

МИНОБРНАУКИ РОССИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

УТВЕРЖДАЮ

Зав.кафедрой,

к. ф.-м. н.

\_\_\_\_\_ С. В. Миронов

**ОТЧЕТ О ПРАКТИКЕ**

студента 4 курса 411 группы факультета КНиИТ

Власова Андрея Александровича

вид практики: учебная

кафедра: математической кибернетики и компьютерных наук

курс: 4

семестр: 2

продолжительность: 2 нед., с 01.07.2020 г. по 14.07.2020 г.

Руководитель практики от университета,

к. ф.-м. н.

\_\_\_\_\_

С. В. Миронов

Руководитель практики от организации (учреждения, предприятия),

к. ф.-м. н.

\_\_\_\_\_

С. В. Миронов

Тема практики: «Создание приложения для анализа генетического алгоритма поиска центральных вершин»

## СОДЕРЖАНИЕ

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	4
ВВЕДЕНИЕ .....	5
1 Использование приложения .....	5
2 Описание технологий и архитектуры приложения .....	6
2.1 Структура базы данных .....	6
Приложение А Нумеруемые объекты в приложении.....	9

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

$|A|$  — количество элементов в конечном множестве  $A$ ;

$\det B$  — определитель матрицы  $B$ ;

ИНС — Искусственная нейронная сеть;

FANN — Feedforward Artifitial Neural Network

## **ВВЕДЕНИЕ**

### **1 Использование приложения**

ывфаыва

## 2 Описание технологий и архитектуры приложения

В качестве языка программирования для решения поставленной задачи был выбран объектноориентированный язык C#. Вместе с тем для создания клиентсерверного приложения был использован фреймворк ASP.NET MVC 5, который позволяет создавать веб-приложения с использованием архитектуры MVC. Кроме этого в качестве системы объектнореляционного отображения используется технология Entity Framework 6. При этом приложение разделено на три слоя абстракции — уровень доступа к данным, уровень бизнеслогики и уровень визуального представления.

### 2.1 Структура базы данных

Хранение графов в базе данных были созданы следующие таблицы: Graphs и Edges. При этом таблица Graphs содержит следующие поля:

- поле Id (типа данных INT) — уникальный идентификатор, внутренний ключ,
- поле N (типа данных INT) — количество вершин в графе,
- поле M (типа данных INT) — количество ребер в графе,
- поле Name (типа данных NVARCHAR) — название графа,
- поле R (типа данных INT) — радиус графа.

Кроме этого таблица Edges состоит из следующих полей:

- поле Id (типа данных INT) — уникальный идентификатор, внутренний ключ,
- поле V1 (типа данных INT) — одна из вершин, которые соединяет ребро,
- поле V2 (типа данных INT) — вторая из вершин, которые соединяет ребро,
- поле Graph\_Id (типа данных INT) — Id графа, которому принадлежит ребро, внешний ключ.

Кроме этого для хранения зарегистрированных пользователей существует таблица Users:

- поле Id (типа данных INT) — уникальный идентификатор, внутренний ключ,
- поле Login (типа данных NVARCHAR) — логин пользователя,
- поле Password (типа данных NVARCHAR) — зашифрованный пароль пользователя.

Полная диаграмма таблиц представлена на рисунке 1:

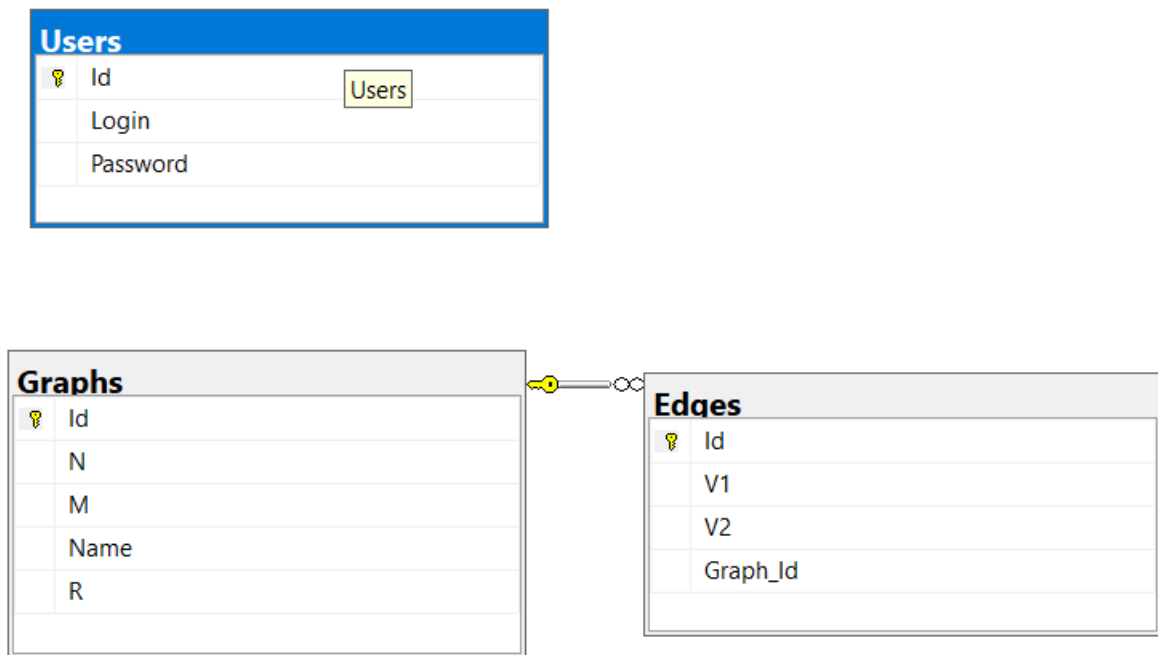


Рисунок 1 – Диаграмма базы данных

Для создания базы данных используется технология Entity Framework, вместе с чем использовался подход Code-first, согласно которому были созданы классы Graph, GraphInfo, Edge, User, описывающие модели данных:

```

1 public class GraphInfo
2 {
3     public int Id { get; set; }
4     public int N { get; set; }
5     public int M { get; set; }
6     public string Name { get; set; }
7     public int R { get; set; }
8 }

1 public class Graph : GraphInfo
2 {
3     public ICollection<Edge> Edges { get; set; }
4     public Graph()
5     {
6         Edges = new List<Edge>();
7     }
8 }

1 public class Edge
2 {
  
```

```
3      public int Id { get; set; }
4      public int V1 { get; set; }
5      public int V2 { get; set; }
6  }
```



## **ПРИЛОЖЕНИЕ А**

### **Нумеруемые объекты в приложении**

HELLO