Процессор 32-разрядный (x86) / 64-разрядный (x64).
* Видеокарта 32 МБ
* .NET Framework 4.5 (или выше);

1. **Руководство пользователя**

Сначала пользователь открывает папку с приложением. Затем запускает файл «Игра».

На экране появляется следующее окно:



- в нижней строчке пишется количество очков

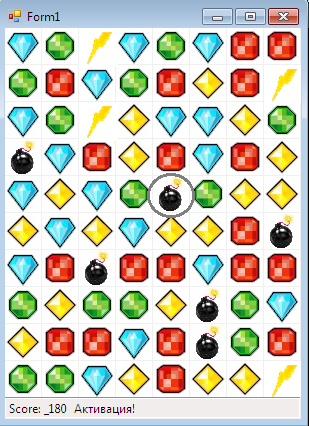
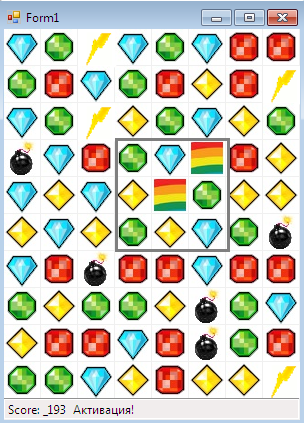
- при составлении цепочки из 3-х и более элементов, эта строчка удаляется

- при нажатии на молнию, элементы перемешиваются

- при нажатии на бомбу, происходит удаление квадрата размером 3х3

1. **Тесты**
2. *Работа программы при нажатии на бомбу*

*Ожидаемый результат: произойдет удаление квадрата размером 3х3*

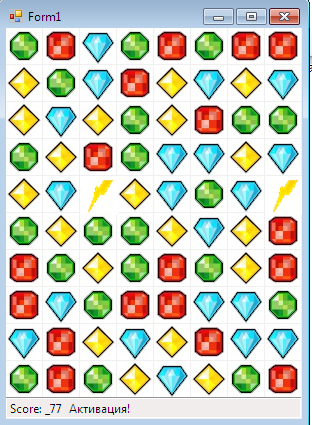
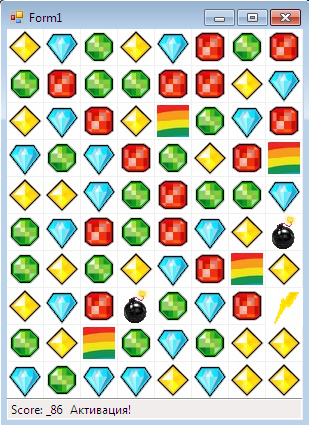


**==>**

*Ожидаемый результат совпал с действительным.*

1. *Работа программы при нажатии на молнию.*

*Ожидаемый результат: все элементы перемешаются*

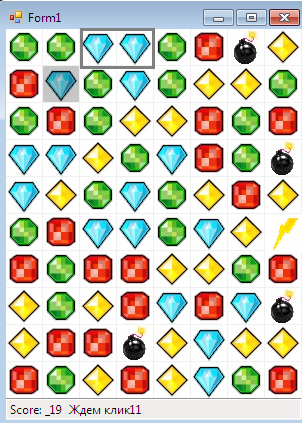
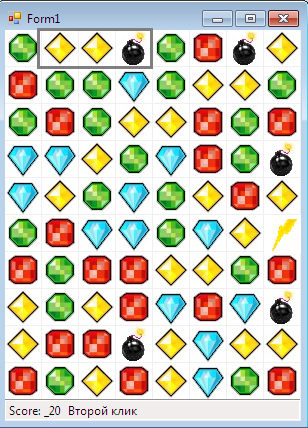


**==>**

*Ожидаемый результат совпал с действительным.*

1. *Работа программы при составлении цепочки из 3-х элементов*

*Ожидаемый результат: вся цепочка удалится*

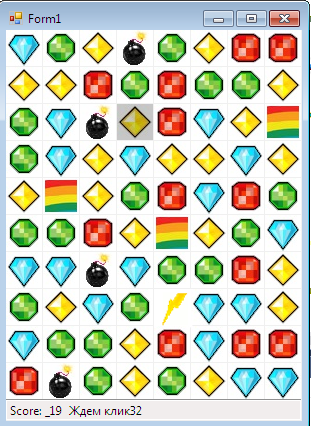
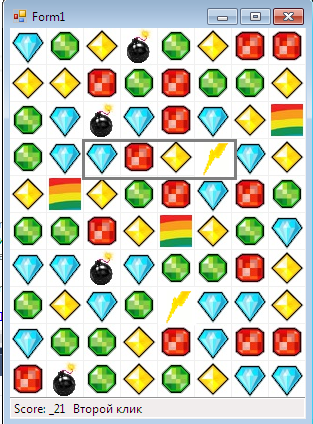


**==>**

*Ожидаемый результат совпал с действительным.*

1. *Работа программы при составлении цепочки из 4-х элементов*

*Ожидаемый результат: вся цепочка удалится*



**==>**

1. **Листинги**

*Текст программ Вы можете посмотреть, перейдя по этой ссылке:*

https://github.com/MAI-302/three\_in\_row/tree/master/Игра