

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»
Дисциплина: «Программирование»

Контрольное домашнее задание

3 модуль

Вариант 12

Выполнил

Студент группы БПИ173

_____/Д. М. Торилов/

« ____ » _____ 2018 г.

Преподаватель: Максименкова О.В.,

старший преподаватель

департамента

программной инженерии

факультета компьютерных наук

Москва 2017

Содержание

1. Условие задачи	3
2. Функции разрабатываемого приложения	3
2.1. Варианты использования	3
2.2. Описание интерфейса пользователя	3
2.2.1. Панель меню	3
2.2.2. Панель управления	4
2.2.3. Панель работы с таблицей	5
2.2.4. Всплывающие подсказки	5
3. Структура приложения	5
3.1. Диаграмма классов	5
3.2. Описание классов, их полей и методов	7
3.2.1. Jarvis	7
3.2.2. Form	8
3.3. QuakeInfo	10
4. Распределение исходного кода по файлам проекта	11
5. Контрольный пример и описание результатов	12
6. Текст (код) программы	12
6.1. Events Library	12
6.1.1. ViewJarvisMessage	12
7. Список литературы	13

1. Условие задачи

Программа контрольного домашнего задания (КДЗ) должна представлять собой небольшую информационно-справочную систему (ИСС), основанную на файлах. В стандартном файле содержатся данные о землетрясениях. Данные из него загружаются в основную таблицу ИСС.

Далее следует описание задания варианта №12:

Для представления данных о землетрясении использовать класс `EarthQuake`. Координаты землетрясения представлять объектом структуры. Класс `QuakeInfo` связан с объектами `EarthQuake` отношением агрегации и позволяет получать списки землетрясений, сгруппированные по количеству уловивших их станций; списки землетрясений с максимальной магнитудой; землетрясение произошедшее на минимальной и максимальной глубине. Модифицировать интерфейс так, чтобы указанные данные можно было отобразить.

2. Функции разрабатываемого приложения

2.1. Варианты использования

Данная ИСС может быть использована для проведения исследований в области изучения землетрясений, в том числе в научных и образовательных целях.

2.2. Описание интерфейса пользователя

Интерфейс программы реализован на русском языке. В реализации использована технология Windows Forms. В дальнейшем будет совершён переход на технологию WPF.

Интерфейс программы включает в себя:

2.2.1. Панель меню

1. Кнопка **Файл**, даёт доступ к кнопкам:

1.1. Кнопка **Открыть**, позволяет вызвать меню выбора файла для открытия

1.2. Кнопка **Сохранить**, позволяет вызвать меню выбора файла для сохранения, либо сохранить в уже открытый файл, даёт доступ к кнопкам:

1.2.1. **Дозаписать**, позволяет вызвать меню выбора файла для дозаписи таблицы в его конец, либо сохранить в уже открытый файл

1.3. Кнопка **Сохранить как**, позволяет вызвать меню выбора файла для сохранения, даёт доступ к кнопкам:

1.3.1. **Дозаписать**, позволяет вызвать меню выбора файла для дозаписи таблицы в его конец

- 1.4. **Заккрыть**, позволяет прекратить работу с файлом, вызывает диалоговое окно с предложением сохранить файл перед закрытием
2. Кнопка **Таблица**, даёт доступ к кнопкам:
 - 2.1. Кнопка **Сортировка**, даёт доступ к кнопкам:
 - 2.1.1. **По номеру**, позволяет отсортировать таблицу по возрастанию идентификационных номеров элементов
 - 2.1.2. **По станциям**, позволяет отсортировать таблицу по возрастанию числа в количественной характеристике станций, засёкших землетрясение, соответствующее элементу
 - 2.2. Кнопка **Удалить выделенную строку**, позволяет удалить выделенную строку из таблицы
3. Кнопка **Информация**, даёт доступ к кнопкам:
 - 3.1. **Предельные значения**, позволяет увидеть короткий список диапазонов изменения характеристических величин элементов таблицы
 - 3.2. **О программе**, позволяет увидеть информацию об авторе программы

2.2.2. Панель управления

1. Поле **Добавление элемента** включает в себя
 - 1.1. Поле ввода **Номер элемента**
 - 1.2. Поле ввода **Широта элемента**
 - 1.3. Поле ввода **Долгота элемента**
 - 1.4. Поле ввода **Глубина элемента**
 - 1.5. Поле ввода **Магнитуда элемента**
 - 1.6. Поле ввода **Станции элемента**
 - 1.7. Кнопка **Добавить**, создаёт элемент с характеристиками 1.1 - 1.6 и добавляет его в таблицу
2. Поле **Землетрясения** включает в себя два подполя, **Максимальная глубина** (информация о землетрясении с максимальной глубиной) и **Минимальная глубина** (информация о землетрясении с минимальной глубиной), несущие полную информацию о таковых в таблице в режиме реального времени
3. Поле **Фильтрация по магнитуде** включает в себя два
 - 3.1. Поле ввода **Магнитуда** для ввода вещественного значения магнитуды
 - 3.2. Кнопка **Фильтровать**, при нажатии удаляет из таблицы все данные о землетрясениях, магнитудой ниже чем в поле 3.1

4. Кнопка **Выход** позволяет выйти из приложения

2.2.3. Панель работы с таблицей

Представляет из себя матрицу неограниченного числа строк и 6 столбцов с возможностью редактирования

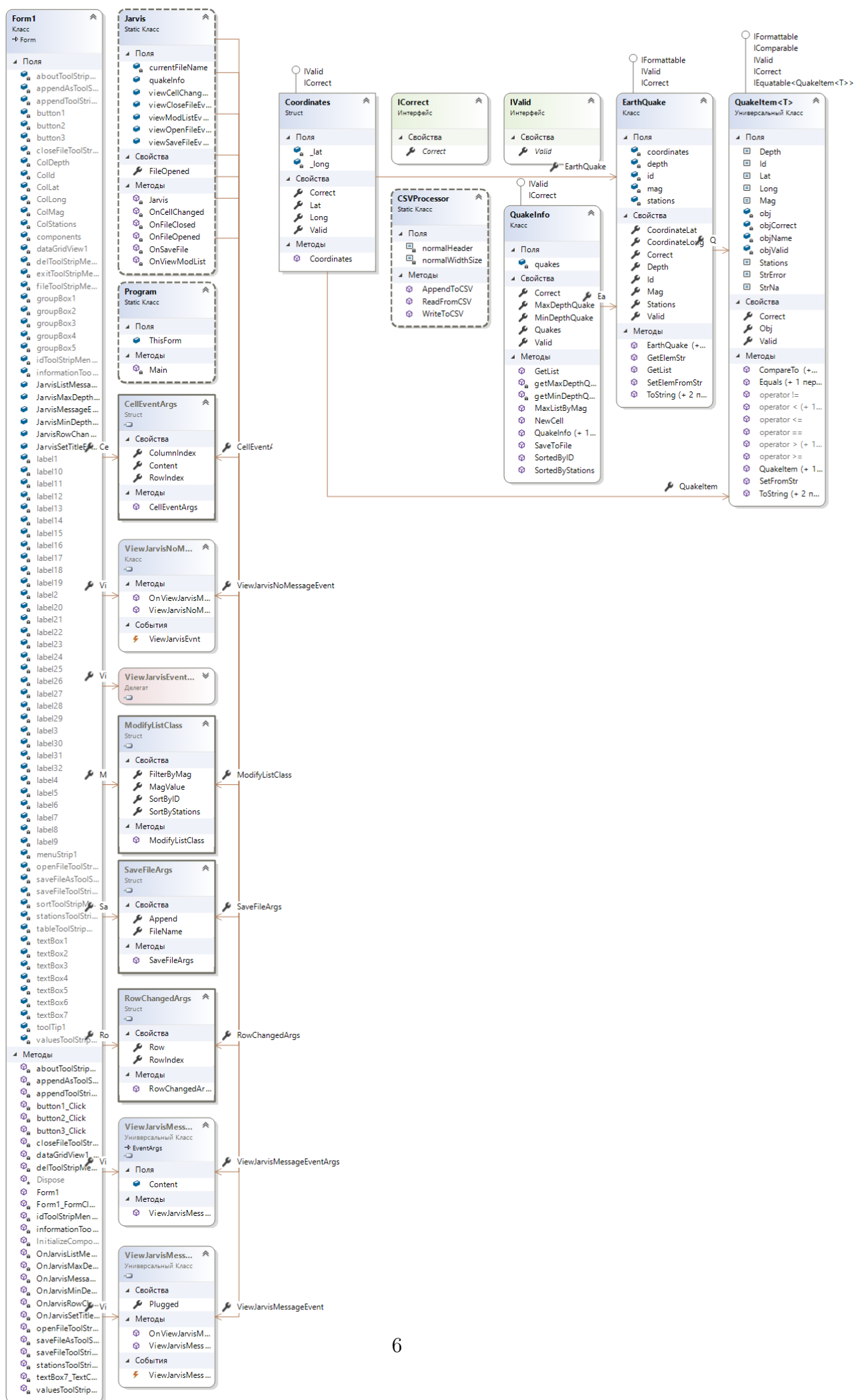
2.2.4. Всплывающие подсказки

Появляются при наведении курсора мыши на необходимый элемент. Несут дополнительную информацию об объекте.

3. Структура приложения

Приложение реализовано с использованием паттерна Model-View-Controller[2]. Model реализована объектом класса QuakeInfo. View реализована объектом Form1. Задачу Controller выполняет статический класс Jarvis. Ниже приведены необходимые для понимания детали структуры приложения.

3.1. Диаграмма классов



3.2. Описание классов, их полей и методов

3.2.1. Jarvis

Выполняет задачу синхронизации Model и View.

Объект класса QuakeInfo связан со статическим классом Jarvis отношением агрегации.

С объектом Form1 реализовано гибкое общение через события[7] с обобщениями[5][6].

Поля:

- `public static QuakeInfo quakeInfo` - Модель агрегирована в контроллер
- `private static string currentFileName = null;` - Имя файла, с которым ведётся работа

Свойства:

- `public static bool FileOpened => currentFileName != null;` - Свойство корректного открытия файла

Методы:

- `private static void OnViewModList(object sender, ViewJarvisMessageEventArgs<ModifyListClass> messageEventArgs)` - Обработчик вызова модификатора списка (сортировки и фильтрации)
- `private static void OnFileOpened(object sender, ViewJarvisMessageEventArgs<string> messageEventArgs)` - Обработчик открытия файла
- `private static void OnSaveFile(object sender, ViewJarvisMessageEventArgs<SaveFileArgs> messageEventArgs)` - Обработчик сохранения файла
- `private static void OnFileClosed()` - Обработчик закрытия файла
- `private static void OnCellChanged(object sender, ViewJarvisMessageEventArgs<CellEventArgs> messageEventArgs)` - Обработчик изменения ячейки
- `static Jarvis()` - Конструктор

События:

- `public static ViewJarvisMessageEvent<string> viewOpenFileEvent = new ViewJarvisMessageEvent<string>();` - Событие обработчика открытия файла
- `public static ViewJarvisMessageEvent<SaveFileArgs> viewSaveFileEvent = new ViewJarvisMessageEvent<SaveFileArgs>();` - Событие обработчика сохранения файла
- `public static ViewJarvisNoMessageEvent viewCloseFileEvent = new ViewJarvisNoMessageEvent();` - Событие закрытия файла

- `public static ViewJarvisMessageEvent<CellEventArgs> viewCellChangedEvent = new ViewJarvisMessageEvent<CellEventArgs>();` - Событие обработчика изменения ячейки таблицы
- `public static ViewJarvisMessageEvent<ModifyListClass> viewModListEvent = new ViewJarvisMessageEvent<ModifyListClass>();` - Событие обработчика модификации списка

3.2.2. Form

Представляет реализацию View[8][3]. Обменивается данными с Jarvis через события.

Методы:

- `private void OnJarvisMessageEvent(object sender, ViewJarvisMessageEventArgs<string> messageEventArgs)` - Обработчик события получения сообщения от класса Jarvis
- `private void OnJarvisListMessageEvent(object sender, ViewJarvisMessageEventArgs<List<List<string>> messageEventArgs)` - Обработчик события получения таблицы от класса Jarvis
- `private void OnJarvisSetTitleEvent(object sender, ViewJarvisMessageEventArgs<string> messageEventArgs)` - Обработчик события вывода сообщения
- `private void OnJarvisRowChangedEvent(object sender, ViewJarvisMessageEventArgs<RowChangedEventArgs> messageEventArgs)` - Обработчик события изменения строки
- `private void OnJarvisMaxDepthUpdatedEvent(object sender, ViewJarvisMessageEventArgs<List<string> messageEventArgs)` - Обработчик обновления информации об элементе с самой большой характеристикой глубины
- `private void OnJarvisMinDepthUpdatedEvent(object sender, ViewJarvisMessageEventArgs<List<string> messageEventArgs)` - Обработчик события обновления информации об элементе с минимальной характеристикой глубины
- `public Form1()` - Конструктор View
- `private void openFileToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик нажатия кнопки открытия файла
- `private void closeFileToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик нажатия кнопки закрытия файла
- `private void dataGridView1CellValueChanged(object sender, DataGridViewCellEventArgs e)` - Обработчик изменения ячейки таблицы
- `private void aboutToolStripMenuItemClick(object sender, EventArgs e)` - Справка

- `private void valuesToolStripMenuItemClick(object sender, EventArgs e)` - Предельные величины
- `private void button1 Click(object sender, EventArgs e)` - Кнопка выхода - обработчик
- `private void button2 Click(object sender, EventArgs e)` - Обработчик нажатия кнопки добавления строки
- `private void Form1 FormClosing1(object sender, FormClosingEventArgs e)` - Обработчик закрытия формы
- `private void idToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик вызова сортировки по ID
- `private void stationsToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик вызова сортировки по станциям
- `private void button3 Click(object sender, EventArgs e)` - Обработчик нажатия кнопки фильтрации
- `private void delToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик нажатия кнопки удаления строки в таблице
- `private void appendToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик нажатия кнопки дозаписи в файл
- `private void appendAsToolStripMenuItem1Click(object sender, EventArgs e)` - Обработчик нажатия кнопки дозаписи в файл
- `private void saveFileAsToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик нажатия кнопки сохранения файла
- `private void saveFileToolStripMenuItemClick(object sender, EventArgs e)` - Обработчик нажатия кнопки сохранения файла

События:

- `public static ViewJarvisMessageEvent<string> JarvisMessageEvent = new ViewJarvisMessageEvent<string>();` - Событие получения сообщения от класса Jarvis
- `public static ViewJarvisMessageEvent<List<List<string>>> JarvisListMessageEvent = new ViewJarvisMessageEvent<List<List<string>>>();` - Событие получения таблицы от класса Jarvis
- `public static ViewJarvisMessageEvent<string> JarvisSetTitleEvent = new ViewJarvisMessageEvent<string>();` - Событие вывода сообщения

- `public static ViewJarvisMessageEvent<RowChangedArgs> JarvisRowChangedEvent = new ViewJarvisMessageEvent<RowChangedArgs>();` - Событие изменения строки
- `public static ViewJarvisMessageEvent<List<string>> JarvisMaxDepthUpdatedEvent = new ViewJarvisMessageEvent<List<string>>();` - Событие обновления информации об элементе с самой большой характеристикой глубины
- `public static ViewJarvisMessageEvent<List<string>> JarvisMinDepthUpdatedEvent = new ViewJarvisMessageEvent<List<string>>();` - Событие обновления информации об элементе с минимальной характеристикой глубины

3.3. QuakeInfo

Описывает модель. Агрегирована в контроллер.

Выполняет файловый ввод-вывод[4][1], проверяет корректность введённых координат[10], глубин[9], магнитуд[11].

Поля

- `private List<EarthQuake> quakes = new List<EarthQuake>();` - Список землетрясений

Свойства

- `public List<EarthQuake> Quakes` - Свойство списка землетрясений
- `public bool Valid` - Реализация интерфейса `IValid`
- `public bool Correct` - Реализация интерфейса `ICorrect`
- `public List<string> MinDepthQuake => getMinDepthQuake().GetList();` - Информация о землетрясении на минимальной глубине. При добавлении элемента происходит полная проверка всего списка. Требуется уменьшения сложности алгоритма. Реализована за линейную сложность из-за сжатых сроков сдачи КДЗ.
- `public List<string> MaxDepthQuake => getMaxDepthQuake().GetList();` - Информация о землетрясении на максимальной глубине. При добавлении элемента происходит полная проверка всего списка. Требуется уменьшения сложности алгоритма. Реализована за линейную сложность из-за сжатых сроков сдачи КДЗ.

Методы

- `public QuakeInfo()` - Конструктор объекта
- `public QuakeInfo(string filename)` - Конструктор объекта с чтением списка из файла
- `public List<EarthQuake> SortedByID()` - Возвращает отсортированный по id список землетрясений

- `public List<EarthQuake> SortedByStations()` - Возвращает отсортированный по количеству станций список землетрясений
- `public List<EarthQuake> MaxListByMag(double minValue)` - Возвращает отсортированный по магнитуде список максимальных землетрясений
- `private EarthQuake getMinDepthQuake()` - Поиск землетрясения, произошедшего на минимальной глубине.
- `private EarthQuake getMaxDepthQuake()` - Поиск землетрясения, произошедшего на максимальной глубине.
- `public List<List<string>> GetList(CultureInfo cultureInfo=null)` - Получить представление в виде списка списков строк
- `public List<string> NewCell(string val, int columnIndex, int rowIndex, CultureInfo cultureInfo)` - Установка нового значения ячейки и возвращение нового списка строк строки таблицы
- `public void SaveToFile(string filename, string mode)` - Запись и дозапись списка в файл

4. Распределение исходного кода по файлам проекта

Описание модели содержится в файле `QuakeInfo.cs`. Все файлы из библиотеки классов `ModelLibrary` являются вспомогательными.

Описание контроллера содержится в файле `Jarvis.cs`.

Работа с интерфейсом описана в `Form1.cs`. События для обмена информацией между `Jarvis` и `Form1` описаны в библиотеке классов `EventsLibrary`.

5. Контрольный пример и описание результатов

Землетрясения ИСС

Файл Таблица Информация

	№	Широта	Долгота	Глубина	Магнитуда	Станции
»*						

Выход

Землетрясения

Максимальная глубина

№: NA
Широта: NA
Долгота: NA
Глубина: NA
Магнитуда: NA
Станции: NA

Минимальная глубина

№: NA
Широта: NA
Долгота: NA
Глубина: NA
Магнитуда: NA
Станции: NA

Фильтрация по магнитуде

Работать с элементами магнитудой больше

Фильтровать

Добавление элемента

№ Широта Долгота Глубина Магнитуда Станции

Добавить

6. Текст (код) программы

6.1. Events Library

6.1.1. ViewJarvisMessage

7. Список литературы

Список литературы

- [1] Andrew. `openfiledialog`, `savefiledialog`. <https://ru.stackoverflow.com/questions/87480/openfiledialog-savefiledialog>.
- [2] cobiot. Охота на мифический `mvc`. Построение пользовательского интерфейса. <https://habrahabr.ru/post/322700/>.
- [3] Microsoft. Класс `datagridview`. [https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/system.windows.forms.datagridview(v=vs.110).aspx).
- [4] Microsoft. Метод `streamreader.read()`. [https://msdn.microsoft.com/ru-ru/library/ath1fht8\(v=vs.110\).aspx](https://msdn.microsoft.com/ru-ru/library/ath1fht8(v=vs.110).aspx).
- [5] professorweb.ru. Обобщенные классы. https://professorweb.ru/my/csharp/charp_theory/level11/11_2.php.
- [6] professorweb.ru. Обработка событий в среде `.net framework`. https://professorweb.ru/my/csharp/charp_theory/level10/10_9.php.
- [7] professorweb.ru. События. https://professorweb.ru/my/csharp/charp_theory/level10/10_7.php.
- [8] theLotus. Changing the form title text programmatically. <https://social.msdn.microsoft.com/Forums/en-US/b958c565-94bb-464d-9075-5163384d3887/changing-the-form-title-text-programmatically?forum=winforms>.
- [9] Wikipedia. Earthquake. <https://en.wikipedia.org/wiki/Earthquake>.
- [10] Wikipedia. Geographic coordinate system. https://en.wikipedia.org/wiki/Geographic_coordinate_system.
- [11] Wikipedia. Moment magnitude scale. https://en.wikipedia.org/wiki/Moment_magnitude_scale.