**Московский Авиационный Институт**

**(Национальный Исследовательский Университет)**

**Институт информационных технологий и прикладной математики**

**Кафедра 806 «Вычислительная математика и программирование»**

**Курсовой проект**

**по курсу “ Объектно-ориентированное программирование”**

**Работу выполнил: Махмудов О.С.**

**Группа: М8O-205Б-18**

**Руководитель: Кузнецова С.В**

**Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

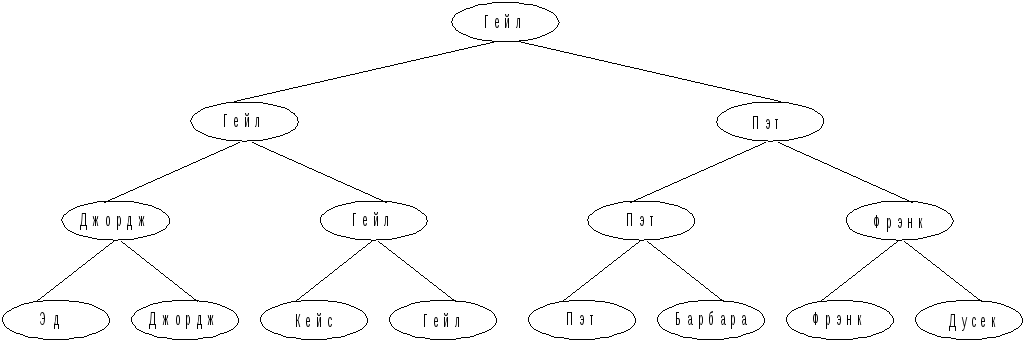
**Москва 2019 г.**

**Постановка задачи**

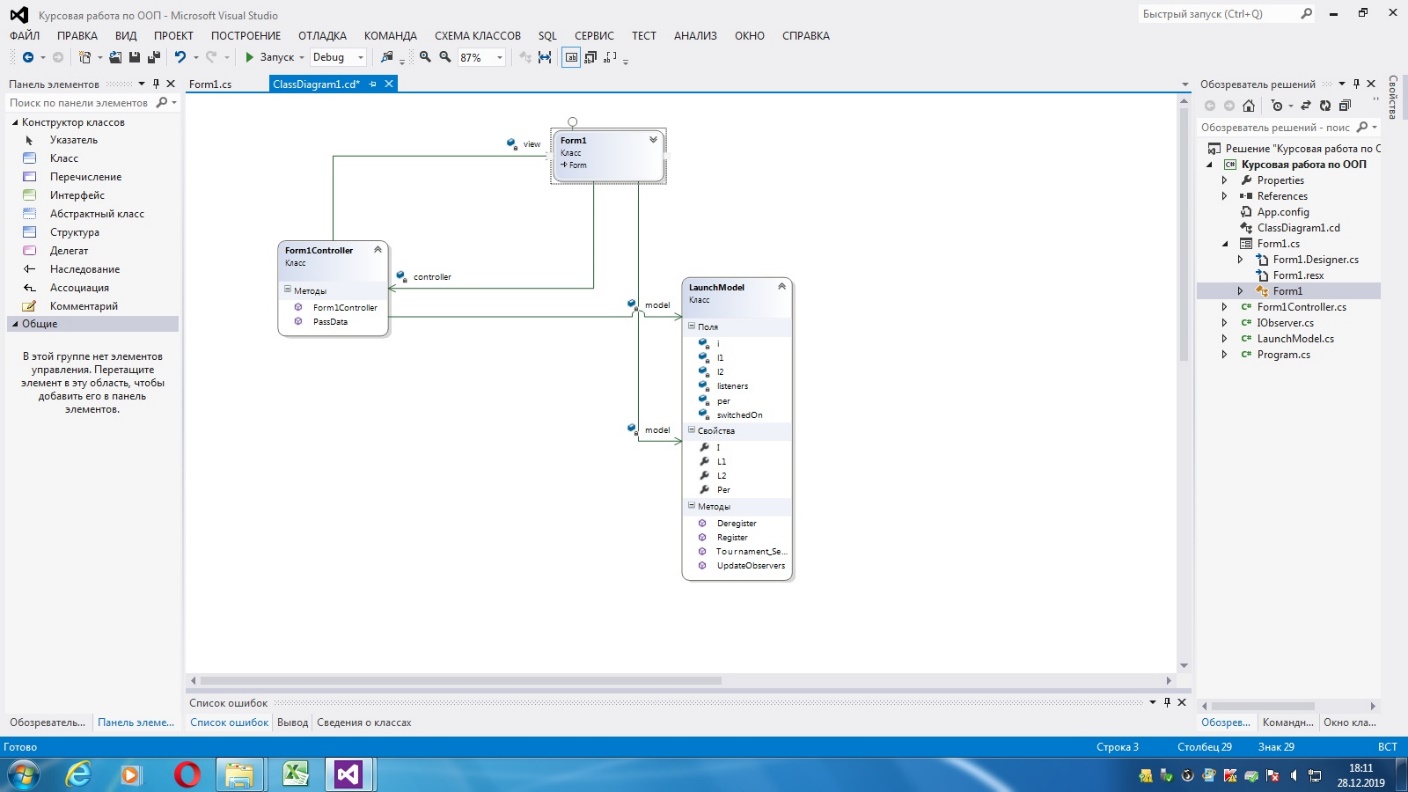
**Вариант 32.** Разработать windows-приложение на языке программирования c#, наглядно демонстрирующее работу алгоритма турнирной выборки (Tournament Selection)

**Описание алгоритма**

Действия метода отражают процесс разыгрывания некоторого турнира, где его участники состязаются друг с другом для определения наилучшего игрока. Эта сортировка также называется сортировкой посредством выбора из дерева. Рассмотрим некоторый турнир в виде дерева. выбора из дерева. Рассмотрим некоторый турнир в виде дерева.

Такое дерево можно понимать следующим образом: Кейс играла в турнире с Гейл и проиграла, т.е. победила Гейл и ее имя помещается в узел дерева и т.д. То есть на представленном дереве показан турнир, в котором победила Гейл.

**Диаграмма классов**

****

**Реализация алгоритма на c#**

public class LaunchModel

{

private bool per;

private int i;

private int l1;

private int l2;

bool switchedOn = false;

public bool Per

{

get { return per; }

}

public int I

{

get { return i; }

}

public int L1

{

get { return l1; }

}

public int L2

{

get { return l2; }

}

public void Tournament\_Selection(int n1, int n2, int i, int l1, int l2)

{

if (n1 > n2)

{

per = true;

this.i = i;

this.l1 = l1;

}

else

{

per = false;

this.i = i;

this.l2 = l2;

}

UpdateObservers();

}

private ArrayList listeners = new ArrayList();

public void Register(IObserver o)

{

listeners.Add(o);

}

public void Deregister(IObserver o)

{

listeners.Remove(o);

}

public void UpdateObservers()

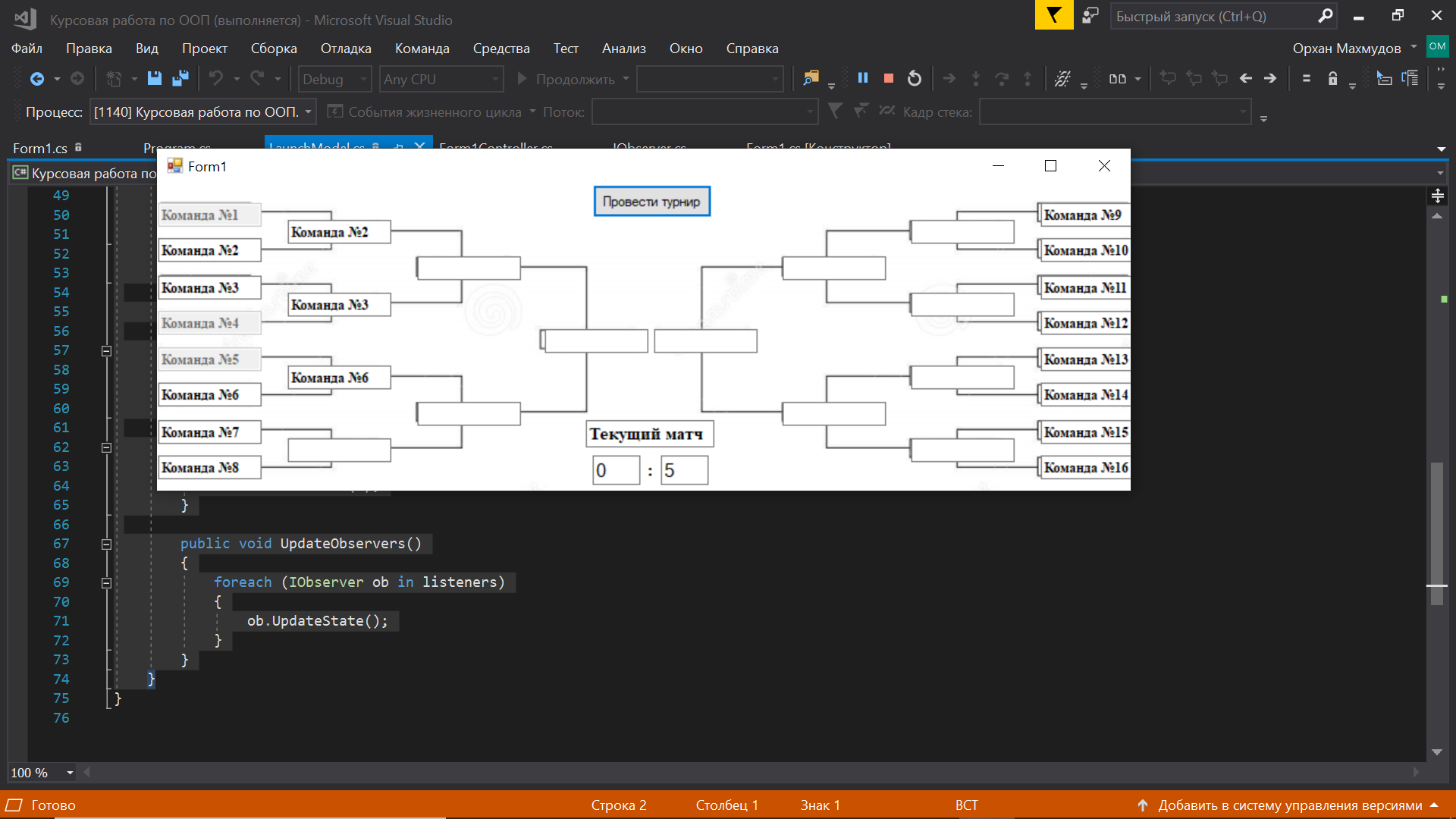
{

foreach (IObserver ob in listeners)

{

ob.UpdateState();

**Графический интерфейc**



**Вывод**

Турнирная выборка – это очень важный и полезный генетический алгоритм, позволяющий отсеивать ненужные данные. Его можно применять в любой таблице, турнирной сетке, базе данных и т.д.

**Оборудование**

*Процессор: Intel core i3 8300*

*Оперативная память: 4Gb*

*Видеокарта: UHD Graphics 620*

**Программное обеспечение**

*Операционная система: windows 10*

*Среда разработки: Visual Studio 2017*

*Версия .net framework: 4.6.4*

**Литература**

# https://studfile.net/preview/2648650/page:10/