Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Иркутский национальный исследовательский технический

университет»

Институт информационных технологий и анализа данных

**О Т Ч Ё Т**

|  |  |  |
| --- | --- | --- |
| о прохождении | | производственной практики |
|  | | (вид практики: учебная/производственная) |
| преддипломная практика | | |
| (тип практики: технологическая/научно-исследовательская работа/преддипломная и др.) | | |
|  | | |
| в | Отдел информационных систем | |
|  | (наименование структурного подразделения) | |

|  |  |  |
| --- | --- | --- |
| Обучающегося | Вовикова Д.Е., ИСМб-19-1 | |
|  | (ФИО, группа, подпись) | |
| Руководитель практики от института ИТиАД | | |
| Черкашин Е.А., доцент | | |
|  | (ФИО, должность, подпись) | |
| Руководитель практики от структурного подразделения | | |
| Журавлева Н.А., программист | | |
|  | (ФИО, должность, подпись) | |
| Оценка по практике | |  |
|  | | |
|  | | (ФИО, подпись, дата) |

Содержание отчета на \_\_\_ стр. Приложение к отчету на \_\_\_ стр.

Иркутск 2023

**Содержание**

[Введение 3](#_Toc135219534)

[1. Анализ предметной области 4](#_Toc135219535)

[1.1 Современные методы исследования землетрясений 4](#_Toc135219536)

[1.2 Анализ существующих систем, направленных на анализ сейсмического процесса 5](#_Toc135219537)

[1.2.1 Earthquake Hazards Program 5](#_Toc135219538)

[1.2.2 БАЗА ДАННЫХ «ЗЕМЛЕТРЯСЕНИЯ РОССИИ» 8](#_Toc135219539)

[1.3 Описание программного пакета 10](#_Toc135219540)

[1.4 Подходы к реинжинирингу 14](#_Toc135219541)

[2. Выбор технологий и средств разработки 16](#_Toc135219542)

[3. Описание текущих возможностей реализованного программного обеспечения 17](#_Toc135219543)

[3.1 Реализация функции «Указать параметры приложения» 17](#_Toc135219544)

[3.2 Реализация функции «Указать энергетические классы» 18](#_Toc135219545)

[3.3 Реализация функции «Указать временной промежуток» 19](#_Toc135219546)

[3.4 Реализация функции «Указать число временных интервалов» 20](#_Toc135219547)

[3.5 Реализация функции «Выбор района исследования» 21](#_Toc135219548)

[3.6 Реализация функций «Отобразить топографию» и «Отобразить эпицентры» 21](#_Toc135219549)

[3.7 Реализация функции «Сохранить карту» 23](#_Toc135219550)

[3.8 Реализация функции «Загрузить данные об эпицентрах» 24](#_Toc135219551)

[3.9 Реализация функции «Загрузить данные о разломах» 25](#_Toc135219552)

[3.10 Реализация функции «Генерировать отчет» 27](#_Toc135219553)

[3.11 Реализация функции «Сохранить график» 28](#_Toc135219554)

[Заключение 29](#_Toc135219555)

[Список использованных источников 30](#_Toc135219556)

Введение

Преддипломная практика проходила в институте систем энергетики им. Л.А. Мелентьева СО РАН, отдел систем искусственного интеллекта.

В ходе практики необходимо: изучить деятельности Отдела систем искусственного интеллекта в энергетике, смоделировать бизнес-процессы поставленных задач, обосновать необходимости выбора базовых программных средств, разработать собственного решения, сформировать требования-спецификацию, спроектировать программное и информационное обеспечение и пользовательский интерфейс, разработать алгоритмы, методики, технологии и программный комплекс в соответствии с индивидуальным заданием руководителя практики от предприятия.

При прохождении производственной практики предстоит выполнить следующие задачи: описать результаты анализа предметной области и выявленных проблем, обосновать предложенные проектные решения, описать текущие возможности реализованного программного обеспечения, составить отчёт по практике, оформленный в соответствии с требованиями СТО 005 ИРНИТУ

1. Анализ предметной области

1.1 Современные методы исследования землетрясений

Изучение землетрясений и оценка их параметров являются одной и важнейших задач сейсмологии так, как предсказание землетрясений поможет улучшить сейсмобезопасность и позволит минимизировать ущерб, наносимый природой. Существует большое количество институтов, занимающихся анализом и прогнозированием землетрясений. Землетрясение – это подземные толчки и колебания земной поверхности, возникающие в результате внезапных смещений и разрывов в земной коре или верхней мантии и передающиеся на большие расстояния в виде упругих колебаний [1]. Сейсмограмма необходима, что зафиксировать землетрясения для дальнейшего анализа. Сейсмограмма – запись колебаний почвы в виде зигзагообразной линии, полученная при помощи сейсмографа [2]. Данный прибор был изобретен в 132 году китайским ученным Чжан Хэном. Сейсмограмма представляет собой график движения грунта в зависимости от времени. Несмотря на то, что человек не замечает движения земной поверхности, сейсмограмма не бывает прямой, потому что современные сейсмографы крайне чувствительны и регистрируют даже малейшее колебания [3]. Землетрясения оцениваются по шкале Рихтера или ML, предложенной 1935 году американским сейсмологом Чарльзом М. Фрэнсисом Рихтеров и немецким ученным Бено Гутенбергом. Для получения магнитуды землетрясений используют сейсмограмму [4].

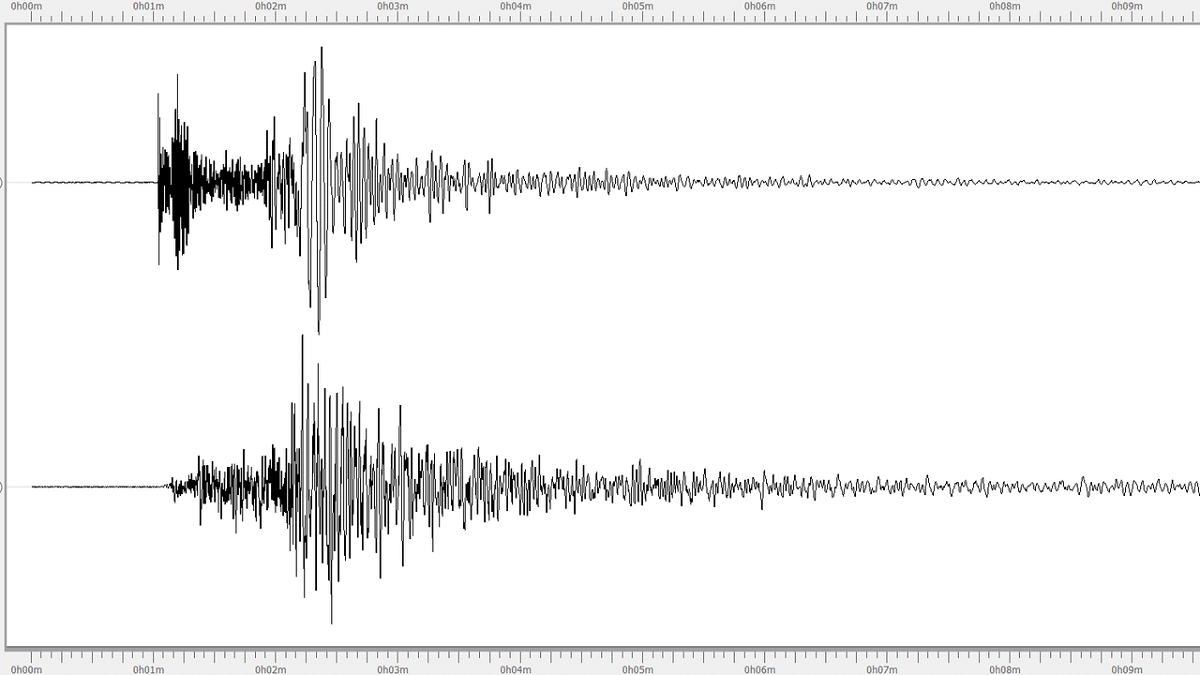


Рисунок 1 – Пример сейсмограммы

. Современные технологии предоставляют большой спектр инструментов, помогающих при работе с землетрясениями. Одним из ключевых элементов оценки землетрясений является геоинформационная система, позволяющая определить места с наиболее сейсмоактивной зоной. Эпицентры землетрясений можно нанести на карту и сопоставить с разломами. За время существования сейсмологии, ученные смогли создать множество способов анализа землетрясений и предпринимаются попытки их предсказания.

В основе предсказаний землетрясений лежит метод EPA (Earthquake-Prone Areas). В классической формулировке Гельфанда-Кейлиса-Борока проблема EPA формулируется следующим образом. В рассматриваемом сейсмически активном районе определяются места, в которых возможны сильные землетрясения. Затем проводится поиск таких мест среди объектов распознавания в этой зоне. Набор объектов распознавания W необходимо разделить на два непересекающихся класса: класс B, состоящий из объектов, которые могут вызвать сильное землетрясение в своей области, и класс H, состоящий из объектов, которые не могут вызвать такое землетрясение. Для выполнения этой классификации применяется алгоритм распознавания образов на основе обучения, при этом обучающая выборка W0 определяется на основе местной сейсмической информации и состоит из двух непересекающихся подмножеств. Результатом работы алгоритма распознавания является определенное решающее правило, относящее объект к одному из классов, и сама классификация объекта. [5].

1.2 Анализ существующих систем, направленных на анализ сейсмического процесса

Существует несколько программ, позволяющих провести отображение и анализ сейсмических процессов, происходящий в земной коре. В результате поиска были найдены следующие программы:

1. Earthquake Hazards Program (EHP) – база данных землетрясений США [6];
2. БАЗА ДАННЫХ «ЗЕМЛЕТРЯСЕНИЯ РОССИИ» предназначена для внесения, хранения, выборки и отображения данных о землетрясениях на территории России [7].

Данные программные продукты закрытие и предоставляют доступ в формате «Только для чтения» и минимальный набор функций, например, фильтрация в виде веб-приложения. Они похожи и имеют схожую концепцию, так как они являются базами данных эпицентров землетрясений. Рассмотрим их подробнее.

1.2.1 Earthquake Hazards Program

Программа Геологической службы США (USGS) по снижению опасности землетрясений является частью Национальной программы снижения опасности землетрясений (NEHRP), возглавляемой Национальным институтом стандартов и технологий (NIST) [6].

Система имеет множество фильтров и условий для поиска и отображения землетрясений. Начиная от указания магнитуды и временного периода, заканчивая выбором типа инцидента. На рисунках 2-4 продемонстрированы критерии фильтрации [6].

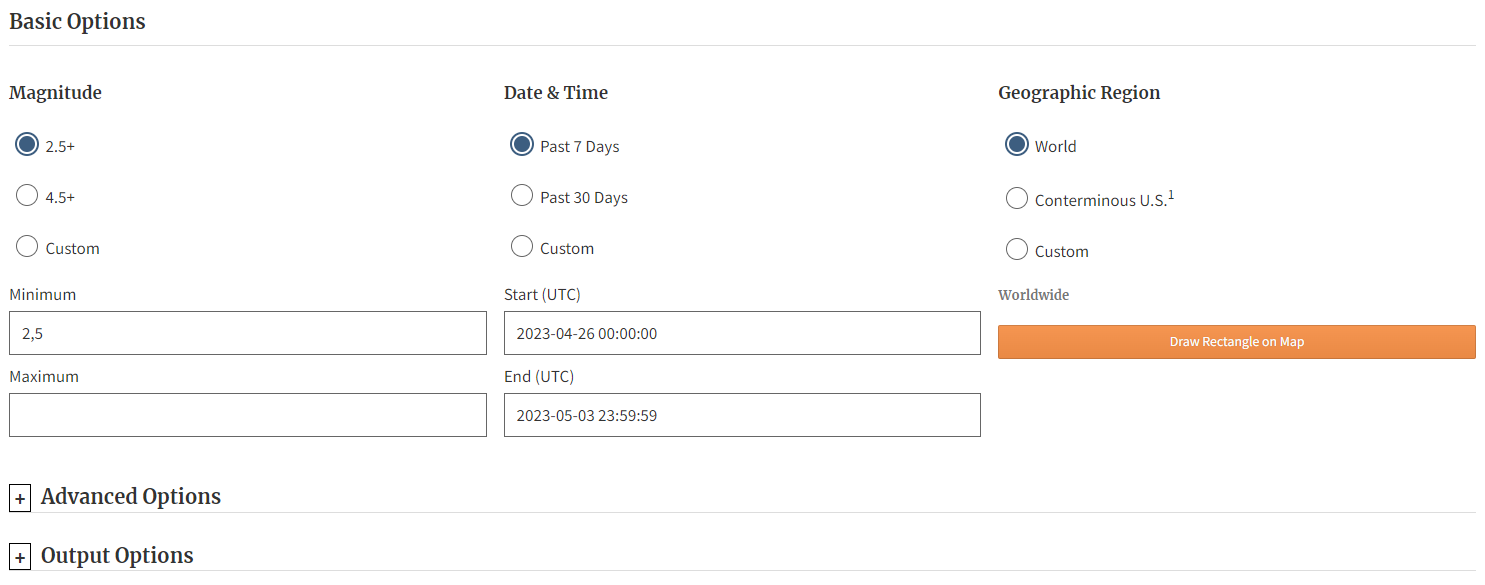


Рисунок 2 – Базовый фильтр EHP

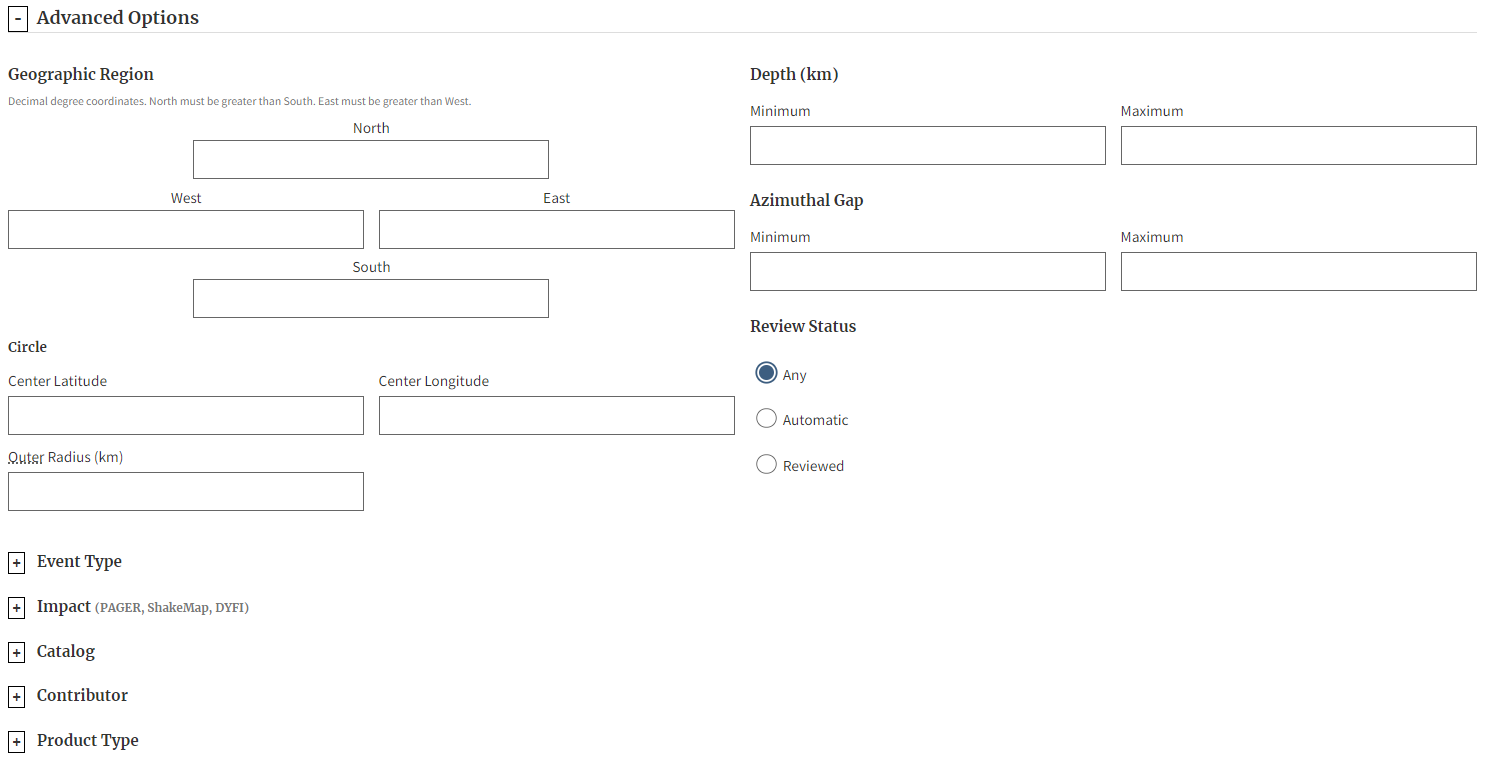


Рисунок 3 – Расширенный фильтр EHP

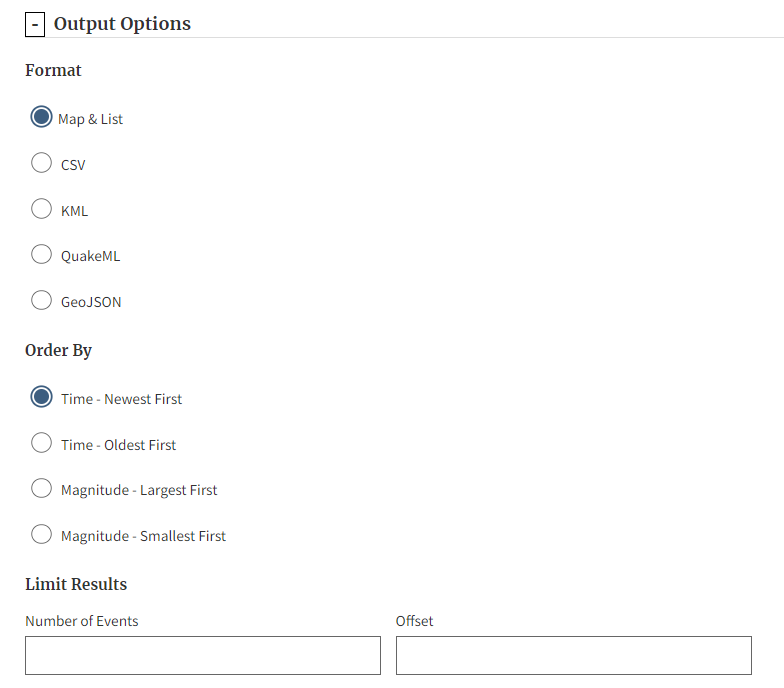


Рисунок 4 – Параметры вывода EHP

После указания всех параметров, найденные эпицентры накладываются на карту. Карта поддерживает несколько слоев для более детального отображения. Имеется возможность смены карты и отображения прочих данных, например, разломов. Эпицентры отличаются по размеру и центры. Эти параметры рассчитываются исходя от даты и их магнитуды. При нажатии на эпицентр можно ознакомиться с информацией: дата и время происшествия, место происшествия, координаты и глубина. На рисунке 5 изображена карта с результатом поиска и легендой. Имеется возможность выгрузки найденных результатов в нескольких форматах (рис. 6).

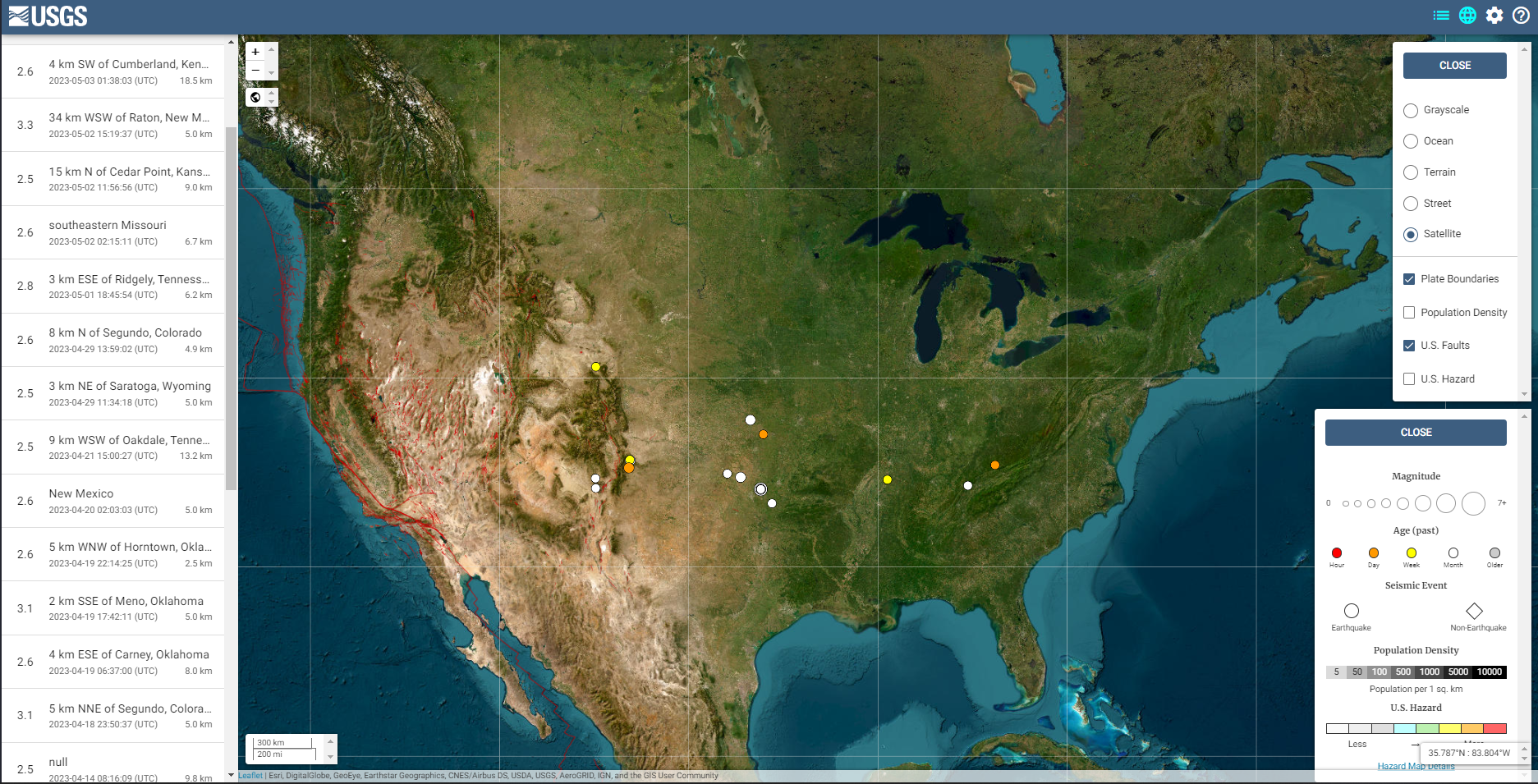


Рисунок 5 – Результаты поиска землетрясений

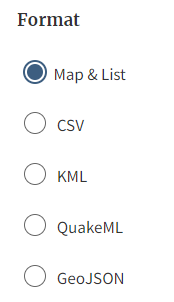


Рисунок 6 – Форматы результатов поиска

1.2.2 БАЗА ДАННЫХ «ЗЕМЛЕТРЯСЕНИЯ РОССИИ»

База данных «Землетрясения России» содержит информацию о землетрясениях в виде каталогов разных сейсмоактивных регионах России. Каталоги содержат множество параметров эпицентров. Основная энергетическая оценка – магнитуда М [7]. Веб-приложение имеет систему поиска с выбор регионов (рис. 7). На данный момент, доступно 11 регионов. Имеется возможность настройки слоев карты. К сожалению, карта не интерактивная и представлена статичной картинкой, куда наноситься данные.

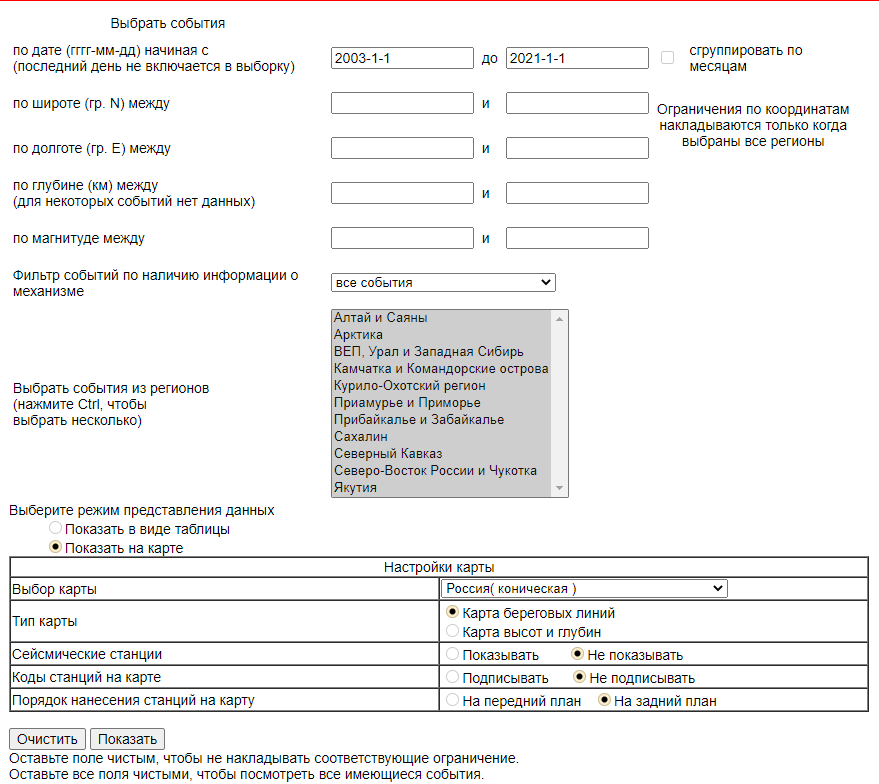


Рисунок 7 – Фильтры базы ДАННЫХ «ЗЕМЛЕТРЯСЕНИЯ РОССИИ»

После указания параметров, происходит генерация карты. После генерации карты, пользователь может изменять накладываемые слои (рис. 8). Внизу карты находится легенда. На рисунке 9 демонстрируется сгенерированная карта. Имеется возможность отобразить эпицентры в виде таблицы с сохранить их в CSV формате (рис. 10).

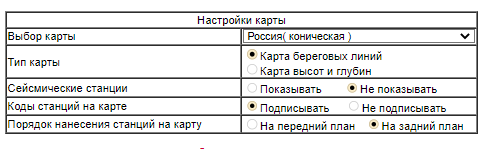


Рисунок 8 – Настройка карты



Рисунок 9 – Сканированная карта

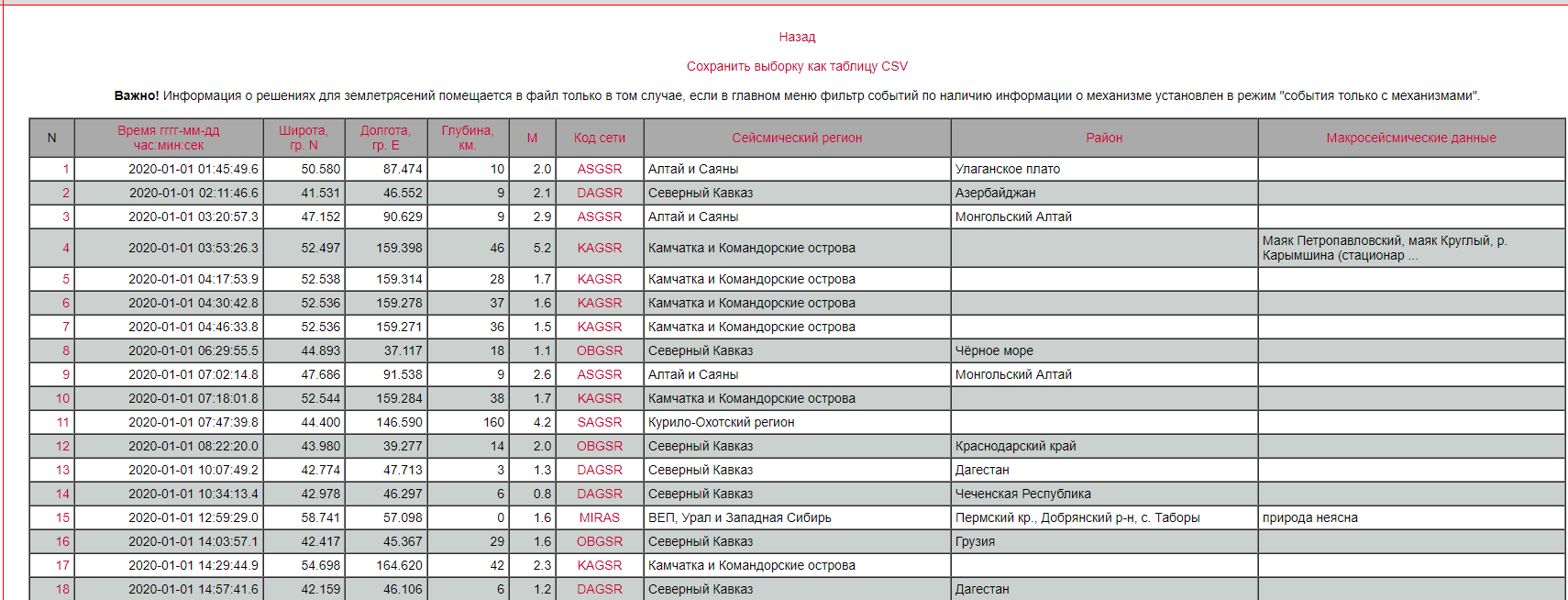


Рисунок 10 – Табличное представление эпицентров

1.3 Описание программного пакета

Программный пакет "КОМПЬЮТЕРНЫЙ АНАЛИЗ СЕЙСМИЧЕСКОГО ПРОЦЕССА" предназначен для графической визуализации развития сейсмического процесса во времени и пространстве. Обработанная информация о землетрясениях может быть представлена в виде различного рода графиков и карт параметров.

Программный пакет ориентирован на специалистов в области сейсмологии и сейсмогеологии, которые занимаются проблемами изучения сейсмических режимов, вопросами обоснования сейсмической опасности конкретных районов и прогноза сильных землетрясений. Программный пакет может быть использован для исследования любого региона земного шара при наличии каталога эпицентров землетрясений.

В текущем состоянии, КАСП способен работать только с текстовыми файлами, хранящими информацию об эпицентрах и расположению всех объектов на карте. Файлы с эпицентрами имеют расширение EQC и хранят координаты, дату и энергетический класс. Топографические данные имеют расширение TOP и описывают 7 различных объектов:

1. Неизменяемые топографические данные, которые на карте будут присутствовать всегда;
2. Озера и острова;
3. Впадины;
4. Геологическое строение;
5. Тектонические разломы, которые могут отображаться линиями разного типа и толщины;
6. Города и населенные пункты в реальном масштабе с их названиями;
7. Государственные и иные границы.

Интерфейс приложения представлен картой и нижней панелью, содержащей кнопки для управления приложением. Приложение состоит из 4 модулей. Первый модуль «Район исследования» необходим для взаимодействия с картой. Карта представлена набором примитивов, описывающим разные объекты, например, реки и озера (рис. 11).

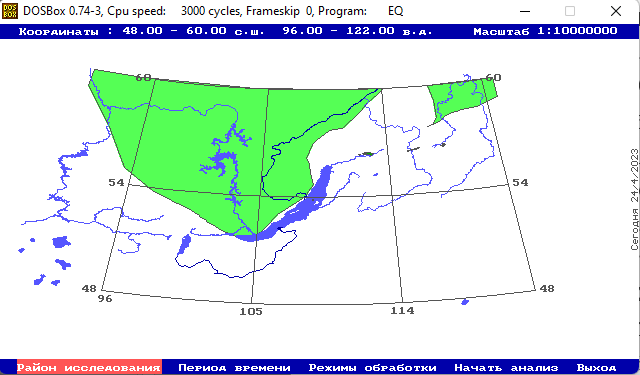


Рисунок 11 – Карта, с отображением рек и озер.

Второй модуль «Период времени» используется для выбора периода времени. КАСП использует два периода времени: доинструментальный и инструментальный. Доинструментальный период начинается с 1 января 1725 года и заканчивается 31 декабря 1949 года (рис. 12). Землетрясения, произошедшие в данное время, имеют низкую точность измерений, так как оценка производилась по косвенным признакам. Инструментальный период начинается с 1 января 1950 года (рис. 13). С этого момента, эпицентры имеют более точную информацию.

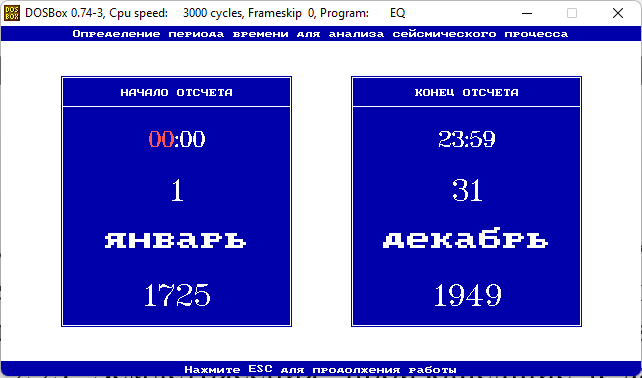


Рисунок 12 – Опция «Доинструментальный период»

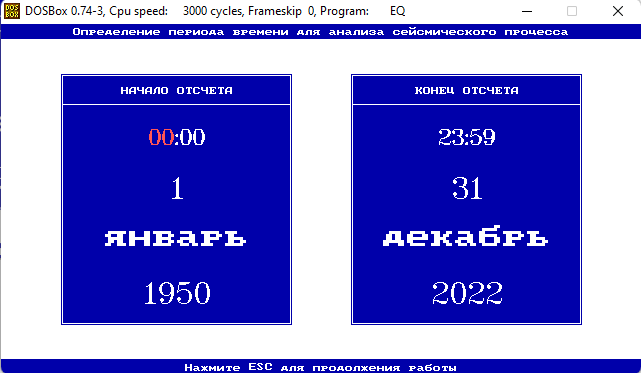


Рисунок 13 – Опция «Инструментальный период»

Третий модуль «Режимы обработки» необходим для указания режимов обработки. Выбрав опцию «Эмпирические графики», пользователь может изменить число временных интервалов, по которым будет производиться расчет графиков (рис. 14). Опция меню «Карты параметров» отвечает за выбор размера элементарного квадрата для расчета карт параметров. На основном экране изображается градусная сетка, разделенная на квадраты. В центре экрана выводятся текущие значения количества квадратов по широте и долготе. На информационной линии указываются размеры квадрата в долях градуса и в километрах, а также его площадь (рис. 15). Для выбора энергетических классов при расчете необходимо выбрать опцию меню «Энергетические классы» (рис. 16).

По умолчанию принимаются следующие режимы обработки:

1. При расчете эмпирических графиков число временных интервалов, разделяющих выбранный период исследования, принимается равным значению, выбранному в программе настройки;
2. Размеры элементарного квадрата для расчета карт параметров автоматически выбирается таким, чтобы длина его стороны составляла 16-17 км;
3. Во всех расчетах учитываются землетрясения с энергетическим классом E ≥ 8.

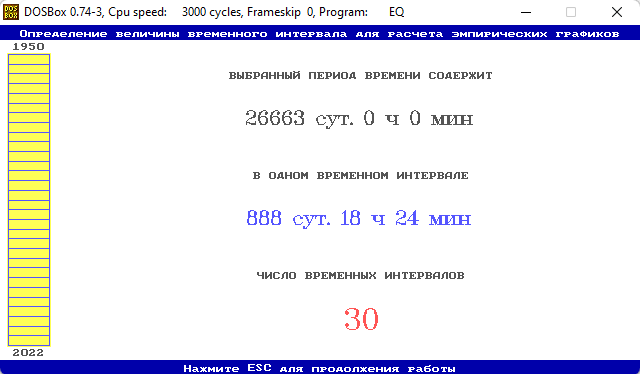


Рисунок 14 – Опция «Эмпирические графики»

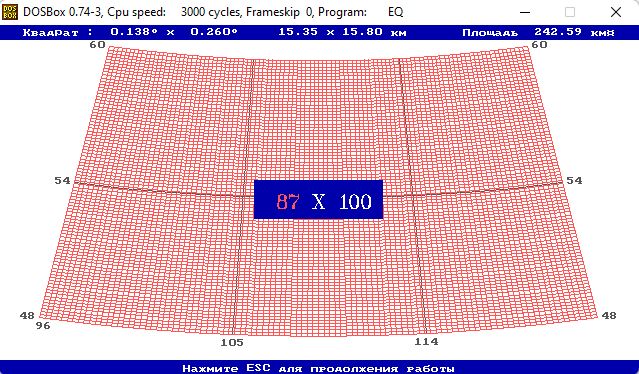


Рисунок 15 – Опция «Карты параметров»

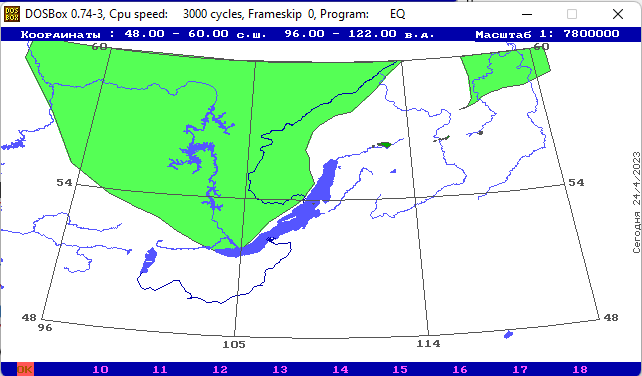


Рисунок 16 – Опция «Энергетические классы»

Четвертый модуль «Начать анализ» необходим для проведения расчетов, построения графиков и нанесения эпицентров на карту. При проекции эпицентров, приложение учитывает энергетический класс (рис. 17). Приложение способно генерировать 2 типа графиков: простые и эмпирические величины. С помощью подменю "Простые величины" можно построить графики, описывающие изменение во времени суммарного количества землетрясений и выделившейся сейсмической энергии в исследуемом районе. Графики эмпирических параметров представлены графиками изменения концентрационных критерия разрушения от времени и графиком повторяемости землетрясений. На рисунке 18 изображен график «Количество» за 1989 год.

КАСП может строить карты параметров. Карты параметров — это такие карты, градусная сетка которых разбита на элементарные квадраты, в которых рассчитываются значения эмпирических параметров. Квадраты закрашиваются определенным цветом в зависимости от значения рассчитанного параметра.

Модуль может генерировать отчеты по отображенным эпицентрам (рис. 19) и графикам (рис. 20 и 21).

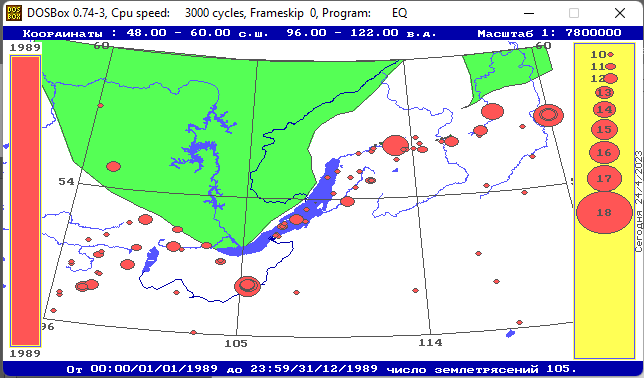


Рисунок .17 – Отображенные эпицентры

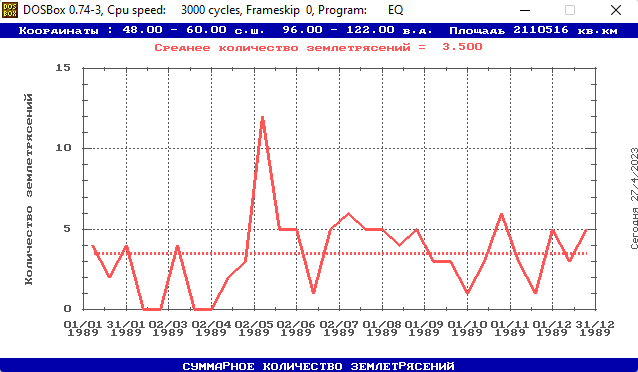


Рисунок .18 – График «Количество» за 1989 год

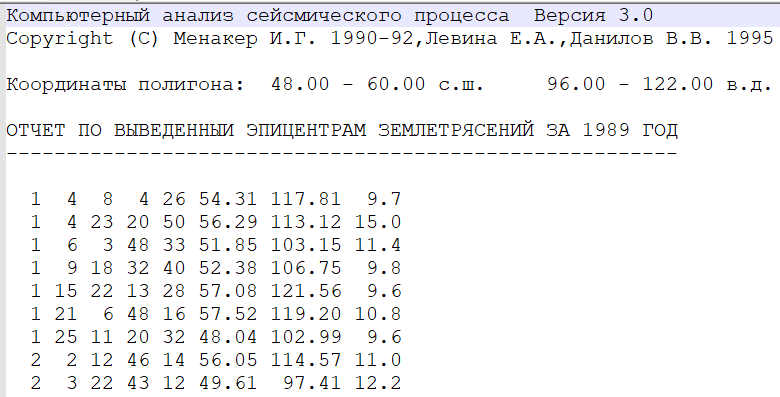


Рисунок 19 – Фрагмент отчета по эпицентрам землетрясений

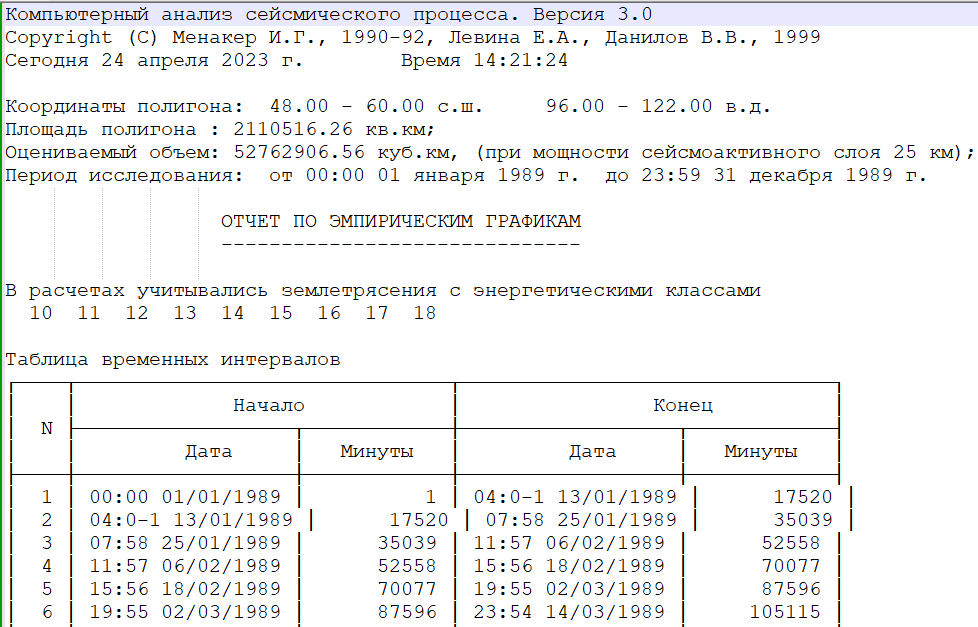


Рисунок 20 – Фрагмент отчета по эмпирическим графикам

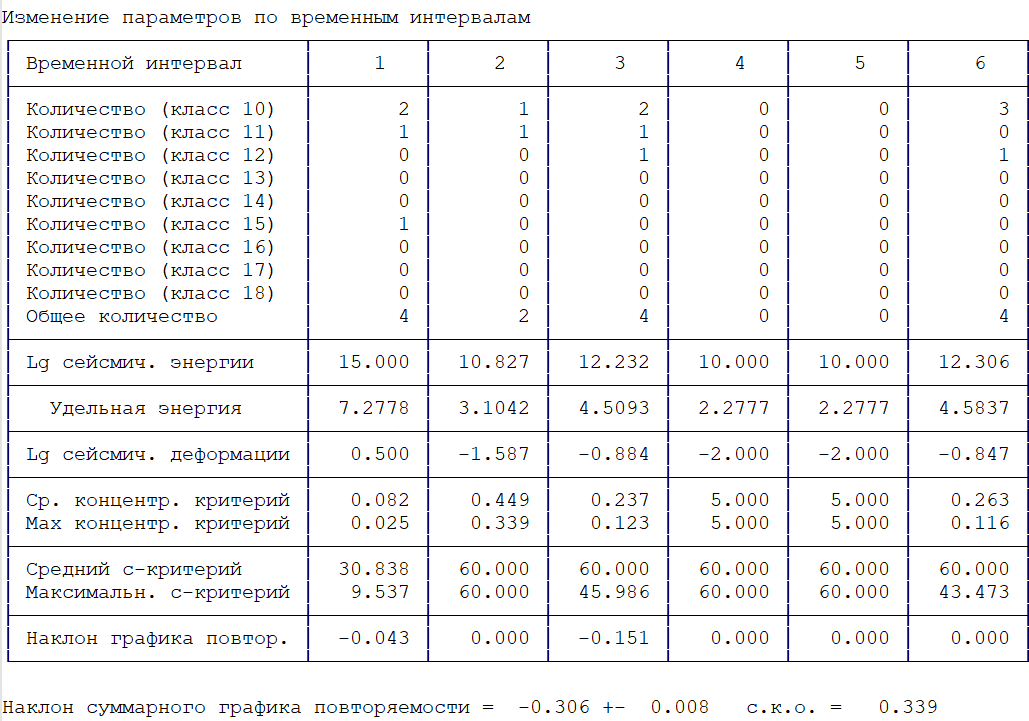


Рисунок 21 – Фрагмент отчета по эмпирическим графикам

1.4 Подходы к реинжинирингу

На сегодняшний день существует 3 способа РИС [8]:

* + 1. Создание новой ИС на основе существующей;
    2. Модификация ИС;
    3. Адаптация готовой ИС от стороннего разработчика.

В данной работе, авторами было принято решение использовать первый подход, так как модификация существующей системы не представляется возможной. Текст исходной программы написан с использованием языка программирования Pascal. КАСП в его текущем состоянии не способен быть запущен на новых системах без использования виртуальных машин со старыми версиями операционных систем. Адаптация ПО от стороннего разработчик является нецелесообразной так, как заказчики на протяжении практически 30 лет работают с существующей системой. Создание системы, основываясь на существующей является лучшим решением, потому что это позволит перенести привычное и доказавшее свою эффективность систему на современные системы, что позволит увеличить скорость работы и его функциональные возможности.

Выбранный подход имеет ряд преимуществ [8]:

* + 1. Создание нового приложение позволит отказаться от устаревших технологий, концепций и решений;
    2. Создание новой информационной системы проще прогнозировать и рассчитывать затраты.

Из недостатков можно отметить большие временные затраты по времени и ресурсам, существует вероятность, что часть задач будет решаться заново.

2. Выбор технологий и средств разработки

При проектировании информационной системы в качестве целевой платформы было принято решение использовать .Net framework. Выбор данного фреймворка обусловлен следующими факторами [9]:

* + 1. C# – это современный, инновационный, кроссплатформенный объектно-ориентированный язык программирования с открытым исходным кодом и один из 5 лучших языков программирования на GitHub;
    2. .Net framework прост в освоении, но с его помощью можно создавать приложения любой сложности. Имеет множество библиотек для работы с географическими данными.

ScootPlot – это бесплатная библиотека с открытым исходным кодом для .NET, которая упрощает интерактивное отображение больших наборов данных [10].

Gmap.Net - это отличный элемент управления с открытым исходным кодом, мощный, бесплатный и кроссплатформенный элемент управления .NET. Позволяет использовать маршрутизацию, геокодирование, направления и карты от Google, Yahoo!, Bing, OpenStreetMap, ArcGIS, Pergo, SigPac, Yendux, Mapy.cz, Maps.lt, iKarte.lv, NearMap, HereMap, CloudMade, WikiMapia, MapQuest и многие другие [11].

Npgsql – это провайдер данных ADO.NET с открытым исходным кодом для PostgreSQL, позволяющий программам, написанным на C#, получать доступ к серверу баз данных PostgreSQL. Он реализован на C#, является бесплатным и имеет открытый исходный код [12].

В качестве СУБД используется PostgreSQL. PostgreSQL - это мощная объектно-реляционная система баз данных с открытым исходным кодом, которая использует и расширяет язык SQL в сочетании со многими функциями, позволяющими безопасно хранить и масштабировать самые сложные рабочие нагрузки с данными [13].

Для работы с географическими данными используется расширение для баз данных – PostGis. PostGIS добавляет поддержку географических объектов в реляционную базу данных [14].

Draw.io – бесплатное кроссплатформенное программное обеспечение для рисования графиков с открытым исходным кодом [15].

StarUml – это программный инструмент визуального моделирования с открытым исходным кодом, который поддерживает стандартизованный язык графического описания UML (Unified Modeling Language) для моделирования систем и программного обеспечения [16].

AllFusion Process Modeler BPwin — это популярная программа для моделирования и анализа бизнес-процессов в нотациях IDEF и DFD, ставшая своего рода классикой для бизнес-аналитиков [17].

Figma – графический редактор для создания прототипов сайтов и приложений [].

3. Описание текущих возможностей реализованного программного обеспечения

3.1 Реализация функции «Указать параметры приложения»

Данное окно предназначено для глобальной настройки приложения (рис. 22). Реализуется с помощью класса AppSetting.

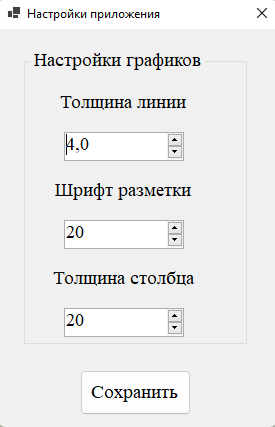


Рисунок 22 – Указать параметры приложения

**Часть листинга класса AppSetting для работы функции**

private void Init()

{

if (!iniFile.KeyExists("LineWegth", "GraghSerring"))

{

iniFile.Write("GraghSerring", "LineWegth", $"{nudGraghLineWigth.Minimum}");

}

if (!iniFile.KeyExists("TickFont", "GraghSerring"))

{

iniFile.Write("GraghSerring", "TickFont", $"{nudGraghFontTick.Minimum}");

}

if (!iniFile.KeyExists("BarWegth", "GraghSerring"))

{

iniFile.Write("GraghSerring", "BarWegth", $"{nudGraghBarWegth.Minimum}");

}

var lineWight = Convert.ToDouble(iniFile.Read("GraghSerring", "LineWegth"));

var tickFont = Convert.ToInt32(iniFile.Read("GraghSerring", "TickFont"));

var barWegth = Convert.ToInt32(iniFile.Read("GraghSerring", "BarWegth"));

nudGraghFontTick.Value = tickFont;

nudGraghLineWigth.Value = (decimal)lineWight;

nudGraghBarWegth.Value = barWegth;

}

private void btnSave\_Click(object sender, EventArgs e)

{

var lineWight = (double)nudGraghLineWigth.Value;

var tickFont = (int)nudGraghFontTick.Value;

var barWegth = (int)nudGraghBarWegth.Value;

iniFile.Update("GraghSerring", "LineWegth", $"{lineWight}");

iniFile.Update("GraghSerring", "TickFont", $"{tickFont}");

iniFile.Update("GraghSerring", "BarWegth", $"{barWegth}");

this.Close();

}

3.2 Реализация функции «Указать энергетические классы»

Окно используется для выбора энергетических классов при фильтрации эпицентров (рис. 23). Реализуется с помощью класса ClassesForm.

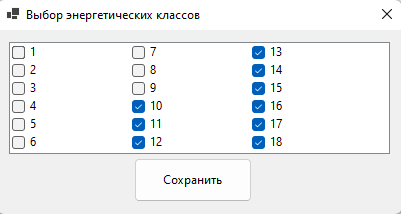


Рисунок 23 – Указать энергетические классы

**Часть листинга класса ClassesForm для работы функции**

private void Init(List<int> classes)

{

clbClasses.SelectedItems.Clear();

for (int i = 0; i < clbClasses.Items.Count; i++)

{

for (var j = 0; j < classes.Count; j++)

{

var name = clbClasses.Items[i].ToString();

if (Convert.ToInt32(name) == classes[j])

{

clbClasses.SetItemChecked(i, true);

}

}

}

}

private void btnSave\_Click(object sender, EventArgs e)

{

if(clbClasses.CheckedItems.Count < 1)

{

MessageBoxPattetn.MessageBoxErrorWithoutChoice("Вы не выбрали энергетические классы");

return;

}

MainForm main = this.Owner as MainForm;

List<int> classes = new();

var checks = clbClasses.CheckedItems;

foreach (var check in checks)

{

classes.Add(Convert.ToInt32(check.ToString()));

}

main.Filter.Classes = classes;

Close();

}

3.3 Реализация функции «Указать временной промежуток»

Форма необходима для указания периода времени при поиске эпицентров (рис. 24). Реализуется с помощью класса TimeForm.

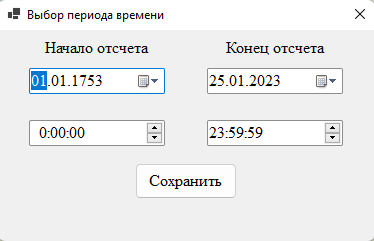


Рисунок 24 – Указать временной промежуток

**Часть листинга класса TimeForm для работы функции**

private void btnSave\_Click(object sender, EventArgs e)

{

var from = new DateTime(fromDate.Value.Year, fromDate.Value.Month,

fromDate.Value.Day, fromTime.Value.Hour,

fromTime.Value.Minute, fromTime.Value.Second);

var to = new DateTime(toDate.Value.Year, toDate.Value.Month,

toDate.Value.Day, toTime.Value.Hour,

toTime.Value.Minute, toTime.Value.Second);

MainForm? main = this.Owner as MainForm;

if (main != null)

{

main.Filter.DateTime.From = from;

main.Filter.DateTime.To = to;

main.Filter.DateTime.TimeInterval = to.Subtract(from) / main.Filter.DateTime.IntervalCount;

this.Close();

}

}

private void toDate\_ValueChanged(object sender, EventArgs e)

{

if (isLoad)

{

return;

}

if (toDate.Value < DateTime.Now)

{

toTime.MaxDate = new DateTime(1999, 12, 31, 23, 59, 59);

}

fromDate.MaxDate = toDate.Value.AddDays(-1);

}

/// <summary>

/// Изменение минимаьной даты

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void fromDate\_ValueChanged(object sender, EventArgs e)

{

if (isLoad)

{

return;

}

toDate.MinDate = fromDate.Value.AddDays(1);

}

3.4 Реализация функции «Указать число временных интервалов»

Окно используется для указания числа временных интервалов при построении графиков (рис. 25). Реализуется с помощью класса EmpiricalGraphsForm.

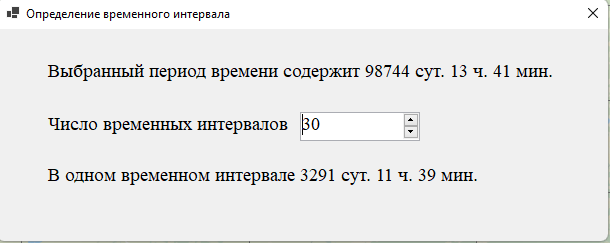


Рисунок 25 – Указать число временных интервалов

**Часть листинга класса EmpiricalGraphsForm для работы функции**

private void EmpiricalGraphsForm\_Load(object sender, EventArgs e)

{

var owner = this.Owner as MainForm;

var firstDate = owner.Filter.DateTime.From;

var secondDate = owner.Filter.DateTime.To;

differ = secondDate.Subtract(firstDate);

lblTotalDate.Text = $"Выбранный период времени содержит {differ.Days} сут. {differ.Hours} ч. {differ.Minutes} мин.";

var interval = differ / owner.Filter.DateTime.IntervalCount;

lblSelected.Text = $"В одном временном интервале {interval.Days} сут. {interval.Hours} ч. {interval.Minutes} мин.";

nudSelect.Value = owner.Filter.DateTime.IntervalCount;

}

/// <summary>

/// Метод, вызываемый при изменении значения в nudSelect

/// </summary>

/// <param name="sender"></param>

/// <param name="e"></param>

private void nudSelect\_ValueChanged(object sender, EventArgs e)

{

var owner = this.Owner as MainForm;

owner.Filter.DateTime.TimeInterval = differ / (int)nudSelect.Value;

owner.Filter.DateTime.IntervalCount = (int)nudSelect.Value;

var interval = differ / (int)nudSelect.Value;

lblSelected.Text = $"В одном временном интервале {interval.Days} сут. {interval.Hours} ч. {interval.Minutes} мин.";

}

3.5 Реализация функции «Выбор района исследования»

Функция используется для выбора района исследования с заданными параметрами (рис. 26). Реализуется с помощью класса SelectArea.

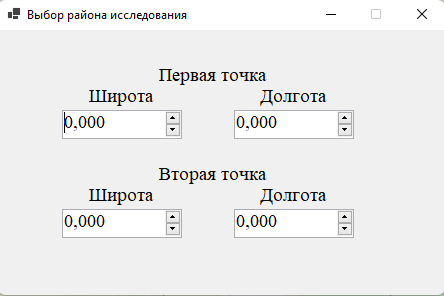


Рисунок 26 – Выбор района исследования

**Часть листинга класса SelectArea для работы функции**

private void nudValueChange(object sender, EventArgs e)

{

var main = this.Owner as MainForm;

main.LayOfUserClick.Markers.Clear();

main.LayOfUserClick.Polygons.Clear();

var lng1 = (double)nudLngFirst.Value;

var lon1 = (double)nudLonFirst.Value;

var lng2 = (double)nudLngSecond.Value;

var lon2 = (double)nudLonSecond.Value;

List<PointLatLng> pointLatLngs = new();

pointLatLngs.Add(new(lng1, lon1));

pointLatLngs.Add(new(lng1, lon2));

pointLatLngs.Add(new(lng2, lon2));

pointLatLngs.Add(new(lng2, lon1));

foreach (var point in pointLatLngs)

{

GMarkerGoogle marker = new(point, GMarkerGoogleType.black\_small);

marker.ToolTipText = $"Широта: {string.Format("{0: .00}", point.Lat)}\nДолгота {string.Format("{0: .00}", point.Lng)}";

main.LayOfUserClick.Markers.Add(marker);

}

GMapPolygon polygon = new(pointLatLngs, "area");

Pen pen = new(Color.Black, 1);

polygon.Stroke = pen;

polygon.Fill = new SolidBrush(Color.FromArgb(15, Color.DeepPink));

main.LayOfUserClick.Polygons.Add(polygon);

main.Filter.District.Polygon = polygon;

}

3.6 Реализация функций «Отобразить топографию» и «Отобразить эпицентры»

Главное окно системы является ключевым, потому что с его помощью происходит вызов всех прочих функций. Также при помощи этой формы происходит визуализация данных. На карту по умолчанию нанесена картографическая сетка, которая изменяет свой масштаб в зависимости от зуммирования карты. Реализация представлена на рисунке 64. Реализуется с помощью класса MainForm.

С помощью функции отобразить топографию, можно вывести на карту топографию. При использовании функций выбор энергетических классов, определение периода времени, выбор района исследования и начать анализ пользователь может визуализировать данные об эпицентрах с прилегающей к ней информацией. Эпицентры различаются по цвету и размеру в зависимости от их энергетического класса. Пример визуализации данных представлен на рисунке 65.

При сравнении рисунков 27 и 28 видно, что панель с информацией меняться зависимости от параметров, веденных пользователем. Например, изменились выведенные энергетические классы и вывелась площадь района исследования.



Рисунок 27 – Главное окно системы

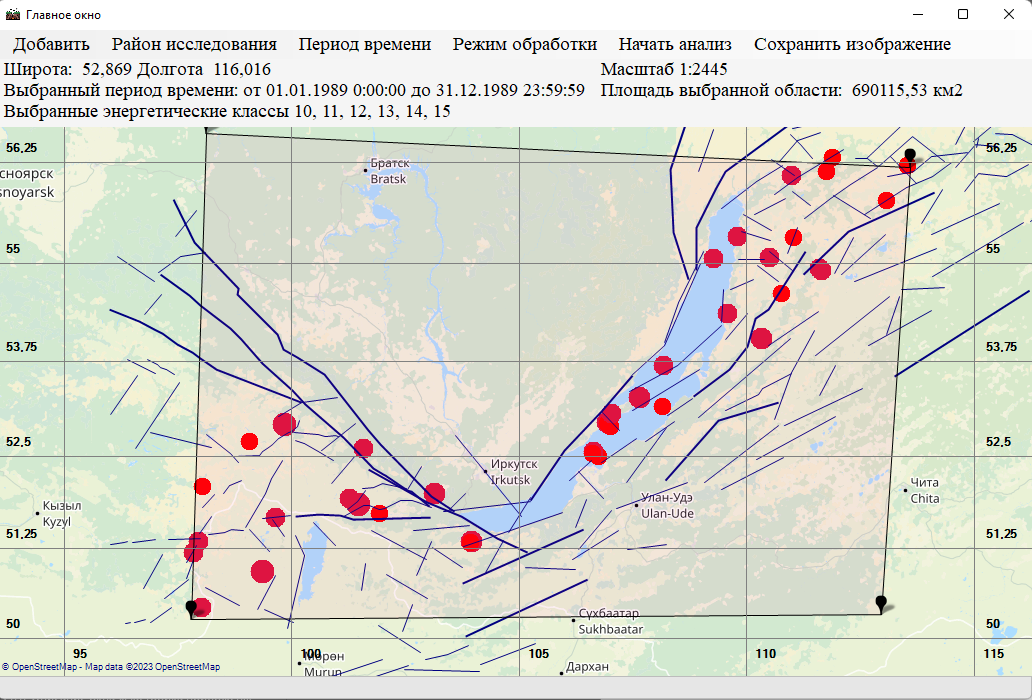


Рисунок 28 – Визуализация данных

**Часть листинга класса MainForm для работы функции**

private async void FaultShowClick(object sender, EventArgs e)

{

//Вкладка

var item = (sender as ToolStripMenuItem);

//Если она уже выбрана

if (item.Checked)

{

//Очистить слой

LayOfFaults.Clear();

LayOfFaults.Dispose();

//Собрать мусор

GC.Collect();

GC.WaitForPendingFinalizers();

//Убрать галку

item.Checked = !item.Checked;

return;

}

if (IsOnAction)

{

MessageBoxPattetn.MessageBoxInformation("Идет выполнение другого действия");

return;

}

IsOnAction = true;

//Если регион не выбран

//Получить все регионы

MapData.Faults = await db.GetRiftAsync();

//Преобразовать регионы в массив

var faults = (from f in MapData.Faults.Values

select f).ToArray();

pbAdd.Maximum = faults.Length;

var progress = new Progress<int>(value =>

{

pbAdd.Value = value;

});

//Нарисовать регионы

var count = await Task.Run(() => ShowFaults(faults, progress));

MessageBoxPattetn.MessageBoxInformation($"Отражено {WordsCorrect.CorrectFaultEnd(count)}");

pbAdd.Value = 0;

item.Checked = true;

GC.Collect();

GC.WaitForPendingFinalizers();

IsOnAction = false;

}

3.7 Реализация функции «Сохранить карту»

Система поддерживает возможность сохранения карты в формате .jpg и .png в любой момент времени со всей нанесенной информацией на ней (рис. 29). Реализуется с помощью класса MainForm.

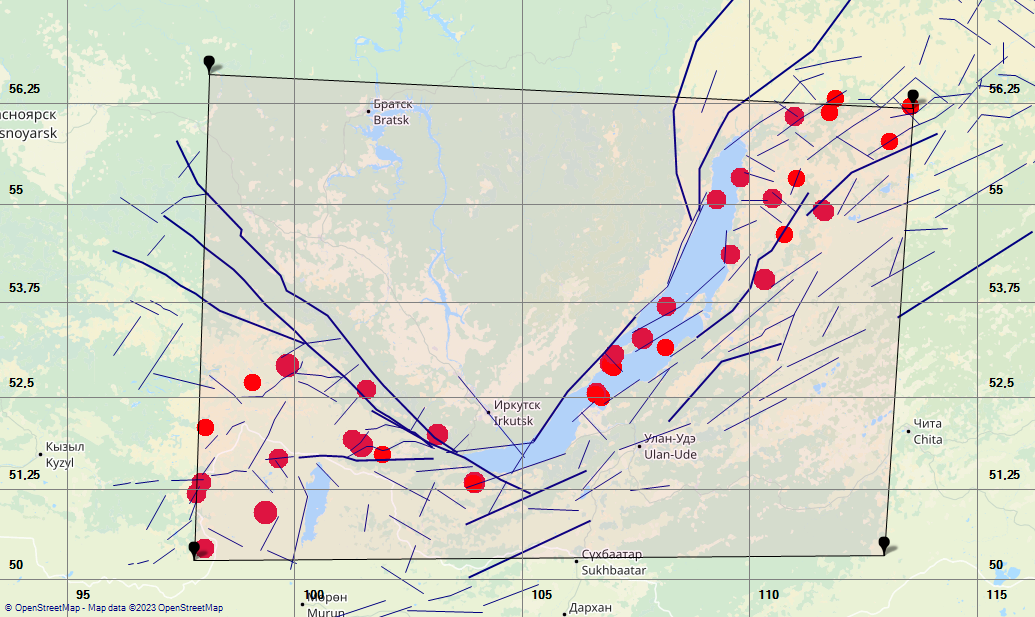


Рисунок 29 – Сохраненная карта в формате .png

**Часть листинга класса MainForm для работы функции**

private void saveMapImg(object sender, EventArgs e)

{

saveImg.FileName = "Карта";

if (saveImg.ShowDialog() == DialogResult.Cancel)

return;

string filename = saveImg.FileName;

var img = Map.ToImage();

img.Save(filename);

}

3.8 Реализация функции «Загрузить данные об эпицентрах»

Для добавление новой информации об эпицентрах в базу данных используется функция «Загрузить данные об эпицентрах». Она поддерживает файлы формата .eqc со специальным формированием документа (рис. 30 и 31). Реализуется с помощью класса MainForm.

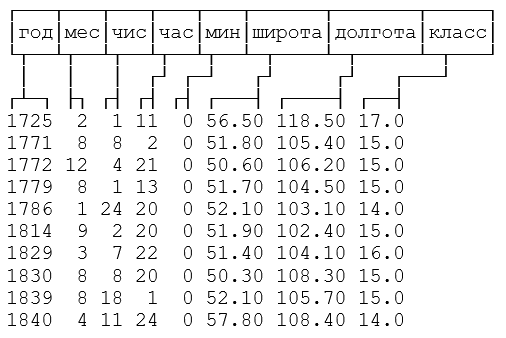


Рисунок 30 – Формат файлов .eqc

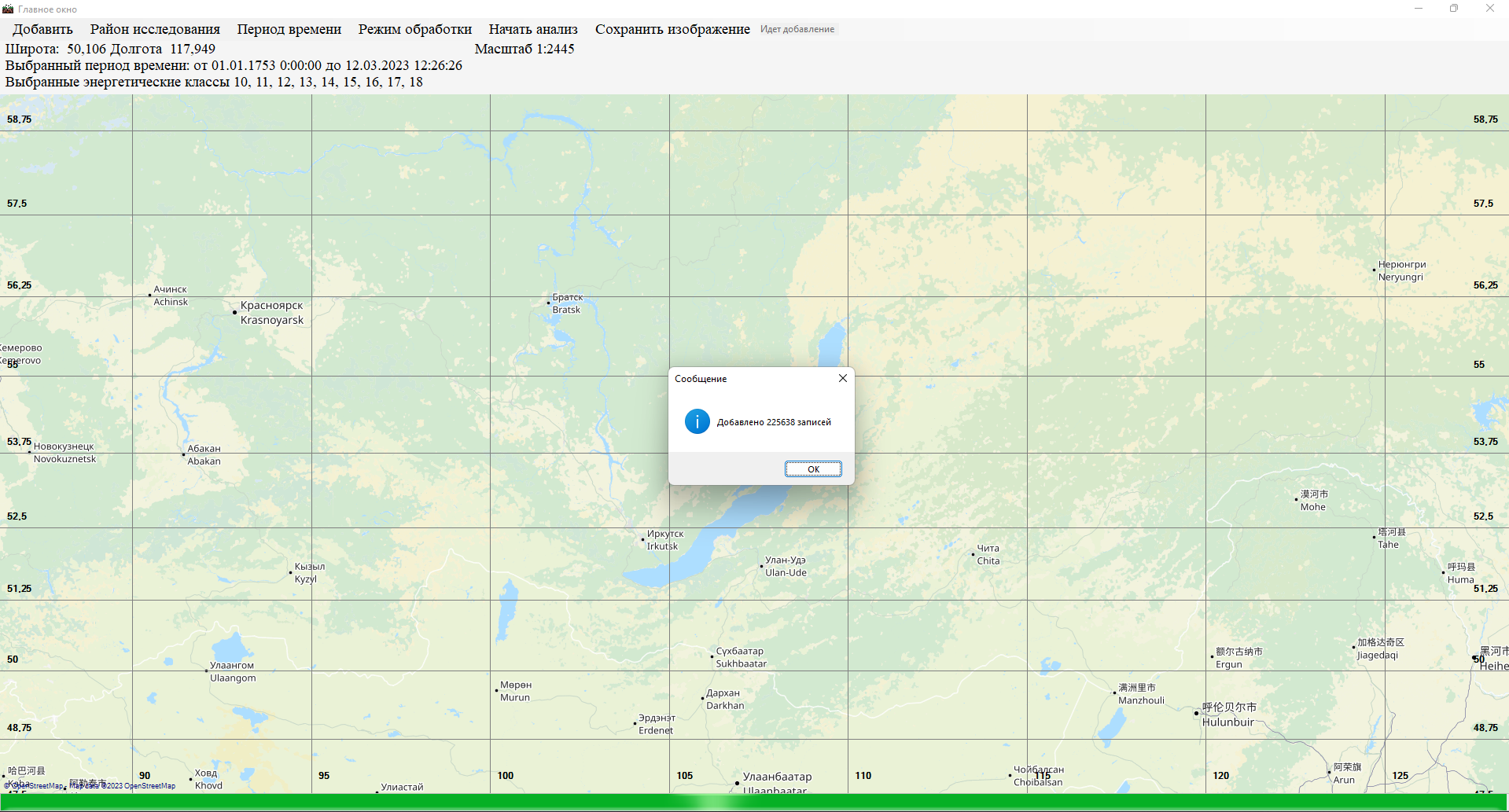


Рисунок 31 – Добавление эпицентров

**Часть листинга класса MainForm для работы функции**

private async void EpicentreFileReadClick(object sender, EventArgs e)

{

if (IsOnAction)

{

MessageBoxPattetn.MessageBoxInformation("Идет выполнение другого действия");

return;

}

openFileDialog1.Filter = "EQC files(\*.EQC)|\*.EQC";

//Если закрыть окно

if (openFileDialog1.ShowDialog() == DialogResult.Cancel) { return; }

IsOnAction = false;

//получить путь

var path = openFileDialog1.FileName;

var file = File.ReadAllLines(path);

pbAdd.Maximum = file.Length;

pbAdd.Value = 0;

toolStripTextBoxState.Text = "Идет добавление";

var progress = new Progress<int>(value =>

{

pbAdd.Value = value;

});

var count = await Task.Run(async () => await AddEpicentre(file, progress));

MessageBoxPattetn.MessageBoxInformation($"Добавлено {WordsCorrect.CorrectRecordEnd(count)}");

pbAdd.Value = 0;

toolStripTextBoxState.Clear();

GC.Collect();

GC.WaitForPendingFinalizers();

IsOnAction = false;

}

3.9 Реализация функции «Загрузить данные о разломах»

Для добавление новой информации о разломах в базу данных используется функция «Загрузить данные о разломах». Она поддерживает файлы формата .top и .txt со специальным формированием документа (рис. 32 и 33). Реализуется с помощью класса MainForm.

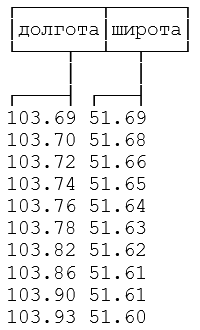


Рисунок 32 – Формат файлов разломов

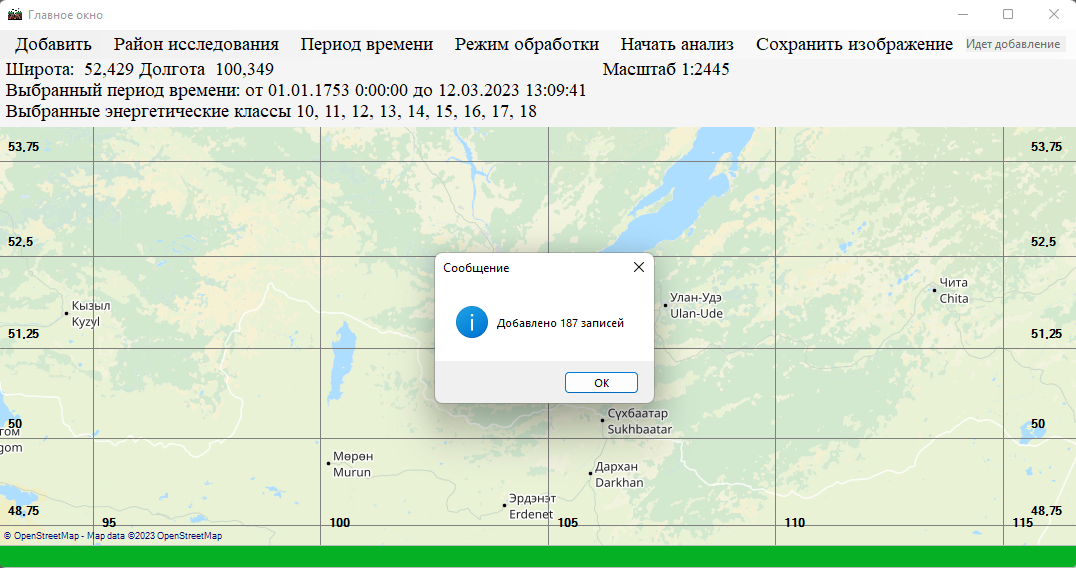


Рисунок 33 – Добавление разломов

**Часть листинга класса MainForm для работы функции**

private async void FaultFileReadClick(object sender, EventArgs e)

{

if (IsOnAction)

{

MessageBoxPattetn.MessageBoxInformation("Идет выполнение другого действия");

return;

}

//Устаносить фильтр. Только TOP файлы и Fault\*.TOP

openFileDialog1.Filter = "TXT files(\*.TXT)|\*.TXT|" +

"TOP files(\*.TOP)|\*.TOP";

//Если закрыть окно

IsOnAction = true;

if (openFileDialog1.ShowDialog() == DialogResult.Cancel)

{

IsOnAction = false;

return;

}

//получить путь

var path = openFileDialog1.FileName;

//Получить все строки

var file = File.ReadAllLines(path);

//Настройка прогресс бара

pbAdd.Maximum = file.Length;

pbAdd.Value = 0;

toolStripTextBoxState.Text = "Идет добавление";

//Изменение значение прогресс бара

var progress = new Progress<int>(value =>

{

pbAdd.Value = value;

});

var count = await Task.Run(async () => await AddFault(file, progress));

MessageBoxPattetn.MessageBoxInformation($"Добавлено {WordsCorrect.CorrectRecordEnd(count)}");

pbAdd.Value = 0;

toolStripTextBoxState.Clear();

GC.Collect();

GC.WaitForPendingFinalizers();

IsOnAction = false;

}

3.10 Реализация функции «Генерировать отчет»

После отображения эпицентров, пользователь может генерировать отчет по найденным эпицентрам в формате .xlsx (рис. 34). Отчет содержит список всех найденных эпицентров и краткую информацию. Реализуется с помощью класса MainForm.

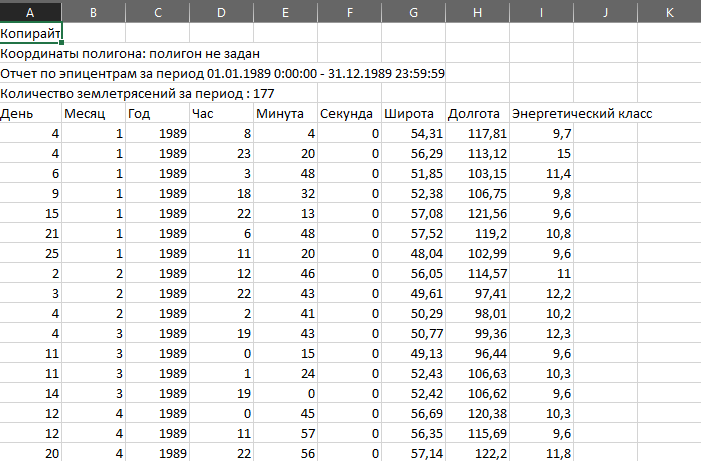


Рисунок 34 – Сгенерированный отчет

**Часть листинга класса MainForm для работы функции**

private async void CreateReportClick(object sender, EventArgs e)

{

toolStripTextBoxState.Text = "Идет генерация отчета";

var epicentres = await db.GetEpicentreAsync(Filter);

if (epicentres is null)

{

return;

}

var progress = new Progress<int>(value =>

{

pbAdd.Value = value;

});

pbAdd.Value = 0;

pbAdd.Maximum = epicentres.Count;

await Task.Run(() => CreateReport(epicentres, progress));

MessageBoxPattetn.MessageBoxInformation("Отчет успешно сгенерирован");

toolStripTextBoxState.Clear();

pbAdd.Value = 0;

}

3.11 Реализация функции «Сохранить график»

После построения графика, пользователь может его сохранить (рис. 35) в формате .png. Для реализации используется класс ChartForm.

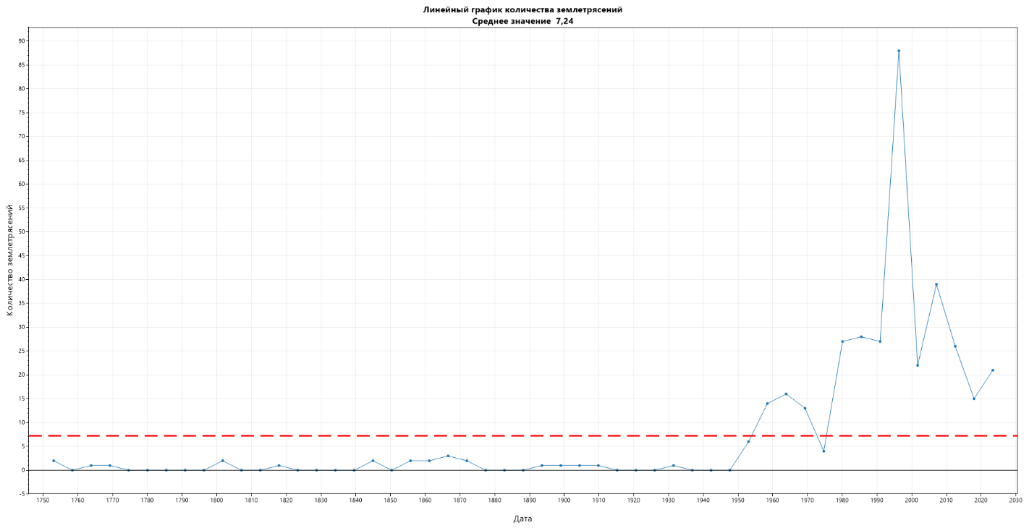


Рисунок 35 – Сохраненный график

**Часть листинга класса MainForm для работы функции**

private void SaveImgClick(object sender, EventArgs e)

{

saveFileDialog1.FileName = "График";

if (saveFileDialog1.ShowDialog() == DialogResult.Cancel)

return;

formsPlot1.Plot.SaveFig(saveFileDialog1.FileName);

}

Заключение

Преддипломная практика проходила в срок с 24.04.2023 по 21.04.23. В ходе производственной практики в институте систем энергетики им. Л.А. Мелентьева СО РАН, отдел систем искусственного интеллекта, была проанализирована предметная область, описаны используемые программные ращения, описаны текущие возможности разработанного программного комплекса и написан отчет по проделанной работе.

Список использованных источников

1. Что такое землетрясение. [Электронный ресурс] URL: <https://41.mchs.gov.ru/deyatelnost/poleznaya-informaciya/zemletryaseniya-i-cunami/chto-takoe-zemletryasenie> (дата обращения: 27.04.2023);
2. ЧТО ТАКОЕ СЕЙСМОГРАММА КРАТКО. [Электронный ресурс] URL: <https://obrazovanie-gid.ru/voprosy/chto-takoe-sejsmogramma-kratko.html> (дата обращения 27.04.2023);
3. Современные методы исследования землетрясений. [Электронный ресурс] URL: <https://studbooks.net/1742234/geografiya/sovremennye_metody_issledovaniya_zemletryaseniy#179> (дата обращения 27.04.2023);
4. Шкала Рихтера. [Электронный ресурс] URL: <https://www.meteorologiaenred.com/ru/шкала-Рихтера.html> (дата обращения 27.04.2023);
5. А. Д. Гвишиани, А. А. Соловьев, Б. А. ДзебоевПРОБЛЕМА РАСПОЗНАВАНИЯ МЕСТ ВОЗМОЖНОГО ВОЗНИКНОВЕНИЯ СИЛЬНЫХ ЗЕМЛЕТРЯСЕНИЙ: АКТУАЛЬНЫЙ ОБЗОР. // Физика Земли, 2020, № 1, [Электронный ресурс] URL: <https://sciencejournals.ru/view-article/?j=fizzemli&y=2020&v=0&n=1&a=FizZemli2001004Gvishiani> (дата обращения 27.04.2023);
6. Earthquake Hazards Program. [Электронный ресурс] URL: <https://www.usgs.gov/programs/earthquake-hazards/what-we-do-earthquake-hazards-program> (дата обращения 28.04.2023);
7. БАЗА ДАННЫХ «ЗЕМЛЕТРЯСЕНИЯ РОССИИ». [Электронный ресурс] URL: <http://eqru.gsras.ru/> (дата обращения 28.04.2023);
8. Основные пути реинжиниринга информационных систем. [Электронный ресурс] URL: <https://studfile.net/preview/423770/page:3/#423770> (дата обращения 30.04.2023);
9. Официальный сайт .Net. [Электронный ресурс] URL: <https://dotnet.microsoft.com/en-us/> (дата обращения 01.05.2023);
10. Документация библиотеки ScootPlot. [Электронный ресурс] URL: <https://scottplot.net> (дата обращения 01.05.23);
11. Страница библиотеки GMap.Net. [Электронный ресурс] URL: <https://www.nuget.org/packages/GMap.NET.Windows/> (дата обращения 02.05.2023);
12. Документация Npgsql. [Электронный ресурс] URL: <https://www.npgsql.org> (дата обращения 02.05.2023);
13. Документация PostgreSQL. [Электронный ресурс] URL: <https://www.postgresql.org/about/> (дата обращения 03.05.2023);
14. Документация PostGis. [Электронный ресурс] URL: <https://postgis.net> (дата обращения 04.05.2023);
15. Официальный сайт Draw.io. [Электронный ресурс] URL: <https://drawio-app.com/product/> (дата обращения 04.05.2023);
16. Страница продукта на площадке soware. [Электронный ресурс] URL: <https://soware.ru/products/staruml>/ (дата обращения 04.05.2023);
17. Страница продукта на площадке soware. [Электронный ресурс] URL: <https://soware.ru/products/allfusion-process-modeler-bpwin>/ (дата обращения 04.05.2023);
18. Первый шаги в дизайне: инструкция по базовым возможностям Figma. [Электронный ресурс] URL: <https://practicum.yandex.ru/blog/chto-takoe-figma-dlya-dizainera> (дата обращения 05.05.2023).