

Содержание

Введение	5
1 Обоснование необходимости выполнения проекта	7
1.1 Обзор аналогичных программных продуктов	7
1.2 Анализ необходимого функционала	10
2 Разработка приложения	11
2.1 Выбор среды разработки	11
2.2 Описание обобщённого алгоритма приложения	12
2.3 Разработка тестов	17
2.4 Руководство пользователя	19
2.5 Руководство программиста	23
2.6 Перспективы развития, использования приложения в дальнейшем	29
3 Безопасность жизнедеятельности	30
3.1 Требования к помещениям при работе за компьютером	30
3.2 Требования к микроклимату, ионному составу и концентрации вредных химических веществ в воздухе помещений	30
3.3 Требования к освещению помещений и рабочих мест	31
3.4 Требования к шуму и вибрации в помещениях	32
3.5 Требования к организации и оборудованию рабочих мест	32
3.6 Режим труда и отдыха при работе с компьютером	33
3.7 Обеспечение электробезопасности и пожарной безопасности на рабочем месте	35
4 Экономическое обоснование	38
4.1 Расчёт трудозатрат на разработку программного продукта	38
4.2 Составление сметы затрат на разработку программного продукта	38
4.3 Экономическая эффективность от использования программного продукта	42
4.4 Обобщение результатов расчётов	43
5 Определение интегрального критерия уровня готовности к информационному обществу по результатам полученным в дипломном проекте	44
Заключение	51
Библиография	52
Приложение А	54

					230105.000009П.308.ПЗ			
Изм.	Лист	№ докум.	Подп.	Дата	Разработка приложения для автоматизации обработки результатов топосъемки Свердловской городской спелеосекции Пояснительная записка	Лит.	Лист	Листов
Разраб.		Никифоров С.В.						
Провер.		Яковлев А.В.					4	62
Реценз.						УрТИСИ ФГОБУ ВПО «СибГУТИ»		
Н.контр.		Тюпина О.М.						
Утверд.		Журавлёва И.Н.						

Введение

Спелеология (от греч. spelaion – пещера и logos – слово, учение) – наука о пещерах, их происхождении и геологических условиях развития, морфологии, отложениях и полезных ископаемых, условиях обитания организмов, истории освоения и практическом использовании. Спелеология как научное направление начала оформляться во 2-й половине 19 века благодаря трудам французского учёного Э. А. Мартеля, который применил этот термин в 1900.

Спелеология рассматривает пещеры как сложные системы, существующие в ограниченном подземном пространстве, все элементы которых взаимосвязаны (подземные ландшафты). Главные объекты исследования в спелеологии обусловили выделение таких разделов, как геоспелеология, региональная спелеология, биоспелеология, антропоспелеология, палеонтоспелеология, прикладная спелеология, техническая спелеология.

Спелеология – комплексная наука. Большой вклад в её развитие вносят учёные разных научных направлений (геология и горное дело, география, биология, археология и др.) и спелеологи-спортсмены.

В России спелеология не является коммерческой наукой, её продвижение не финансируется и держится на энтузиазме участников. Несмотря на это, есть немало людей, готовых взяться за эту нелёгкую, опасную, но интересную работу.

Один из этапов изучения пещеры – это топографическая съёмка. Топосъёмка – это совокупность камеральных и полевых геодезических работ, которые выполняются на местности с целью составления топографических планов и карт. В процессе топосъёмки определяется планово-высотное расположение характерных точек местности.

Для построения системы отсчета в нашем физическом мире необходимо тело отсчета и два вектора (или направления). Третий доопределяется до правой (левой) тройки. Для двух направлений выбирается: «север-юг» и «верх-низ». На широтах отличных от экватора направления магнитных силовых линий не горизонтальны, но на небольших расстояниях (порядка километра), на которых, как правило, располагаются пещеры с точностью достаточной для спелеонужд кривизной Земли и системы координат «широта-долгота» можно пренебречь. В результате получается декартова система координат.

Измерять декартовы координаты напрямую в пещерах невозможно, также как и измерять всегда относительно одного тела отсчета. Поэтому измеряют координаты точек относительно друг друга, используя локальные тела отсчета и системы координат, а потом пересчитывают все в общую систему координат. Так как кривизной Земли в данных условиях можно пренебречь, то принимаются направления на север и вертикали во всех точках параллельными друг другу.

В спелеологии применяется измерения в двух видах координатных систем: сферической (классический вариант) и цилиндрической (применяется в

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		5

вертикально-ориентированных пещерах). В данной работе будет рассмотрена сферическая система.

Принцип топосъемки следующий: выбирается начальная точка отсчёта – как правило, это вход в пещеру, либо точка с уже известными координатами – репер, определяются опорные точки – пикеты, затем, начиная от начального пикета до всех, находящихся в прямой видимости измеряется расстояние, угол относительно горизонта и магнитный азимут. Затем такие же измерения повторяются для остальных пикетов, но уже за начальные принимаются те пикеты, которые были измерены на предыдущих шагах. Таким образом становятся известны сферические координаты текущего пикета относительно предыдущего.

Большое количество измерений неизбежно ведёт за собой накопление погрешности. Для снижения этой погрешности применяется проводится серия измерений, конечный пикет которой совпадает с начальным (замыкание хода, либо измерение хода сначала в одну сторону, затем в обратную по разным пикетам), затем все пикеты, входящие в данную серию измерений (а так же измеренные после них) смещаются на некоторое расстояние до совпадения начального и конечного.

Целью выполнения работы является разработка приложения для автоматизации обработки результатов топосъёмок, проводимых Свердловской городской спелеосекцией. Для достижения поставленной цели необходимо выполнить ряд последовательных задач:

- изучить предметную область;
- разработать алгоритм преобразования входных данных в декартовы координаты;
- разработать алгоритм исправления возникающих при измерении погрешностей;
- выбрать язык программирования и среду разработки приложения;
- выбрать технологию графического отображения и редактирования данных в графическом режиме;
- закодировать разработанный ранее алгоритм;
- разработать необходимую пользовательскую и справочную документацию;
- провести обучение членов Свердловской городской спелеосекции работе с приложением.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		6

1 Обоснование необходимости выполнения проекта

1.1 Обзор аналогичных программных продуктов

Для построения карт пещер существует не очень много программ. Большинство из них разработано американскими, французскими и немецкими спелеологами.

CAPS (CAve maPS) – продукт, созданный американским спелеологом Хьюбертом Кроувелом. Позволяет вводить и редактировать данные, обладает скромным интерфейсом и минимальным набором инструментов. На рисунке 1.1 представлен внешний вид программы.

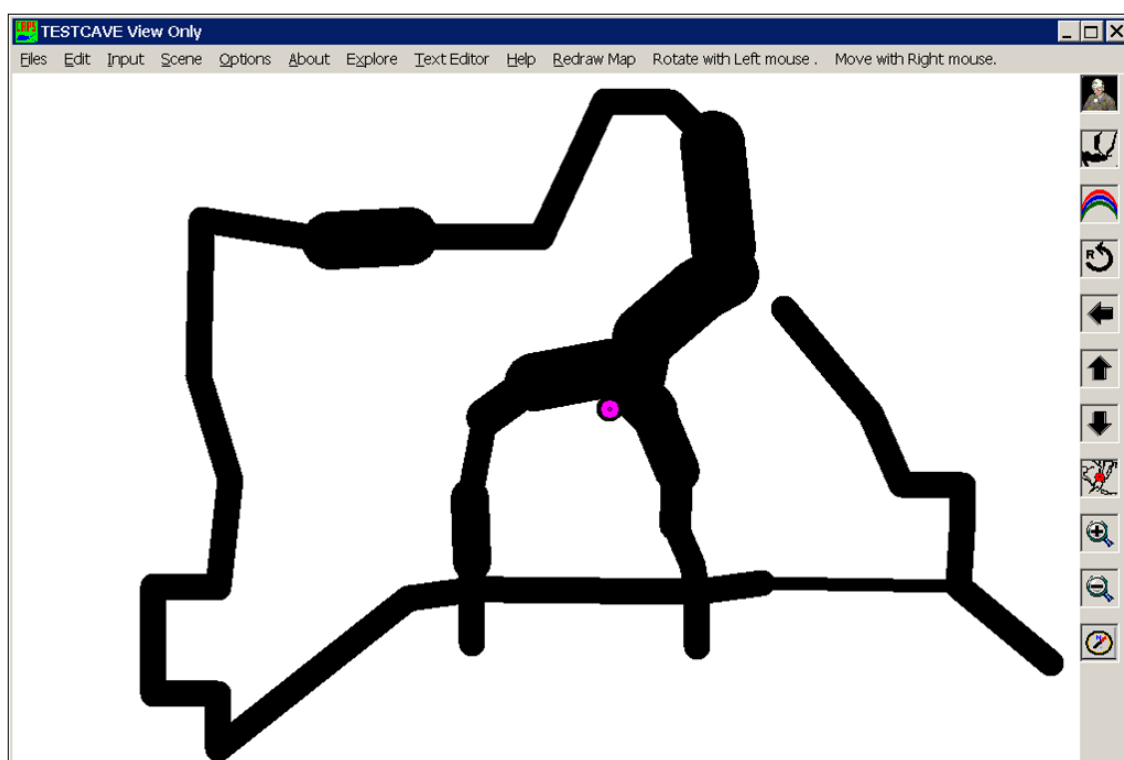


Рисунок 1.1 – Внешний вид приложения CAPS

CaveRender – мощная программа для построения карт. Позволяет вводить, редактировать данные, строить карты, выводить на печать или в файл полученные результаты, строить 3D модели пещер. Демо-версия приложения не поддерживает печати и сохранения. Внешний вид программы в режиме отображения 3D модели представлен на рисунке 1.2.

Compass – мощный и дорогой продукт для комплексной обработки данных топосъёмки. Ввод, редактирование данных, аналитическое представление

результатов обработки, нанесение карт пещеры на географические карты (поддержка Google-maps), построение 3D-модели пещеры. Программа поддерживает цветное отображения высот.

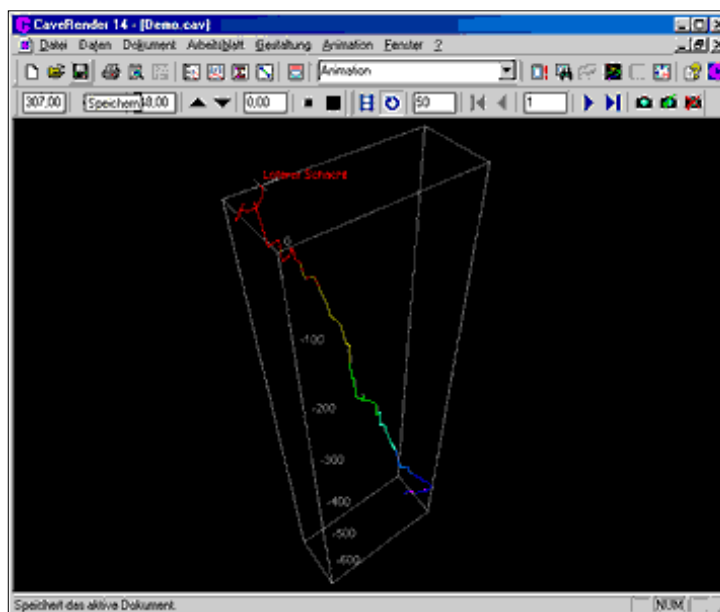


Рисунок 1.2 – Внешний вид приложения CaveRender

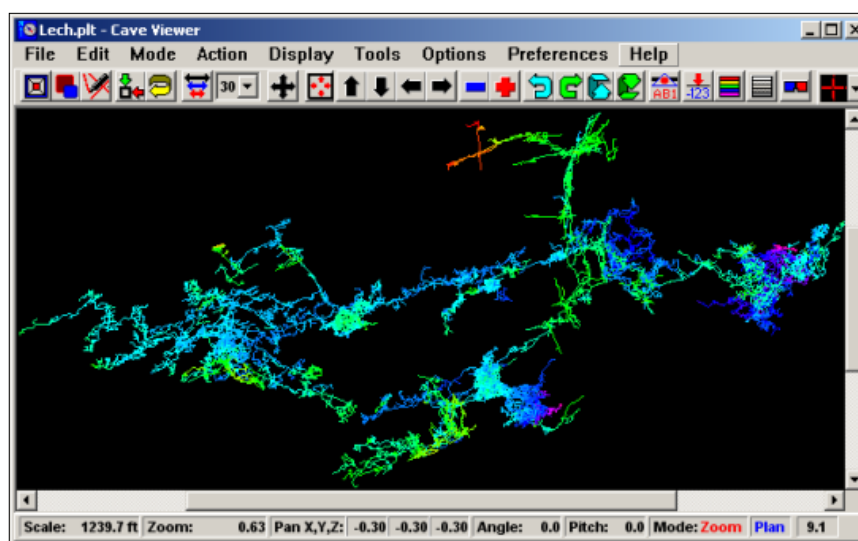


Рисунок 1.3 – Внешний вид приложения Compass

Visual Топо – кроме стандартного набора функций позволяет привязать к каждому пикету графический файл. Позволяет сохранить данные топосъёмки в формат MS Excel или в файл для загрузки в Compass. Внешний вид приложения представлен на рисунке 1.4.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		8

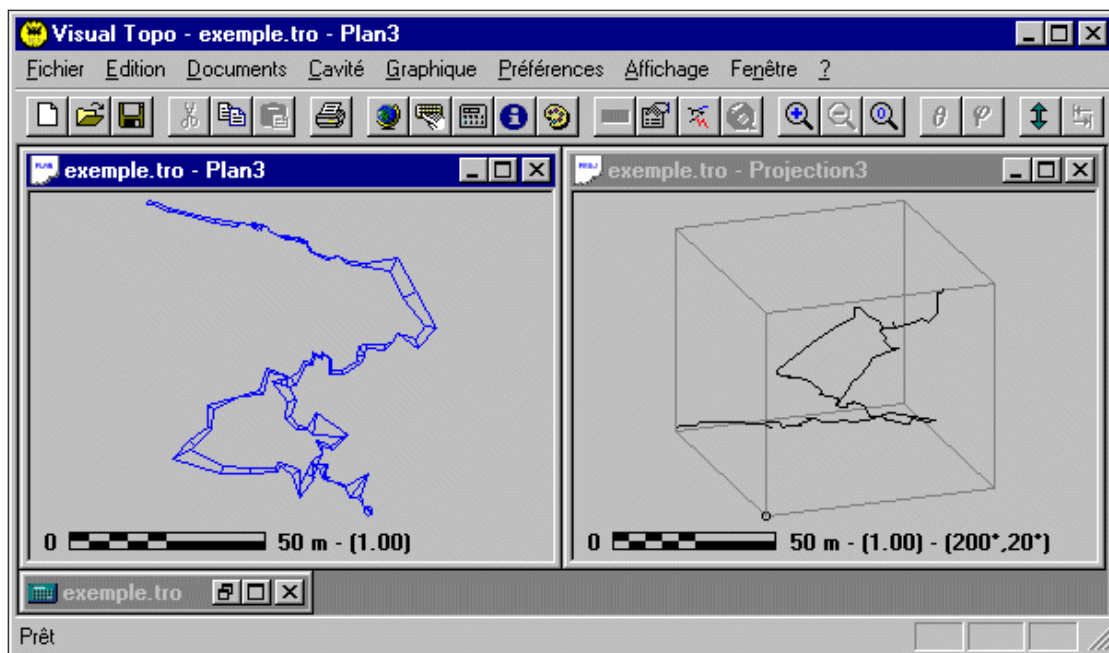


Рисунок 1.4 – Внешний вид приложения Visual Topo

Therion – бесплатное кросс-платформенное приложение для обработки данных топоъёмки, основанное на TCL (скриптовый язык высокого уровня). Данный продукт наиболее распространён среди российских спелеологов благодаря его гибкости и богатству функций, но обладает высоким порогом вхождения ввиду сложности работы с ним. Внешний вид программы в режиме сохранения карты в файл представлен на рисунке 1.5.

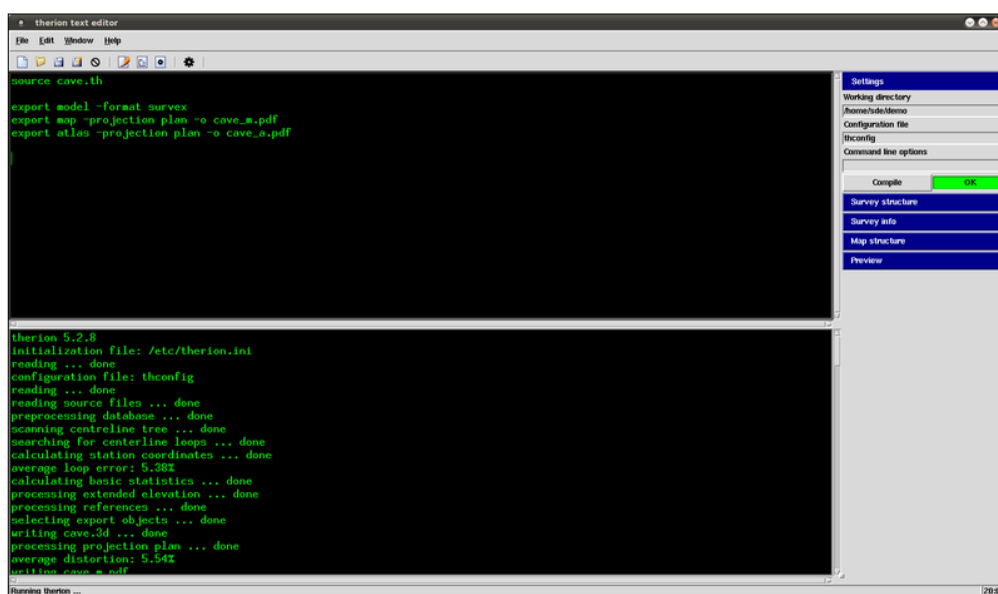


Рисунок 1.5 – Внешний вид приложения Therion

Изм.	Лист	№ докум.	Подп.	Дата

230105.000009П.308.ПЗ

Лист

9

1.2 Анализ необходимого функционала

В процессе составления карты пещеры перед спелеологом встаёт ряд последовательных задач.

Первым делом, по составленным таблицам и схемам, необходимо определить начальный пикет. Затем, последовательно переходя от пикета к пикету, нанести все точки на карту, убедиться что её внешний вид соответствует действительности, руководствуясь схемам, составленным во время съёмки, и, при необходимости внести исправления. Определить величину погрешности и, по возможности, исправить её. После этого необходимо определить ширину ходов пещеры и нанести изображения стен. На завершающем этапе нанести условные графические обозначения (УГО) и пометки.

На основе такой последовательности, определяется список задач, которые можно автоматизировать и возложить на компьютер.

Такие задачи, как математический расчёт координат пикетов, их отображение, внесение правок и графическая обработка результата быстрее и проще выполняется с помощью компьютера.

Тогда последовательность работы спелеолога меняется на следующую: определить начальный пикет; ввести полученные в результате топосъёмки данные в приложение; после того как приложение обработает данные и исправит обнаруженные погрешности, визуально оценить правильность внешнего вида, руководствуясь схемами, составленными при проведении топосъёмки; при необходимости внести правки во входные данные; обрисовать внутреннюю часть пещеры линиями, обозначающими стены; нанести на конечный результат пометки и УГО; сохранить результат в файл и вывести его на печать.

На основании вышеизложенного определяется минимальный функционал приложения:

- ввод и редактирование данных;
- преобразование исходных данных в относительные координаты;
- расчёт погрешностей измерения с их исправлением;
- вывод нитки хода на экран;
- предоставление инструмента для нанесения обрисовки стен;
- предоставление инструмента для нанесения пометок и УГО;
- предоставление инструмента для редактирования введённых графических данных;
- сохранение карты в графический файл.

Данные функции являются базовыми и не ограничивают дальнейшее развитие приложения. По мере работы с участниками спелеосекции функции приложения будут расширяться.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		10

2 Разработка приложения

2.1 Выбор среды разработки

При разработке эффективного продукта необходимо учитывать такие факторы, как соответствие продукта современным технологиям, хорошая производительность, устойчивость к сбоям, защищенность от постороннего доступа к важным данным. При этом на разработку продукта должно уходить разумное количество времени.

Основой создания приложения стала технология Windows Form. Это связано с тем, что данная технология имеет следующие возможности:

- полные возможности взаимодействия с существующим кодом;
- наглядность разработки, относительная простота кода;
- библиотека базовых классов, которая обеспечивает сокрытие всех сложностей, связанных с непосредственным использованием вызовов API;
- действительное упрощение процесса развертывания приложения;
- удобство использования для конечного потребителя;
- наглядность представления конечного результата;
- интуитивно понятный интерфейс.

В качестве языка программирования был выбран язык C#. C# – это язык программирования, синтаксические конструкции которого унаследованы не только от C++, но и от Visual Basic. Например, в C#, как и в Visual Basic, используются свойства классов. Как C++, C# позволяет производить перегрузку операторов для созданных типов. C# – это фактически гибрид разных языков. При этом C# синтаксически не менее (если не более) чист, чем Java, так же прост, как Visual Basic, и обладает практически той же мощностью и гибкостью, что и C++.

Подводя итоги, еще раз выделим основные особенности C#:

- управление памятью производится автоматически;
- в C# предусмотрены встроенные синтаксические конструкции для работы с перечислениями, структурами и свойствами классов;
- в C# осталась возможность перегружать операторы, унаследованные от C++. При этом значительная часть возникавших при этом сложностей ликвидирована;
- предусмотрена полная поддержка использования программных интерфейсов;
- предусмотрена полная поддержка аспектно-ориентированных программных технологий (таких как атрибуты), что позволяет присваивать типам характеристики для описания в будущем поведения данной сущности.

Visual Studio обеспечивает редактирование и расширение, что делает эту IDE первоклассным инструментом независимо от целевой платформы. Механизм макросов управляемого кода и IDE значительно упрощают запись и выполнение макросов. Усовершенствованный API расширения открывает доступ к тем частям

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		11

IDE, которые никогда не были доступны разработчикам, использующим Visual Studio.

В качестве графического движка выбрана свободно-распространяемая библиотека OpenGL. Выбор данной библиотеки обусловлен её открытостью, хорошей документированностью и кросс-платформенностью. Для работы с библиотекой в среде Visual Studio существует большое количество сторонних компонентов, таких как, например, Tao framework.

2.2 Описание обобщённого алгоритма приложения

2.2.1 Математическая часть приложения

После запуска приложения пользователь вводит данные топосъёмки в таблицу. По окончании ввода приложение преобразует сферические координаты точек в декартовы относительно начальной точки. Преобразование производится по формуле (2.1).

$$\begin{cases} x = r \cdot \cos(\theta) \cdot \sin(\varphi) \\ y = r \cdot \cos(\theta) \cdot \cos(\varphi) \\ z = r \cdot \sin(\theta), \end{cases} \quad (2.1)$$

где r – расстояние до следующей точки;
 θ – угол относительно горизонтали;
 φ – азимут (угол относительно севера).

В процессе преобразование исправляются погрешности, обнаруженные повторным измерением или измерением кольцевых ходов, путём смещения точек. Пример обнаруженной погрешности при измерении кольцевого хода приведён на рисунке 2.1.

Для разработки наиболее корректного алгоритма смещения точек необходимо принять во внимание следующие факты:

- погрешность измерения тем выше, чем больше расстояние между точками;
- суммарная погрешность увеличивается пропорционально количеству измерений.

Учитывая вышесказанное, можно определить следующие условия смещения: чем длиннее трасса или чем дальше очередной пикет от начальной точки, чем сильнее он смещается при корректировке. Таким образом, точка А на рисунке 2.1 должна иметь нулевое смещение, смещение точек В и F имеет уже ненулевую длину, при этом точка F смещается на большее расстояние чем точка В, но в обратном направлении, и так далее [16]. Принцип алгоритма показан на рисунке 2.2.

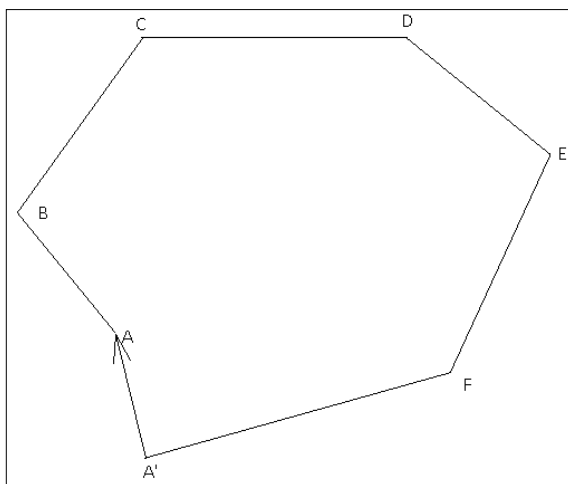


Рисунок 2.1 – Обнаружение погрешности при измерении кольцевого хода

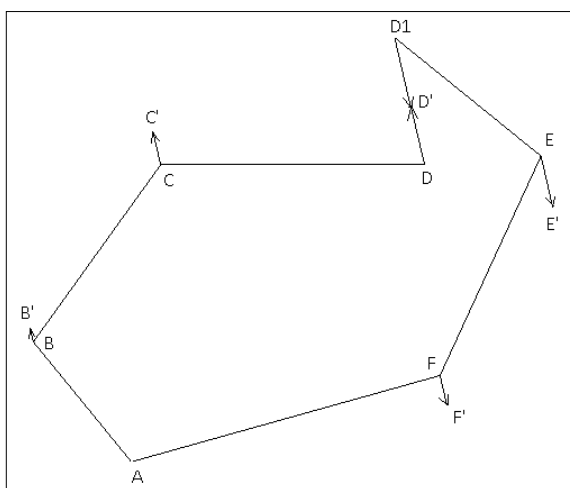


Рисунок 2.2 – Принцип алгоритма смещения

Следует заметить, что вектор смещения всех точек имеет одинаковый угол, но может отличаться по знаку. Конкретную длину каждого вектора можно определить по формуле (2.2).

$$d_i = \frac{d \cdot l_i}{l}, \quad (2.2)$$

где d_i – длина вектора для i -й точки;
 d – величина погрешности (длина отрезка DD1 на рисунке 2.2);
 l_i – расстояние от начальной до i -й точки;

l – общая длина кольцевого хода.

Полученные точки, соединённые линиями в соответствии с проведёнными измерениями, выводятся на экран в отдельном окне. Здесь же отображается упрощённая проекция пещеры, построенная при помощи трапеций. По этой проекции пользователь может получить представление о физических параметрах пещеры. По полученным трапециям пользователь проводит линии, соответствующие стенам пещеры, наносит УГО и пометки для удобства работы с картой. После чего карта сохраняется в файл для дальнейшей работы.

2.2.2 Графическая часть приложения

При помощи графического редактора можно вводит несколько видов символов: кривые, прямые, эллипсы, треугольники и другие.

Для рисования кривых используются сплайны.

Сплайн – гладкая кривая, которая проходит через две или более опорных точек, а также имеет расположенные вне ее управляющие точки, влияющие на форму сплайна. Наиболее общие типы сплайнов – кривые Безье и В-сплайны (B-spline curves). Сплайны состоят из вершин и сегментов. Каждая вершина сплайна имеет касательные векторы, снабженные на концах управляющими точками, или маркерами. Маркеры касательных векторов управляют кривизной сегментов сплайна при входе в вершину, которой принадлежат касательные векторы, и выходе из нее.

Кривые Безье разработаны математиком Пьером Безье. Кривые и поверхности Безье использовались в 60-х годах компанией «Рено» для компьютерного проектирования формы кузовов автомобилей. В настоящее время они широко используются в компьютерной графике.

Кривые Безье описываются в параметрической форме по формуле (2.3):

$$\begin{cases} x = P_x(t) \\ y = P_y(t) \end{cases} \quad (2.3)$$

где t – параметр, которому отвечают координаты отдельной точки линии.

Основные свойства кривой Безье:

- непрерывность заполнения сегмента между начальной и конечной точками;
- кривая всегда располагается внутри фигуры, образованной линиями, соединяющими контрольные точки;
- при наличии только двух контрольных точек сегмент представляет собой отрезок прямой линии;
- прямая линия образуется при коллинеарном (на одной прямой) размещении управляющих точек;
- кривая Безье симметрична, т.е. обмен местами между начальной и конечной точками (изменение направления траектории) не влияет на форму кривой;

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		14

- масштабирование и изменение пропорций кривой Безье не нарушает ее стабильности, так как она, с математической точки зрения, «аффинно инвариантна»;

- изменение координат хотя бы одной из точек ведет к изменению формы всей кривой Безье;

- степень кривой всегда на единицу меньше числа опорных точек (т.е. при трех опорных точках форма кривой – парабола);

- размещение дополнительных опорных точек вблизи одной позиции увеличивает ее «вес» и приводит к приближению траектории кривой к данной позиции;

- окружность не может быть описана параметрическим уравнением кривой Безье;

- невозможно создать параллельные кривые Безье, за исключением тривиальных случаев (прямые линии и совпадающие кривые).

Интерполяция сплайнами – метод, позволяющий получить в результате гладкую функцию. Достигается это следующим образом: между каждыми двумя точками функция интерполируется кубической параболой (то есть, полиномом третьей степени, причем функция подбирается так, чтобы на концах отрезка (в точках задания значений) совпадали с заранее заданными числами и сами значения функции, и значения производной. Совпадение значений функции требуется для обеспечения непрерывности получаемой кривой, а совпадение значений производной – для непрерывности производной, то есть для «гладкости» кривой. Таким образом, результат интерполяции получается составленным из кусков разных кубических парабол.

Кубический сплайн Эрмита по характеристикам близок к кривой Безье. Его можно рассматривать как иной способ задания кривой Безье. Вместо указания опорных точек в каждой точке кривой задается вектор. Направление и величина этого вектора определяют скорость с которой кривая будет отклоняться к заданной точке [15].

На рисунке 2.3 слева вид кривой определяется векторами U и V , заданными для точек A и D , а справа та же кривая задана опорными точками B и C . Для преобразования векторов к опорным точкам кривой Безье необходимо воспользоваться формулой (2.4):

$$\begin{cases} B = A + \frac{U}{3} \\ C = D + \frac{V}{3} \end{cases} \quad (2.4)$$

где A, B, C, D – опорные точки сплайна;

U, V – управляющие векторы сплайна.

Обратное преобразование производится по формуле (2.5):

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		15

$$\begin{cases} U = 3 \cdot (B - A) \\ V = 3 \cdot (D - C) \end{cases} \quad (2.5)$$

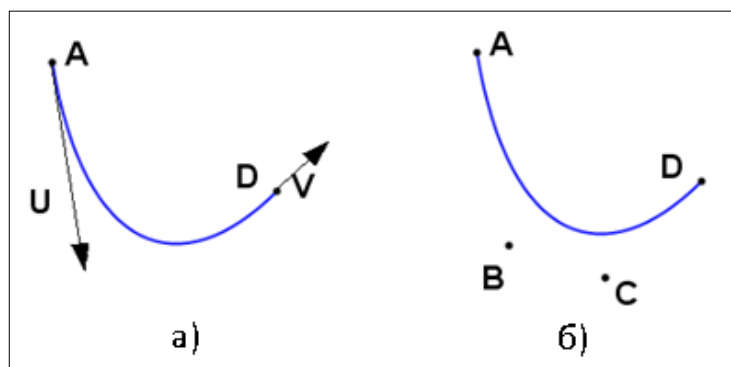


Рисунок 2.3 – Кривая определяемая векторами (а) и опорными точками (б)

Управляющий вектор для текущей точки равен половине вектора от предыдущей до следующей точки кривой. Например, на рисунке 2.4 нужно получить вектор для точки D из последовательности A, D, E. Для этого нужно взять отрезок AE, соединяющий две соседние с точкой D точки, и перенести его так, чтобы его центр совпал с точкой D. Половина этого отрезка и даст нужный вектор.

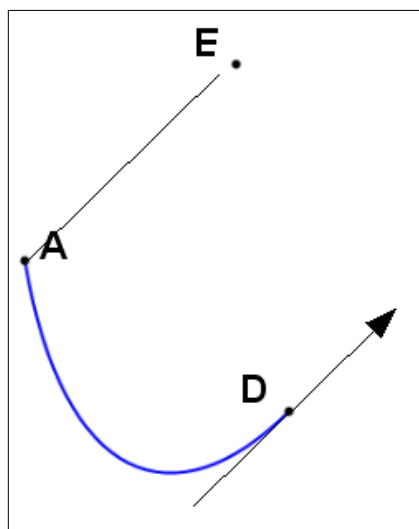


Рисунок 2.4 – Построение управляющего вектора

На концах последовательности применяется иной алгоритм получения направляющего вектора. В этом случае крайняя точка кривой считается соседней самой себе.

2.2.3 Сохранение и вывод результатов работы

Сохранение результатов работы приложения осуществляется на каждом её этапе. Все данные сохраняются в файл с рабочим расширением *.cv, представляющий собой ZIP-архив, содержащий в себе XML-файлы с описанием данных для работы приложения.

Для нормальной работы в файлы сохраняется следующая информация:

- входные данные;
- координаты пикетов;
- пометки и примечания;
- графические кривые;
- УГО.

Полученная в результате работы карта сохраняется в файл формата PDF, содержащий в себе название пещеры, изображение карты, данные об авторах съёмки, направление на север.

2.3 Разработка тестов

Для контроля правильности работы программы необходимо разработать серию тестов. Так как основную сложность представляет отладка математической части, тесты будут разработаны именно для этих функций приложения.

На вход функций будут подаваться данные топосъёмки с заранее известными декартовыми координатами пикетов, на выходе планируется получение данных с допустимой погрешностью.

Для проведения тестов был разработан пример изображающий проведённую топосъёмку с одним кольцевым ходом. Создание примера и все измерения были проведены при помощи инженерной системы автоматизированной разработки nanoCAD.

Графическое изображение тестового примера приведено на рисунке 2.3.

Входные данные подаются в виде последовательности значений: начальный пикет, конечный пикет, расстояние, угол относительно горизонта, угол относительно северного направления (азимут). Расстояниями до стен, пола и потолка для тестирования математической части приложения можно пренебречь. Так же для облегчения тестирования координата Z всех точек принимается равной 0, то есть угол относительно горизонта во входных данных всегда будет равен 0. Тестовые входные данные для приложения представлены в таблице 2.1.

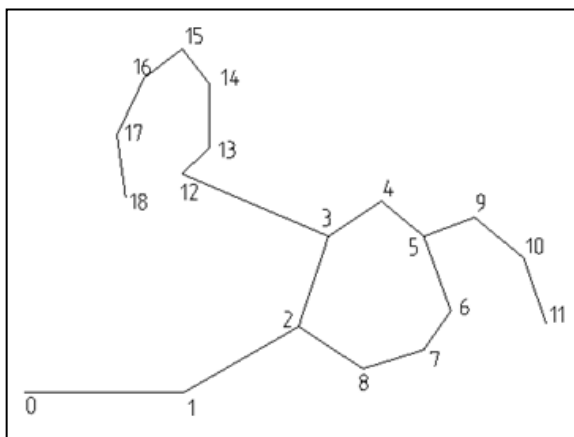


Рисунок 2.3 – Графическое изображение тестового примера

Таблица 2.1 – Тестовые входные данные

Начальный пикет	Конечный пикет	Расстояние, м	Угол над горизонтом	Азимут
0	1	5,0	0	90
1	2	4,2	0	60
2	3	3,0	0	18
3	4	2,0	0	56
4	5	1,7	0	130
5	6	2,5	0	160
6	7	1,5	0	215
7	8	2,0	0	-108
8	2	2,4	0	-57
5	9	1,7	0	65
9	10	2,0	0	130
10	11	2,2	0	161
3	12	5,0	0	-67
12	13	1,2	0	46
13	14	2,0	0	0
14	15	1,4	0	-38
15	16	1,5	0	233
16	17	2,0	0	205
17	18	2,0	0	172

Данные подаются на вход серии преобразовательных функций, в результате которых последовательно возвращаются такие данные как вычисленные координаты пикетов, набор списков пикетов, образующих циклы, длины этих циклов и векторы смещения пикетов для корректировки погрешностей, координаты пикетов после корректировки.

Длина кольцевого хода в тестовом примере составляет 15,1 м.

В результате обработки длина вектора смещения составила 0,034 м.

Результаты работы математического блока приложения приведены в таблице 2.2.

Таблица 2.2 – Координаты пикетов, полученные в результате работы

Пикет	Измеренные координаты		Вычисленные координаты		Координаты после корректировки	
	X	Y	X	Y	X	Y
0	0,000	0,000	0,000	0,000	0,000	0,000
1	0,000	5,000	0,000	5,000	0,000	5,000
2	8,642	2,092	8,637	2,100	8,637	2,100
3	9,577	4,942	9,564	4,953	9,571	4,955
4	11,241	6,052	11,222	6,072	11,242	6,075
5	12,577	4,942	12,525	4,979	12,539	4,983
6	13,425	2,591	13,413	2,64	13,4	2,636
7	12,571	1,358	12,552	1,411	12,543	1,408
8	10,661	0,764	10,65	0,793	10,645	0,791
9	14,169	5,539	14,065	5,697	14,08	5,702
10	15,710	4,264	15,598	4,412	15,612	4,416
11	16,426	2,184	16,314	2,332	16,328	2,336
12	4,980	6,909	4,962	6,907	4,968	6,909
13	5,838	7,747	5,825	7,740	5,832	7,742
14	5,838	9,747	5,825	9,740	5,832	9,742
15	4,980	10,85	4,963	10,844	4,970	10,846
16	3,785	9,943	3,765	9,941	3,772	9,943
17	2,932	8,134	2,920	8,128	2,926	8,130
18	3,203	6,153	3,198	6,148	3,205	6,150

2.4 Руководство пользователя

После запуска приложения пользователь вводит данные, полученные в результате проведения топографической съёмки. Данные вводятся в таблицу главного окна приложения (рисунок 2.4). При вводе следует соблюдать следующие правила:

- названия пикетов могут состоять из цифр и букв русского и латинского алфавита;
- названия пикетов могут содержать дополнительные символы (кавычки, апострофы и т.п.), но следует помнить, что подобные символы замедляют чтение карты;

- в названиях пикетов не должно быть пробелов, запятых, точек;
- данные о расстояниях между точками заносятся в метрах, об углах – в градусах;
- для отделения дробной части числа от целой следует использовать символ точки либо запятой;
- в поле ввода значений измеренных величин запрещается вводить какие-либо символы кроме цифр и разделителя между целой и дробной частью числа, (за исключением значений расстояний от пикета до стен, пола и потолка пещеры, здесь допускается написание двух чисел (одно для начального, второе для конечного пикета), разделённых символом обратной косой черты).

ЕТоро									
Файл Карта Съёмка									
От	До	Длина	Азимут	Угол	Влево	Вправо	Вверх	Вниз	Примечание
15	16	7.15	41.15	4.3	4.65\4.65	4.15\4.15	2.66\2.66	0.13\0.13	
16	17	3.12	322	-31.5	0.34	0.5	0.51	0	дорога в ад
16	а46	3.15	39	-4.3	1.9	1.3	1.9	0.1	
а46	18	1.92	308	-23.4	1.6	0.7	1	0.3	
18	19	1.65	320	-25.7	0.5	0.4	1.1	0.3	камни
19	20	1.68	33	-4.6	1.3	0.4	1.1	0	
20	21	0.76	48	9	0.6	0.6	0.7	0.4	
21	22	3.81	10	-20.8	1.4	1.7	1.9	0.3	влево 2 хода
а46	24	2.46	41	-4.3	1.6	1.5	2.4	0	
24	25	1.58	299	-10.4	0.3	0.3	0.6	0	
25	21	2.23	306	-32.2	0.6	0.6	0.7	0.4	
24	26	6.31	40.3	5.6	1.5	1.2	2.3	0.1	
26	27	5.22	40.5	6.1	1.5	0.9	1.7	0	
27	28	3.06	61	11.3	1.4	2.4	2	0.5	
28	29	3.65	54	-3.3	1.2	0.8	2.1	0	
29	30	3.89	43	-0.7	1.8	1.2	2.9	0.3	
30	а52	3.13	28	-8.4	4	0.8	2.5	2.7	
а52	K1	3.03	70	-53.2	3.2	3.1	5.6	0.2	
K1	K2	3	32.3	-25.8	0.8	1.1	6.3	0	
K2	31	8.16	27.5	31.9	3.1	1.7	2.3	0	

Рисунок 2.4 – Главное окно приложения

Вносятся данные о месте (название пещеры) и времени проведения топосъёмки, перечисляются авторы съёмки. Для этого необходимо выбрать пункт меню «Съёмка», подпункт «Данные съёмки», либо воспользоваться комбинацией клавиш Ctrl+D. При занесении данных об авторах так же следует придерживаться некоторых правил:

- в каждой строке поля вводятся данные (фамилия, имя и отчество) одного человека

- не следует разделять между собой данные разных людей с помощью запятых, точек и прочих знаков препинания.

Внешний вид окна занесения данных о съёмке приведён на рисунке 2.5.

По окончании ввода рекомендуется сохранить данные. Сохранение производится при помощи меню «Файл», подменю «Сохранить», либо комбинация клавиш Ctrl+S.

После того как данные введены, необходимо завершить графическое построение карты. Для этого следует перейти в окно графического редактора при помощи меню «Карта», подменю «2D», либо комбинацией клавиш Ctrl+G. В открывшемся окне сразу выводится полученное в результате обработки схематическое изображение карты пещеры. Здесь можно оценить правильность отображения (сравнить с зарисованной в блокноте схемой) и, в случае необходимости, вернуться в главное окно и внести изменения.

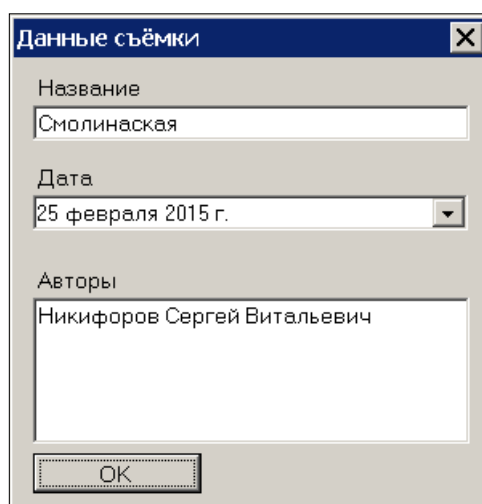


Рисунок 2.5 – Окно ввода данных о съёмке

В графическом редакторе для обрисовки стен пещеры нужно установить флажок Элементы карты и выбрать пункт Стены. Затем последовательно щелкнуть в ключевых точках в поле графического редактора, через указанные точки будет автоматически проведена кривая. По окончании ввода конкретной кривой следует нажать клавишу Enter, после чего кривая, обозначающая стену, зафиксируется в поле редактора, а её условное название отобразится в поле Список элементов. Нарисованную кривую можно изменить при помощи ключевых точек, перемещая их при помощи мыши.

Прекратить ввод можно нажатием клавиши Esc, в этом случае все введённые для текущей кривой точки будут удалены.

Удалить уже введенную кривую можно выделив её условное название в списке элементов (при этом кривая подсветится красным цветом) и нажать клавишу Delete.

Для обозначения продолжения пещеры (например неотснятые ходы) нужно выбрать пункт Ход, после чего редактор переходит в режим рисования прямой линии, которая задаётся начальной и конечной точками (вводятся щелчком мыши в соответствующих местах на схеме). После ввода конечной точки ввод данной прямой завершается, она фиксируется в поле редактора и её условное наименование отображается в поле «Список элементов».

После окончания ввода стен, можно приступить к вводу УГО. Для этого следует установить флажок «УГО» (при этом флажок «Элементы карты» автоматически снимется) и выбрать необходимый пункт. Следует заметить, что обрывы вводятся по тем же правилам, что и стены, остальные УГО обозначаются на карте простым нажатием в поле графического редактора.

По окончании работы с редактором, данные сохраняются в файл, либо экспортируются в файл PDF. Для сохранения необходимо воспользоваться меню «Файл», подменю «Сохранить» в окне графического редактора, либо сочетанием клавиш Ctrl+S. Экспорт в файл PDF осуществляется при помощи пункта меню «Файл», подпункт «Экспорт». Внешний вид графического редактора приведен на рисунке 2.6.

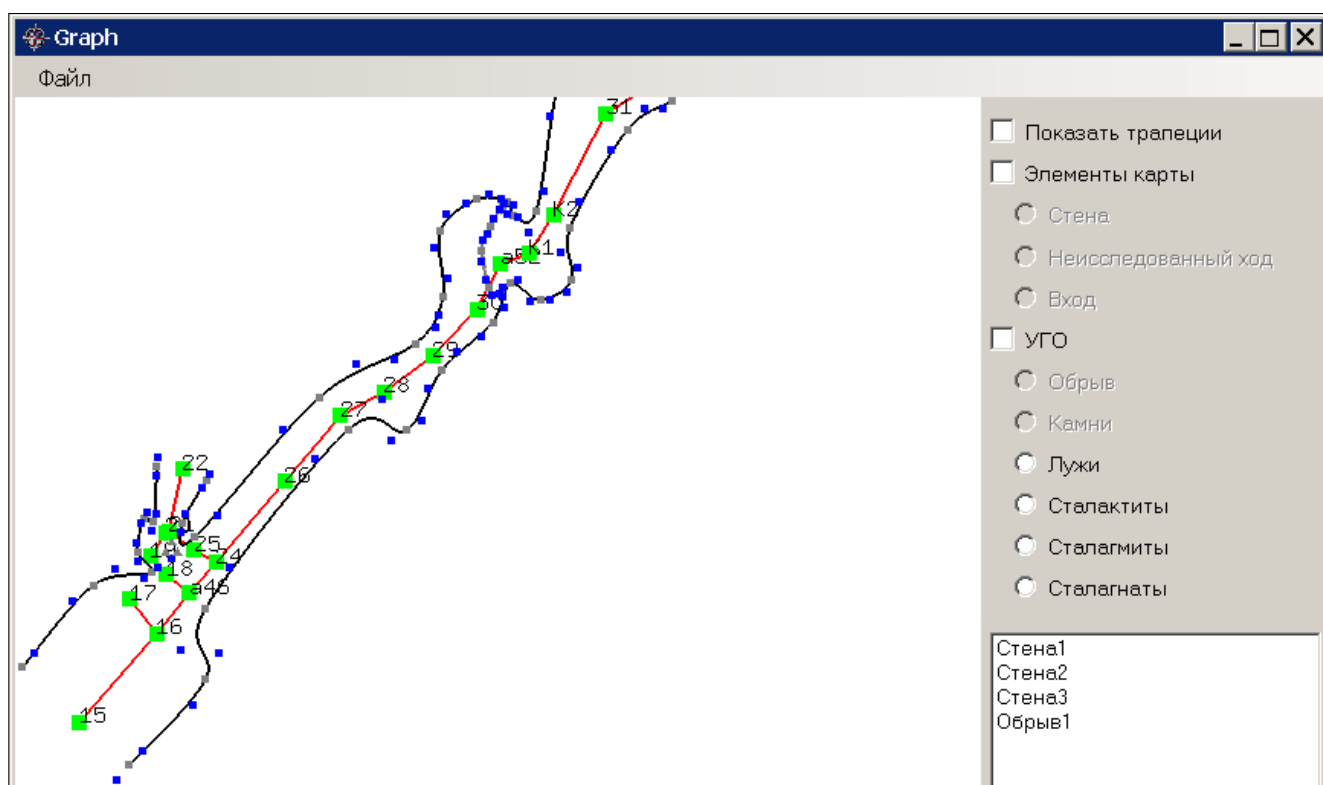


Рисунок 2.6 – Внешний вид графического редактора

2.5 Руководство программиста

Приложение предназначено для математической обработки результатов топосъемки пещер, последующей графической обработки материала и вывода результирующих данных в графическом представлении. Приложение выполняет следующие функции:

- предоставление интерфейса для ввода данных;
- проверка корректности введенных данных;
- преобразование введенных данных из сферических координат в декартовы;
- вычисление физических параметров пещеры (длина ходов, максимальная и минимальная высота, величина погрешности измерения);
- коррекция погрешностей измерения;
- построение схемы проведения топосъемки («нитка хода»);
- предоставление интерфейса для нанесения на схему стен и УГО;
- вывод полученных результатов в файл.

Для работы приложения предъявляются следующие минимальные требования к аппаратному и программному обеспечению:

- процессор: тактовая частота 1 ГГц;
- 1 Гб оперативной памяти;
- 10 Мб свободного места на жестком диске;
- графическое устройство DirectX 9;
- монитор;
- клавиатура;
- манипулятор «мышь»;
- операционная система MS Windows 7;
- .Net Framework версии 4.0.

Для нормальной работы приложения, выполняется контроль и минимальная корректировка входных данных (удаление лишних пробелов, замена разделителя десятичных знаков). Во избежание повреждения данных, осуществляется контроль корректности рабочих файлов при открытии с выдачей предупреждения в случае ошибки.

Входными являются данные, введенные пользователем. Формат данных результатов топосъемки описан в таблице 2.3.

Таблица 2.3 – Формат входных данных

Параметр	Обозначение в интерфейсе	Формат данных	Единица измерения	Тип данных
Начальный пикет	От	Текст	-	Строковый
Конечный пикет	До	Текст	-	Строковый
Расстояние между пикетами	Длина	Целое, либо десятичная дробь	Метры	Числовой с плавающей запятой

Продолжение таблицы 2.3

Параметр	Обозначение в интерфейсе	Формат данных	Единица измерения	Тип данных
Угол относительно северного направления	Азимут	Целое, либо десятичная дробь	Градусы	Числовой с плавающей запятой
Угол относительно горизонтального положения	Угол	Целое, либо десятичная дробь	Градусы	Числовой с плавающей запятой
Расстояние от конечного пикета до левой стены	Влево	Целое, либо десятичная дробь	Метры	Числовой с плавающей запятой
Расстояние от конечного пикета до правой стены	Вправо	Целое, либо десятичная дробь	Метры	Числовой с плавающей запятой
Расстояние от конечного пикета до потолка	Вверх	Целое, либо десятичная дробь	Метры	Числовой с плавающей запятой
Расстояние от конечного пикета до пола	Вниз	Целое, либо десятичная дробь	Метры	Числовой с плавающей запятой
Примечание	Примечание	Текст	-	Строковый

Выходными данными является графический файл в формате PDF с информацией о проведённой топосъёмке (название пещеры, дата проведения, авторы), указанием северного направления и картой пещеры.

В результате работы программы пользователю могут быть выданы информационные или предупреждающие сообщения. Текст сообщений и рекомендуемые действия описаны в таблице 2.4.

Таблица 2.4 – Обработка сообщений

Текст сообщений	Причина появления	Рекомендуемые действия
Ошибка чтения файла	Файл повреждён, либо имеет неизвестный формат	Выбрать другой файл для работы
Длина 100 м. Глубина 5 м. Неточность 2%	Сообщение имеет информационный характер, появляется после окончания математической обработки.	Закрыть окно сообщения и продолжить работу.
Сохранено	Сообщение имеет информационный характер, появляется после окончания процесса сохранения рабочего файла на диск	Закрыть окно сообщения и продолжить работу
GL_INVALID_ENUM - An unacceptable value has been specified for an enumerated argument. The offending function has been ignored.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение

Продолжение таблицы 2.4

Текст сообщений	Причина появления	Рекомендуемые действия
GL_INVALID_VALUE - A numeric argument is out of range. The offending function has been ignored.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
GL_INVALID_OPERATION - The specified operation is not allowed in the current state. The offending function has been ignored.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
GL_STACK_OVERFLOW - This function would cause a stack overflow. The offending function has been ignored.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
GL_STACK_UNDERFLOW - This function would cause a stack underflow. The offending function has been ignored.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
GL_OUT_OF_MEMORY - There is not enough memory left to execute the function. The state of OpenGL has been left undefined.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Window creation error. No window handle.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Can not create a GL device context.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Can not find a suitable PixelFormat.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Can not set the chosen PixelFormat.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Can not create a GL rendering context.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Can not activate the GL rendering context.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
No device or rendering context available!	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение
Unknown GL error. This should never happen.	Возникла ошибка в модуле OpenGL	Сохранить данные и перезапустить приложение

Для сохранения данных, используется файл с расширением *.cv, представляющий собой ZIP-архив, содержащий в себе XML-файлы с описанием данных для работы приложения. Структура файла *.cv описана в таблице 2.5.

Каждый файл в архиве представляет собой XML-файл с иерархической структурой, описывающий определённую сущность, необходимую для работы приложения.

Таблица 2.5 – Состав рабочего файла

Имя файла	Описание
trace.xml	Файл, содержащий входные данные результатов топосъемки. Структура файла описана в таблице 2.6
piquet.xml	Файл, содержащий данные по пикетам. Структура файла описана в таблице 2.7
spline.xml	Файл, содержащий данные по кривым карты. Структура файла описана в таблице 2.8
cgf.xml	Файл, содержащий данные по УГО карты. Структура файла описана в таблице 2.9

Таблица 2.6 – Описание файла trace.xml

Название элемента	Уровень вложенности	Родительский элемент	Описание
cave	0	-	Родительский элемент иерархической структуры, содержащий данные о пещере
title	1	cave	Название пещеры
survey	1	cave	Родительский элемент, содержащий данные о результатах конкретной топосъемки
date	2	survey	Дата проведения топосъемки
team	2	survey	Родительский элемент, содержащий данные об авторах топосъемки
author	3	team	Данные об авторе топосъемки
segment	2	survey	Родительский элемент, содержащий данные об отрезке между двумя пикетами
from	3	segment	Название начального пикета
to	3	segment	Название конечного пикета
tape	3	segment	Расстояние между пикетами
compass	3	segment	Угол между направлением на пикет и направлением на магнитный север
clino	3	segment	Угол между направлением на пикет и горизонталью
left	3	segment	Расстояние между конечным пикетом и левой стеной пещеры
right	3	segment	Расстояние между конечным пикетом и правой стеной пещеры
up	3	segment	Расстояние между конечным пикетом и потолком пещеры
down	3	segment	Расстояние между конечным пикетом и полом пещеры
f_left	3	segment	Расстояние между начальным пикетом и левой стеной пещеры
f_right	3	segment	Расстояние между начальным пикетом и правой стеной пещеры
f_up	3	segment	Расстояние между начальным пикетом и потолком пещеры
f_down	3	segment	Расстояние между начальным пикетом и полом пещеры
note	3	segment	Примечание к отрезку (к конечному пикету)

Таблица 2.7 – Описание файла piquet.xml

Название элемента	Уровень вложенности	Родительский элемент	Описание
cave	0	-	Родительский элемент иерархической структуры, содержащий данные о пещере
piquet	1	cave	Родительский элемент, содержащий данные о пикете
name	2	piquet	Название пикета
x	2	piquet	Координата X пикета
y	2	piquet	Координата Y пикета
z	2	piquet	Координата Z пикета
step	2	piquet	Минимально возможное количество пикетов между текущим пикетом и начальным
note	2	piquet	Примечание к пикету

Таблица 2.8 – Описание файла spline.xml

Название элемента	Уровень вложенности	Родительский элемент	Описание
cave	0	-	Родительский элемент иерархической структуры, содержащий данные о пещере
spline	1	cave	Родительский элемент, содержащий данные о кривой
name	2	spline	Название кривой
type	2	spline	Тип кривой
dirrection	2	spline	Направление кривой
points	2	spline	Родительский элемент, содержащий данные о точках кривой
x	3	points	Координата X точки
y	3	points	Координата Y точки
ra_x	3	points	Координата X направляющего вектора сегмента кривой, для которой точка является начальной
ra_y	3	points	Координата Y направляющего вектора сегмента кривой, для которой точка является начальной
rb_x	3	points	Координата X направляющего вектора сегмента кривой, для которой точка является конечной
rb_y	3	points	Координата Y направляющего вектора сегмента кривой, для которой точка является конечной

Таблица 2.9 – Описание файла cgn.xml

Название элемента	Уровень вложенности	Родительский элемент	Описание
cave	0	-	Родительский элемент иерархической структуры, содержащий данные о пещере
cgcn	1	cave	Родительский элемент, содержащий данные об УГО
name	2	cgcn	Название УГО
type	2	cgcn	Тип УГО

Продолжение таблицы 2.9

Название элемента	Уровень вложенности	Родительский элемент	Описание
X	2	сгп	Координата X УГО
Y	2	сгп	Координата Y УГО

					230105.000009П.308.ПЗ	Лист
						28
Изм.	Лист	№ докум.	Подп.	Дата		

2.6 Перспективы развития, использования приложения в дальнейшем

Текущая версия приложения является начальным этапом разработки, она работает лишь на операционных системах семейства MS Windows, более того, протестирована она была только на версиях MS Windows 7 с разрядностью 32 и 64 бит. В будущем предполагается перенос приложений на Linux-платформы, а так же перенос на мобильные платформы: Android, iOS, Windows Phone. Технологии, применяемые при разработке приложения, были выбраны с учётом возможности переноса на другие платформы.

Так же планируется значительное расширение функционала. В частности, данная версия приложения работает с пещерой как с одноуровневой сущностью, в дальнейшем планируется работа со слоями, для возможности прорисовки, отображения и тщательной работы с многоуровневыми пещерами, пещерами имеющими преимущественно вертикальное расположение. Внедрение слоёв позволяет работать с каждым уровнем пещеры как с отдельной сущностью.

Для детального изучения пещеры не всегда достаточно только её карты, чтобы оценить изменение рельефа по вертикали, зачастую требуется ещё и план-развёртка пещеры, изображающая её проекцию не на горизонтальную поверхность (плоскость XY), а на вертикальную. Поэтому в качестве результатов планируется выводить не только карту пещеры, но и её план-развёртку, позволяющую изучить пещеру в вертикальных разрезах.

Для поиска неизвестных ещё пещер перед спелеологами стоит задача исследования географической местности, определения областей карстования. Для проведения такого рода исследований необходимо нанесение существующих карт пещер на географические карты местности. В будущих версиях приложения необходимо будет обеспечить привязку карты пещеры к географическим координатам местности, предусмотреть возможность загружать фото и другие файлы, привязанные к той или иной области пещеры, предусмотреть возможность непосредственного нанесения карты пещеры на географические карты.

Для удобства работы конечного пользователя необходимо предусмотреть добавление примечаний и другой дополнительной информации прямо в графическом редакторе.

Разные спелеоорганизации используют различные методы съёмки. Зачастую совместная работа осложняется именно отсутствием возможности удалённой работы и результатами рабы других спелеологов. Для решения этой проблемы нужно рассмотреть возможность ввода входных данных в иных форматах и других системах координат (цилиндрической, снятой методом трилатерации).

В целях дальнейшего развития приложения, а так же для возможности обмена опытом и данными между спелеологами других спелеоорганизаций, предусмотреть сохранение данных в файлы других известных приложений, таких как Theriion, Compass и Visual Topo.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		29

3 Безопасность жизнедеятельности

3.1 Требования к помещениям при работе за компьютером

Помещения должны иметь естественное и искусственное освещение. Расположение рабочих мест за мониторами для взрослых пользователей в подвальных помещениях не допускается.

Площадь на одно рабочее место с компьютером для взрослых пользователей должна составлять не менее 6 м^2 , а объем не менее 20 м^3 .

Помещения с компьютерами должны оборудоваться системами отопления, кондиционирования воздуха или эффективной приточно-вытяжной вентиляцией.

Для внутренней отделки интерьера помещений с компьютерами должны использоваться диффузно-отражающие материалы с коэффициентом отражения для потолка – от 0,7 до 0,8; для стен – от 0,5 до 0,6; для пола – от 0,3 до 0,5.

Поверхность пола в помещениях эксплуатации компьютеров должна быть ровной, без выбоин, нескользкой, удобной для очистки и влажной уборки, обладать антистатическими свойствами.

В помещении должны находиться аптечка первой медицинской помощи, углекислотный огнетушитель для тушения пожара.

3.2 Требования к микроклимату, ионному составу и концентрации вредных химических веществ в воздухе помещений

На рабочих местах пользователей персональных компьютеров (ПК) должны обеспечиваться оптимальные параметры микроклимата в соответствии с СанПин 2.2.4.548-96. Согласно этому документу для категории тяжести работ 1а температура воздуха должна быть в холодный период года от 22 до 24°C , в теплый период года от 20 до 25°C . Относительная влажность должна составлять от 40 до 60%, скорость движения воздуха – $0,1 \text{ м/с}$. Для поддержания оптимальных значений микроклимата используется система отопления и кондиционирования воздуха. Для повышения влажности воздуха в помещении следует применять увлажнители воздуха с дистиллированной или кипяченой питьевой водой.

Ионный состав воздуха должен содержать следующее количество отрицательных и положительных аэроионов; минимально необходимый уровень 600 и 400 ионов в 1 см^3 воздуха; оптимальный уровень от 3000 до 5000 и 2000 ионов в 1 см^3 воздуха; максимально допустимый – 50000 ионов в 1 см^3 воздуха.

					230105.000009П.308.ПЗ	Лист
						30
Изм.	Лист	№ докум.	Подп.	Дата		

3.3 Требования к освещению помещений и рабочих мест

В компьютерных залах должно быть естественное и искусственное освещение. Естественное освещение обеспечивается через оконные проемы с коэффициентом естественного освещения (КЕО) не ниже 1,2% в зонах с устойчивым снежным покровом и не ниже 1,5% на остальной территории. Световой поток из оконного проема должен падать на рабочее место оператора с левой стороны.

Искусственное освещение в помещениях эксплуатации компьютеров должно осуществляться системой общего равномерного освещения.

Освещенность на поверхности стола в зоне размещения документа должна быть от 300 до 500 лк. Допускается установка светильников местного освещения для подсветки документов. Местное освещение не должно создавать бликов на поверхности экрана и увеличивать освещенность экрана более 300 лк. Прямую блескость от источников освещения следует ограничить. Яркость светящихся поверхностей (окна, светильники), находящихся в поле зрения, должна быть не более 200 кд/м².

Отраженная блескость на рабочих поверхностях ограничивается за счет правильного выбора светильника и расположения рабочих мест по отношению к естественному источнику света. Яркость бликов на экране монитора не должна превышать 40 кд/м². Показатель ослепленности для источников общего искусственного освещения в помещениях должен быть не более 20, показатель дискомфорта в административно-общественных помещениях не более 40. Соотношение яркости между рабочими поверхностями не должно превышать 3:1 – 5:1, а между рабочими поверхностями и поверхностями стен и оборудования 10:1.

Для искусственного освещения помещений с ПК следует применять светильники с зеркализированными решетками, укомплектованные высокочастотными пускорегулирующими аппаратами. Допускается применять светильники прямого света, преимущественно отраженного света с люминисцентными лампами. Допускается применение светильников местного освещения с лампами накаливания. Светильники должны располагаться в виде сплошных или прерывистых линий сбоку от рабочих мест параллельно линии зрения пользователя при разном расположении компьютеров. При периметральном расположении – линии светильников должны располагаться локализованно над рабочим столом ближе к его переднему краю, обращенному к оператору. Защитный угол светильников должен быть не менее 40 градусов. Светильники местного освещения должны иметь непросвечивающийся отражатель с защитным углом не менее 40 градусов.

Для обеспечения нормативных значений освещенности в помещениях следует проводить чистку стекол оконных проемов и светильников не реже двух раз в год и проводить своевременную замену перегоревших ламп.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		31

3.4 Требования к шуму и вибрации в помещениях

Уровни шума на рабочих местах пользователей ПК не должны превышать значений, установленных СанПиН 2.2.4/2.1.8.562-96 и составляют не более 50 дБА. На рабочих местах в помещениях для размещения шумных агрегатов уровень шума не должен превышать 75 дБА, а уровень вибрации в помещениях допустимых значений по СН 2.2.4/2.1.8.566-96 категория 3, тип «в».

Снизить уровень шума в помещениях можно использованием звукопоглощающих материалов с максимальными коэффициентами звукопоглощения в области частот от 63 до 8000 Гц для отделки стен и потолка помещений. Дополнительный звукопоглощающий эффект создают однотонные занавески из плотной ткани, повешенные в складку на расстоянии от 15 до 20 см от ограждения. Ширина занавески должна быть в 2 раза больше ширины окна.

3.5 Требования к организации и оборудованию рабочих мест

Рабочие места с ПК по отношению к световым проемам должны располагаться так, чтобы естественный свет падал сбоку, желательно слева.

Схемы размещения рабочих мест с ПК должны учитывать расстояния между рабочими столами с мониторами: расстояние между боковыми поверхностями мониторов не менее 1,2 м, а расстояние между экраном монитора и тыльной частью другого монитора не менее 2,0 м.

Рабочий стол может быть любой конструкции, отвечающей современным требованиям эргономики и позволяющей удобно разместить на рабочей поверхности оборудование с учетом его количества, размеров и характера выполняемой работы. Целесообразно применение столов, имеющих отдельную от основной столешницы специальную рабочую поверхность для размещения клавиатуры. Используются рабочие столы с регулируемой и нерегулируемой высотой рабочей поверхности. При отсутствии регулировки высота стола должна быть в пределах от 680 до 800 мм.

Глубина рабочей поверхности стола должна составлять 800 мм (допускаемая не менее 600 мм), ширина – соответственно 1600 мм и 1200 мм. Рабочая поверхность стола не должна иметь острых углов и краев, иметь матовую или полуматовую фактуру.

Рабочий стол должен иметь пространство для ног высотой не менее 600 мм, шириной – не менее 500 мм, глубиной на уровне колен – не менее 450 мм и на уровне вытянутых ног – не менее 650 мм.

Быстрое и точное считывание информации обеспечивается при расположении плоскости экрана ниже уровня глаз пользователя, предпочтительно

					230105.000009П.308.ПЗ	Лист
						32
Изм.	Лист	№ докум.	Подп.	Дата		

перпендикулярно к нормальной линии взгляда (нормальная линия взгляда 15 градусов вниз от горизонтали).

Клавиатура должна располагаться на поверхности стола на расстоянии от 100 до 300 мм от края, обращенного к пользователю.

Для удобства считывания информации с документов применяются подвижные подставки (пюпитры), размеры которых по длине и ширине соответствуют размерам устанавливаемых на них документов. Пюпитр размещается в одной плоскости и на одной высоте с экраном.

Для обеспечения физиологически рациональной рабочей позы, создания условий для ее изменения в течение рабочего дня применяются подъемно-поворотные рабочие стулья с сиденьем и спинкой, регулируемые по высоте и углам наклона, а также расстоянию спинки от переднего края сидения.

Конструкция стула должна обеспечивать:

- ширину и глубину поверхности сиденья не менее 400 мм;
- поверхность сиденья с закругленным передним краем;
- регулировку высоты поверхности сиденья в пределах от 400 до 550 мм и углом наклона вперед до 15 градусов и назад до 5 градусов;
- высоту опорной поверхности спинки 300 ± 20 мм, ширину – не менее 380 мм и радиус кривизны горизонтальной плоскости 400 мм;
- угол наклона спинки в вертикальной плоскости в пределах 0 ± 30 градусов;
- регулировку расстояния спинки от переднего края сидения в пределах от 260 до 400 мм;
- стационарные или съемные подлокотники длиной не менее 250 мм и шириной от 50 до 70 мм;
- регулировку подлокотников по высоте над сиденьем в пределах 230 ± 30 мм и внутреннего расстояния между подлокотниками в пределах от 350 до 500 мм.;
- поверхность сиденья, спинки и подлокотников должна быть полумягкой, с нескользящим неэлектризующимся, воздухопроницаемым покрытием, легко очищаемым от загрязнения.

Рабочее место должно быть оборудовано подставкой для ног, имеющей ширину не менее 300 мм, глубину не менее 400 мм, регулировку по высоте в пределах до 150 мм и по углу наклона опорной поверхности подставки до 20 градусов. Поверхность подставки должна быть рифленой и иметь по переднему краю бортик высотой 10 мм.

3.6 Режим труда и отдыха при работе с компьютером

Режим труда и отдыха предусматривает соблюдение определенной длительности непрерывной работы на ПК и перерывов, регламентированных с

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		33

учетом продолжительности рабочей смены, видов и категории трудовой деятельности.

Виды трудовой деятельности на ПК разделяются на 3 группы: группа А – работа по считыванию информации с экрана с предварительным запросом; группа Б – работа по вводу информации; группа В – творческая работа в режиме диалога с ПК.

Если в течение рабочей смены пользователь выполняет разные виды работ, то его деятельность относят к той группе работ, на выполнение которой тратится не менее 50% времени рабочей смены.

Категории тяжести и напряженности работы на ПК определяются уровнем нагрузки за рабочую смену: для группы А – по суммарному числу считываемых знаков; для группы Б – по суммарному числу считываемых или вводимых знаков; для группы В – по суммарному времени непосредственной работы на ПК. В таблице 3.1 приведены категории тяжести и напряженности работ в зависимости от уровня нагрузки за рабочую смену.

Таблица 3.1 – Виды категорий трудовой деятельности с ПК

Категория работы по тяжести и напряжённости	Уровень нагрузки за рабочую смену при видах работы на ПК		
	Группа А, количество знаков	Группа Б, количество знаков	Группа В, время работы, ч
I	До 20000	До 15000	До 2,0
II	До 40000	До 30000	До 4,0
III	До 60000	До 40000	До 6,0

Количество и длительность регламентированных перерывов, их распределение в течение рабочей смены устанавливается в зависимости от категории работ на ПК и продолжительности рабочей смены.

При восьмичасовой рабочей смене и работе на ПК регламентированные перерывы следует устанавливать:

- для первой категории работ через 2 часа от начала смены и через 2 часа после обеденного перерыва продолжительностью 15 минут каждый;
- для второй категории работ – через 2 часа от начала рабочей смены и от 1,5 до 2,0 часов после обеденного перерыва продолжительностью 15 минут каждый или продолжительностью 10 минут через каждый час работы;
- для третьей категории работ – от 1,5 до 2,0 часов от начала рабочей смены и от 1,5 до 2,0 часов после обеденного перерыва продолжительностью 20 минут каждый или продолжительностью 15 минут через каждый час работы.

При двенадцатичасовой рабочей смене регламентированные перерывы должны устанавливаться в первые 8 часов работы аналогично перерывам при восьмичасовой рабочей смене, а в течение последних 4 часов работы, независимо от категории и вида работ, каждый час продолжительностью 15 минут.

Продолжительность непрерывной работы на ПК без регламентированного перерыва не должна превышать 2 часа.

При работе на ПК в ночную смену продолжительность регламентированных перерывов увеличивается на 60 минут независимо от категории и вида трудовой деятельности.

Эффективными являются нерегламентированные перерывы (микропаузы) длительностью до 3 минут.

Регламентированные перерывы и микропаузы целесообразно использовать для выполнения комплекса упражнений и гимнастики для глаз, пальцев рук, а также массажа. Комплексы упражнений целесообразно менять через 3 недели.

Пользователям ПК, выполняющим работу с высоким уровнем напряженности, показана психологическая разгрузка во время регламентированных перерывов и в конце рабочего дня в специально оборудованных помещениях (комнатах психологической разгрузки).

Медико-профилактические и оздоровительные мероприятия. Все профессиональные пользователи ПК должны проходить обязательные предварительные медицинские осмотры при поступлении на работу, периодические медицинские осмотры с обязательным участием терапевта, невропатолога и окулиста, а также проведением общего анализа крови и ЭКГ.

Близорукость, дальнозоркость и другие нарушения рефракции должны быть полностью скорректированы очками. Для работы должны использоваться очки, подобранные с учетом рабочего расстояния от глаз до экрана дисплея. При более серьезных нарушениях состояния зрения вопрос о возможности работы на ПК решается врачом-офтальмологом.

3.7 Обеспечение электробезопасности и пожарной безопасности на рабочем месте

3.7.1 Электробезопасность

На рабочем месте пользователя размещены дисплей, клавиатура и системный блок. При включении дисплея на электронно-лучевой трубке создается высокое напряжение в несколько киловольт. Поэтому запрещается прикасаться к тыльной стороне дисплея, вытирать пыль с компьютера при его включенном состоянии, работать на компьютере во влажной одежде и влажными руками.

Перед началом работы следует убедиться в отсутствии свешивающихся со стола или висящих под столом проводов электропитания, в целостности вилки и провода электропитания, в отсутствии видимых повреждений аппаратуры и рабочей мебели, в отсутствии повреждений и наличии заземления приэкранного фильтра.

					230105.000009П.308.ПЗ	Лист
						35
Изм.	Лист	№ докум.	Подп.	Дата		

Токи статического электричества, наведенные в процессе работы компьютера на корпусах монитора, системного блока и клавиатуры, могут приводить к разрядам при прикосновении к этим элементам. Такие разряды опасности для человека не представляют, но могут привести к выходу из строя компьютера. Для снижения величин токов статического электричества используются нейтрализаторы, местное и общее увлажнение воздуха, использование покрытия полов с антистатической пропиткой.

3.7.2 Пожарная безопасность

Пожарная безопасность – состояние объекта, при котором исключается возможность пожара, а в случае его возникновения предотвращается воздействие на людей опасных его факторов и обеспечивается защита материальных ценностей.

Противопожарная защита – это комплекс организационных и технических мероприятий, направленных на обеспечение безопасности людей, предотвращение пожара, ограничение его распространения, а также на создание условий для успешного тушения пожара.

Пожарная безопасность обеспечивается системой предотвращения пожара и системой пожарной защиты. Во всех служебных помещениях обязательно должен быть план эвакуации людей при пожаре, регламентирующий действия персонала в случае возникновения очага возгорания и указывающий места расположения пожарной техники.

Пожары в вычислительных центрах (ВЦ) представляют особую опасность, так как сопряжены с большими материальными потерями. Характерная особенность ВЦ – небольшие площади помещений. Как известно, пожар может возникнуть при взаимодействии горючих веществ, окислителя и источников зажигания. В помещениях ВЦ присутствуют все три основных фактора, необходимые для возникновения пожара.

Горючими компонентами на ВЦ являются: строительные материалы для акустической и эстетической отделки помещений, перегородки, двери, полы, перфокарты и перфоленты, изоляция кабелей и другие.

Источниками зажигания в ВЦ могут быть электрические схемы от электронных вычислительных машин (ЭВМ), приборы, применяемые для технического обслуживания, устройства электропитания, кондиционирования воздуха, где в результате различных нарушений образуются перегретые элементы, электрические искры и дуги, способные вызвать загорания горючих материалов.

В современных ЭВМ очень высокая плотность размещения элементов электронных схем. В непосредственной близости друг от друга располагаются соединительные провода, кабели. При протекании по ним электрического тока выделяется значительное количество теплоты. При этом возможно оплавление изоляции. Для отвода избыточной теплоты от ЭВМ служат системы вентиляции и кондиционирования воздуха. При постоянном действии эти системы представляют собой дополнительную пожарную опасность.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		36

Для большинства помещений ВЦ установлена категория пожарной опасности В.

Одна из наиболее важных задач пожарной защиты – защита строительных помещений от разрушений и обеспечение их достаточной прочности в условиях воздействия высоких температур при пожаре. Учитывая высокую стоимость электронного оборудования ВЦ, а также категорию его пожарной опасности, здания для ВЦ и части здания другого назначения, в которых предусмотрено размещение ЭВМ, должны быть первой и второй степени огнестойкости. Для изготовления строительных конструкций используются, как правило, кирпич, железобетон, стекло, металл и другие негорючие материалы. Применение дерева должно быть ограничено, а в случае использования необходимо пропитывать его огнезащитными составами.

					230105.000009П.308.ПЗ	Лист
						37
Изм.	Лист	№ докум.	Подп.	Дата		

4 Экономическое обоснование

4.1 Расчёт трудозатрат на разработку программного продукта

В общем, на разработку отводится 600 часов. Все работы проводятся одним человеком. Структура трудозатрат на разработку приложения представлена в таблице 4.1.

Таблица 4.1 – Величина и структура трудозатрат на разработку приложения

Виды трудозатрат	Индекс	Трудоёмкость, человеко-час	Структура трудозатрат, %
Затраты труда на исследование предметной области и разработку алгоритма	t_r	160	33,33
Затраты труда на разработку блок-схемы приложения	t_b	4	0,83
Затраты труда на программирование	t_p	236	49,17
Затраты труда на отладку приложения на ПК	t_d	20	4,17
Затраты труда на подготовку документации	t_{doc}	60	12,50
Итого	T	480	100,00

4.2 Составление сметы затрат на разработку программного продукта

Смета затрат на разработку программного продукта включает в себя такие статьи как:

- затраты на оплату машинного времени;
- затраты на оплату электроэнергии;
- затраты на оплату труда программисту;
- страховые взносы во внебюджетные фонды;
- прочие затраты.

Свердловская городская спелеосекция является некоммерческой организацией, все исследования и работы проводятся на энтузиазме, поэтому затраты на оплату труда программиста составляют 0 рублей.

Затраты на оплату машинного времени при разработке и отладке программы определяется по формуле (4.1).

$$З_м = С \cdot Т, \quad (4.1)$$

где $Т$ – трудоёмкость разработки программного продукта;
 $С$ – цена машино-часа (руб/час).
 Цена машино-часа рассчитывается по формуле (4.2).

$$С = \frac{З_{ам} + З_{вм} + З_{тр} + З_{пр}}{Т_{пк}}, \quad (4.2)$$

где $Т_{пк}$ – действительный годовой фонд времени ПК, час/год;
 $З_{ам}$ – годовые издержки на амортизацию, руб/год;
 $З_{вм}$ – годовые издержки на вспомогательные материалы, руб/год;
 $З_{тр}$ – годовые издержки на текущий ремонт компьютера, руб/год;
 $З_{пр}$ – годовые издержки на прочие и накладные расходы, руб/год.
 Действительный годовой фонд времени ПК определяется по формуле (4.3).

$$Т_{пк} = N_m \cdot N_d \cdot N_{ч}, \quad (4.3)$$

где N_m – число месяцев в году;
 N_d – среднее число рабочих дней в месяце;
 $N_{ч}$ – средняя продолжительность рабочего дня.

Если принять среднюю продолжительность рабочего дня за 8 часов, а среднее число рабочих дней в месяце за 22 дня, то действительный годовой фонд времени ПК составит $Т_{пк} = 12 \cdot 22 \cdot 8 = 2112$ рублей.

Годовые издержки на амортизацию вычисляются по формуле (4.4).

$$З_{ам} = С_{бал} \cdot N_{ам}, \quad (4.4)$$

где $С_{бал}$ – балансовая стоимость компьютера;
 $N_{ам}$ – норма амортизации, %.
 Балансовая стоимость определяется по формуле (4.5).

$$С_{бал} = С_{рын} \cdot З_{уст}, \quad (4.5)$$

где $С_{рын}$ – рыночная стоимость компьютера;
 $З_{уст}$ – затраты на доставку и установку ПК.
 Затраты на доставку и установку ПК обычно составляют от 1 до 10% от стоимости компьютера. Годовые издержки на вспомогательные материалы

составляют от 1 до 2% от балансовой стоимости компьютера. Затраты на текущий ремонт ПК составляют от 5 до 6% от балансовой стоимости компьютера. Годовые издержки на прочие и накладные расходы включают затраты на оплату услуг ЖКХ, связи, аренды и пр. и составляют от 5 до 8% от балансовой стоимости компьютера.

Рыночная стоимость компьютера складывается из стоимости составляющих его компонентов, она представлена в таблице 4.2.

Таблица 4.2 – Рыночная стоимость ПК

Наименование составляющей	Описание	Стоимость, руб
Процессор	CPU Intel Core i3-4160 3.6 GHz/2core/SVGA HD Graphics4400/0.5+3Mb/54W/5 GT/s LGA1150	8841,00
Материнская плата	ASUS H81M-K (RTL) LGA1150 <H81> PCI-E Dsub+DVI GbLAN SATA MicroATX 2DDR-III	2740,00
Оперативная память	2 шт. Crucial <CT25664BA160B> DDR-III DIMM 2Gb <PC3-12800>	2704,00
Жесткий диск	HDD 1 Tb SATA 6Gb/s Seagate Barracuda <ST1000DM003> 3.5" 7200rpm 64Mb	3798,00
Корпус	Minitower INWIN EMR002 <Black> Micro ATX 350W (24+4пин) <6053538>	3522,00
Монитор	18.5" ЖК монитор PHILIPS 193V5LSB2 / 10 / 62 (LCD, Wide, 1366x768, D-Sub)	5012,00
Мышь проводная	A4-Tech Optical Mouse < OP-720-Silver > (RTL) PS / 2 3btn+Roll	170,00
DVD-привод	DVD RAM & DVD±R/RW & CDRW Samsung SH-224DB <Black> SATA (OEM)	984,00
Клавиатура	A4-Tech KR-85 < USB > 104КЛ	382,00
Кулер	Arctic Cooling Alpine 11 GT rev.2 Cooler (775/1155, 500-2000об/мин, Al)	516,00
Программное обеспечение	Microsoft "Windows 7 Домашняя Базовая" Русская версия DVD (BOX)	6720,00
Итого		35389,00

После определения рыночной стоимости компьютера вычисляются остальные, зависящие от неё величины.

Затраты на доставку и установку ПК (7% от рыночной стоимости):

$$Z_{\text{уст}} = 35389 \cdot \frac{7}{100} = 2477,23 \text{ рублей}$$

Балансовая стоимость ПК:

$$C_{\text{бал}} = 35389 + 2477,23 = 37866,23 \text{ рублей.}$$

Годовые издержки на вспомогательные материалы (1,5% от балансовой стоимости):

$$З_{\text{вм}} = 37866,23 \cdot \frac{1,5}{100} = 2477,23 = 567,99 \text{ рублей.}$$

Затраты на текущий ремонт ПК (5,5% от балансовой стоимости):

$$З_{\text{тр}} = 37866,23 \cdot \frac{5,5}{100} = 2082,64 \text{ рублей.}$$

Годовые издержки на прочие и накладные расходы (7% от балансовой стоимости):

$$З_{\text{пр}} = 37866,23 \cdot \frac{7}{100} = 2650,64 \text{ рублей.}$$

Затраты на амортизацию оборудования при норме амортизации $H_{\text{ам}} = 20\%$ составляют:

$$З_{\text{ам}} = 37866,23 \cdot \frac{20}{100} = 7573,25 \text{ рублей.}$$

Учитывая проведённые вычисления, цена машино-часа составляет:

$$C = \frac{7573,25 + 567,99 + 2082,64 + 2650,64}{2112} = 6,1 \text{ рублей.}$$

Тогда затраты на оплату машинного времени при написании и отладке приложения составляют:

$$З_{\text{м}} = 6,1 \cdot (236 + 20 + 60) = 1926,1 \text{ рублей.}$$

Затраты на оплату электроэнергии определяются по формуле (4.6).

$$З_{\text{э}} = C_{\text{э}} \cdot P \cdot T, \quad (4.6)$$

где $C_{\text{э}}$ – стоимость электроэнергии;

P – мощность, потребляемая одним ПК;

T – Суммарное время программирования, отладки и подготовки документации.

Мощность, потребляемая одним ПК складывается из мощности, потребляемой монитором и мощности, потребляемой системным блоком. Суммарная мощность составляет 0,8 кВт. Средняя стоимость электроэнергии для Екатеринбурга составляет 3,4 рублей за кВт.

Тогда затраты на оплату электроэнергии составят:

$$З_{\text{э}} = 3,4 \cdot 0,8 \cdot (236 + 20 + 60) = 859,52 \text{ рублей.}$$

Калькуляция сметной стоимости приведена в таблице 4.3.

Таблица 4.3 – Калькуляция сметной стоимости

Название статьи затрат	Сумма, руб
Затраты на оплату машинного времени	1926,10
Затраты на оплату электроэнергии	859,52
Затраты на оплату труда программисту	0,00

Продолжение таблицы 4.3

Страхование взносов во внебюджетные фонды	0,00
Итого:	2785,62

В соответствии с расчётами, себестоимость разработки приложения составляет 2785,62 рублей.

4.3 Экономическая эффективность от использования программного продукта

Экономическую эффективность от использования программного продукта в случае с некоммерческой организацией можно оценить как экономию времени работы и связанную с этим экономию затрат на оплату электроэнергии и оплату машинного времени в соответствии с формулой (4.7). Так же играют роль экономия на альтернативных издержках, но этот показатель плохо поддаётся вычислениям в виду своей неоднозначности.

$$\mathcal{E} = \mathcal{E}_m + \mathcal{E}_z, \quad (4.7)$$

где \mathcal{E}_m – экономия затрат на оплату машинного времени;

\mathcal{E}_z – экономия затрат на оплату электроэнергии.

Свердловская городская спелеосекция проводит порядка 6 съёмок пещер средней сложности ежемесячно. Обработка полученных данных с использованием сторонних программных продуктов занимает от 7 до 10 часов. Опытным путём определено, что обработка данных с использованием разрабатываемого приложения занимает от 4 до 5 часов. Таким образом, экономия времени, согласно формуле (4.8), за один месяц составляет:

$$\mathcal{E}C_m = (8 - 4) \cdot 6 = 24 \text{ часа.}$$

$$\mathcal{E}C_m = (C_c - C_n) \cdot K, \quad (4.8)$$

где C_c – количество часов на обработку данных по одной пещере с использованием сторонних программных продуктов;

C_n – количество часов на обработку данных по одной пещере с использованием разрабатываемого приложения;

K – количество съёмок пещер за месяц.

Для вычисления экономии затрат на оплату машинного времени можно вновь воспользоваться формулой (4.1), но вместо трудоёмкости разработки

программного продукта подставляется месячная экономия времени. И она составит:

$$\mathcal{E}_m = 24 \cdot 6,1 = 146,4 \text{ рублей.}$$

Для вычисления экономии затрат на оплату электроэнергии так же можно воспользоваться уже известной формулой (4.6). Экономия составит:

$$\mathcal{E}_3 = 3,4 \cdot 0,8 \cdot 24 = 65,28 \text{ рублей.}$$

Итоговая экономия составит:

$$\mathcal{E} = 146,4 + 65,28 = 211,68 \text{ рублей в месяц.}$$

Срок окупаемости программного продукта рассчитывается по формуле (4.9).

$$C_o = \frac{Z_c}{\mathcal{E}}, \quad (4.9)$$

где Z_c – затраты на разработку программного продукта (себестоимость);

\mathcal{E} – экономия от использования программного продукта.

Таким образом срок окупаемости составит:

$$C_o = \frac{2785,62}{211,68} = 13,16 \text{ месяцев.}$$

4.4 Обобщение результатов расчётов

Результаты расчётов сводятся в таблицу 4.4.

Таблица 4.4 – Сводная таблица показателей

Наименование показателя	Значение
Трудозатраты на разработку программного продукта, человеко-час	480
Затраты на разработку программного продукта, руб.	2785,62
Ежемесячная экономия от использования программного продукта, руб.	211,68
Срок окупаемости, мес.	13,16

По результатам оценки экономической эффективности, видно, что проект окупится менее чем через 14 месяцев, следовательно, можно сделать вывод, что разработка приложения является экономически эффективной.

5 Определение интегрального критерия уровня готовности к информационному обществу по результатам полученным в дипломном проекте

Сегодня использование информационных и коммуникационных технологий (ИКТ) во многих сферах услуг и промышленности стало совершенно необходимым элементом конкурентной борьбы и стратегического развития. Глобальное значение этого явления состоит в том, что знания, зафиксированные в технологиях, компьютерных системах, становятся объективным фактором развития. В настоящий момент масштабы данного процесса становятся столь значительны, что придают ему новое качество – возникает информационная цивилизация, где знания становятся основой существования. Причем знания эти настолько специализированы и сложны, что для их использования необходимы специальные средства – информационные и коммуникационные технологии.

Широко распространенный термин «экономика знания» четко характеризует тенденцию формирования экономики информационного общества. Конечно, традиционная промышленность, сельское хозяйство и сфера услуг попрежнему преобладают в абсолютных объемах производства, но наибольшие темпы роста, доходность наблюдаются именно в отраслях информационной индустрии. Производство знаний становится важнейшей функцией общества, обеспечивающей его выживание и дальнейший прогресс.

В этих условиях особую актуальность приобретает задача – превратить потенциальные возможности ИКТ в реальную силу развития. По мнению многих ведущих экспертов, эти технологии, хотя и не решат всех проблем, но могут за счет качественного улучшения возможностей обмена информацией значительно облегчить создание новой социальной и экономической инфраструктуры. А именно новая инфраструктура и обеспечивает устойчивое экономическое развитие.

Критерии оценки уровня готовности России, ее структур административно-хозяйственного деления и отраслей к информационному обществу предназначены для определения их принадлежности к одному из четырех уровней готовности к информационному обществу.

Первый уровень – условия для перехода к информационному обществу отсутствуют;

Второй уровень – условия для перехода к информационному обществу созданы и используются малой частью общества;

Третий уровень – условия для перехода к информационному обществу созданы и используются значительной частью общества. Существует электронное расслоение общества;

Четвертый уровень – большая часть общества использует в жизни и в профессиональной деятельности инфокоммуникационные технологии, услуги и информационные ресурсы. Осуществляются меры по преодолению электронного расслоения общества и других негативных последствий информатизации.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		44

Интегральный критерий для региона К рассчитывается по методу Международной Академии Связи и представляется вектором, имеющим модуль (длина вектора) и угол, определяющий положение вектора относительно единичного вектора с единичными координатами, либо относительно вектора, выбранного за точку отсчета в n-мерном пространстве. Для его расчета необходимо определить следующие критериальные показатели.

Экономический критерий $K_{\text{Э}}$ определяется по формуле (5.1):

$$K_{\text{Э}} = \frac{K_{\text{Эф}} - K_{\text{Эmin}}}{K_{\text{Эmax}} - K_{\text{Эmin}}}, \quad (5.1)$$

где $K_{\text{Эф}}$ – процент ВВП региона, обеспеченный за счет информационно-телекоммуникационной экономики; он может быть исчислен как отношение доходов от инфокоммуникаций к ВВП;

$$K_{\text{Эmin}} = 0\%;$$

$$K_{\text{Эmax}} = 50\%.$$

На 2009 год:

$$K_{\text{Э}} = \frac{11 - 0}{50 - 0} = 0,22.$$

на 2014 год:

$$K_{\text{Э}} = \frac{22 - 0}{50 - 0} = 0,44.$$

Комплексный информационно-технологический критерий $K_{\text{ИТ}}$ определяется с использованием методики программы развития Организации Объединенных Наций по формуле (5.2):

$$K_{\text{ИТ}} = \frac{K_{\text{Т}} + K_{\text{Д}} + K_{\text{И}} + K_{\text{РИ}}}{4}, \quad (5.2)$$

где $K_{\text{Т}}, K_{\text{Д}}, K_{\text{И}}, K_{\text{РИ}}$ – критериальные показатели.

Критерий территориальности $K_{\text{Т}}$ характеризует процент покрытия территории региона инфокоммуникационными сетями, определяется с использованием методики Программы развития Организации Объединенных Наций по формуле (5.3):

$$K_{\text{Т}} = \frac{K_{\text{Тф}} - K_{\text{Тmin}}}{K_{\text{Тmax}} - K_{\text{Тmin}}}, \quad (5.3)$$

где $K_{Тф}$ – фактический процент территории региона, покрытой инфокоммуникационными сетями;

$$K_{Тmin} = 0\%;$$

$$K_{Тmax} = 100\%.$$

На 2009 год:

$$K_{Т} = (85 - 0)/(100 - 0) = 0,85.$$

На 2014 год:

$$K_{Т} = (97 - 0)/(100 - 0) = 0,97.$$

Критерий доступности $K_{Д}$ определяет время, необходимое жителю страны для достижения инфокоммуникаций.

Критерий доступности рассчитывается по формуле (5.4):

$$K_{Д} = \alpha \cdot K_{Д1} + \beta \cdot K_{Д2}, \quad (5.4)$$

где $K_{Д1}$ – критерий доступности для городских жителей и жителей крупных поселков;

$K_{Д2}$ – критерий доступности для жителей труднодоступных и малонаселенных районов;

α, β – доли населения, проживающего, соответственно, в городской и сельской местности.

В свою очередь, критерии $K_{Д1}, K_{Д2}$ рассчитываются по формуле (5.5):

$$K_{Д1,2} = 1 - \frac{t - t_{min}}{t_{max} - t_{min}}, \quad (5.5)$$

где t – время, необходимое любому жителю для достижения инфокоммуникаций в регионе (стране);

$$\left. \begin{array}{l} t_{min} = 5 \text{ мин.} \\ t_{max} = 15 \text{ мин.} \end{array} \right\} \text{ для } K_{Д1};$$

$$\left. \begin{array}{l} t_{min} = 30 \text{ мин.} \\ t_{max} = 90 \text{ мин.} \end{array} \right\} \text{ для } K_{Д2};$$

На 2009 год

$$K_{Д1} = 1 - \frac{10 - 5}{15 - 5} = 0,5,$$

$$K_{Д2} = 1 - \frac{50 - 30}{90 - 30} = 0,67,$$

$$K_{Д} = 0,8 \cdot 0,5 + 0,2 \cdot 0,67 = 0,53.$$

На 2014 год:

$$K_{Д1} = 1 - \frac{6 - 5}{15 - 5} = 0,9,$$

$$K_{Д_2} = 1 - \frac{40 - 30}{90 - 30} = 0,83,$$

$$K_{Д} = 0,8 \cdot 0,9 + 0,2 \cdot 0,83 = 0,89.$$

Информационный критерий $K_{И}$ определяется как индекс объема интерактивной информации в год на 1 жителя и рассчитывается по формуле (5.6):

$$K_{И} = \frac{\lg I - \lg I_{\min}}{\lg I_{\max} - \lg I_{\min}} \quad (5.6)$$

$$I = \frac{N_{ТА} \cdot V \cdot T}{H}, \quad (5.7)$$

где t – объем интерактивной информации в год на 1 жителя региона (страны);
 $N_{ТА}$ – количество телефонов (стационарных и мобильных) в регионе (стране);

H – население региона;

V – средняя скорость передачи информации;

T – среднее время занятия канала;

$I_{\min} = 10^5$ байт;

$I_{\max} = 10^{10}$ байт.

На 2009 год:

$$I = \frac{(824916 + 1374860) \cdot 64 \cdot 10^3 \cdot 32850}{1458960} = 3,17 \cdot 10^9 \text{ байт},$$

$$K_{И} = \frac{\lg 3,17 \cdot 10^9 - \lg 10^5}{\lg 10^{10} - \lg 10^5} = 0,9.$$

На 2014 год:

$$I = \frac{(1312853 + 1571660) \cdot 64 \cdot 10^3 \cdot 32850}{1855642} = 3,27 \cdot 10^9 \text{ байт},$$

$$K_{И} = \frac{\lg 3,27 \cdot 10^9 - \lg 10^5}{\lg 10^{10} - \lg 10^5} = 0,91.$$

Критерий развития инфокоммуникаций $K_{Т}$ характеризует уровень наполнения общества инфокоммуникационными устройствами (терминалами) пользователя, определяется с использованием методики Программы развития Организации Объединенных Наций по формуле (5.8):

$$K_{РИ} = \frac{I_{ТА} + I_{СОТ} + I_{ПК} + I_{ТВ}}{4} \quad (5.8)$$

$$I_{TA} = \frac{D - D_{\min}}{D_{\max} - D_{\min}} \quad (5.9)$$

$$D = \frac{N_{TA} \cdot 100}{H}, \quad (5.10)$$

где D – плотность стационарных телефонов в регионе (стране);

H – население региона;

$D_{\max} = 80\%$;

$D_{\min} = 0\%$;

I_{COT} , I_{PK} , I_{TB} рассчитываются аналогично I_{TA} ;

$D_{COT_{\max}} = D_{PK_{\max}} = D_{TB_{\max}} = 100\%$;

$D_{COT_{\min}} = D_{PK_{\min}} = D_{TB_{\min}} = 0\%$;

На 2009 год:

$$I_{TA} = \frac{\frac{824916 \cdot 100}{1458960} - 0}{80 - 0} = 0,71;$$

$$I_{COT} = \frac{\frac{1374860 \cdot 100}{1458960} - 0}{100 - 0} = 0,94;$$

$$I_{PK} = \frac{\frac{216891 \cdot 100}{1458960} - 0}{100 - 0} = 0,15;$$

$$I_{TB} = \frac{\frac{1050451 \cdot 100}{1458960} - 0}{100 - 0} = 0,72;$$

$$K_{PI} = \frac{0,71 + 0,94 + 0,15 + 0,72}{4} = 0,63.$$

На 2014 год:

$$I_{TA} = \frac{\frac{1312853 \cdot 100}{1855642} - 0}{80 - 0} = 0,88;$$

$$I_{COT} = \frac{\frac{1571660 \cdot 100}{1855642} - 0}{100 - 0} = 0,85;$$

$$I_{PK} = \frac{\frac{742257 \cdot 100}{1855642} - 0}{100 - 0} = 0,4;$$

$$I_{TB} = \frac{\frac{1410288 \cdot 100}{1855642} - 0}{100 - 0} = 0,76;$$

$$K_{\text{ри}} = \frac{0,88 + 0,85 + 0,4 + 0,76}{4} = 0,72.$$

Определим комплексный информационно-технологический критерий по формуле

На 2009 год:

$$K_{\text{ит}} = \frac{0,85 + 0,53 + 0,9 + 0,63}{4} = 0,73.$$

На 2014 год:

$$K_{\text{ит}} = \frac{0,97 + 0,89 + 0,91 + 0,72}{4} = 0,87.$$

Результаты расчетов сведены в таблицу 5.1

Таблица 5.1 - Результаты расчетов.

Критерии	2009	2014
$K_{\text{э}}$	0,22	0,44
$K_{\text{т}}$	0,85	0,97
$K_{\text{д}}$	0,53	0,89
$K_{\text{и}}$	0,90	0,91
$K_{\text{ри}}$	0,63	0,72
$K_{\text{ит}}$	0,73	0,87

Все критериальные показатели являются относительными величинами, не имеющими размерности и изменяющимися от нуля до единицы. Приближение к единице означает движение к информационному обществу. Следует заметить, что определение конкретного значения показателя $K_{\text{ит}}$ в регионах России в настоящее время затруднительно из-за недостатка справочных данных: нет данных по доступности и степени покрытия инфокоммуникаций в регионах, время занятия канала и скорость передачи информации известны только как средние величины в целом по России. Поэтому $K_{\text{ит}}$ может быть пока оценен только критерием $K_{\text{ри}}$.

Данные критерии оценки готовности не в полной мере отвечают в целом сегодняшнему состоянию российской экономики. С точки зрения Всемирного банка, Россия входит в число 22 трансформирующихся стран. Однако в дореформенный период в нашей стране были заложены определенные предпосылки для развития процессов информатизации, которые отсутствуют в развивающихся странах. Вместе с тем, как в стране с переходной экономикой, в которой государство во многих случаях берет на себя функции рыночных институтов, в России только начинается формироваться конкурентная среда, что является серьезным препятствием на пути информатизации. Все же полученная в результате интегральная оценка готовности России или региона к информационному обществу позволит сопоставить имеющуюся ситуацию с положением дел в развивающихся, других странах или оценить динамику движения к информационному обществу в конкретном регионе административно-

территориального образования. Общую готовность России к жизни в информационном мире оценивают как близкую к третьему уровню из четырех, что отвечает состоянию «в движении» (начало развития информационного общества, массового использования ИКТ населением страны, а также появление информационного неравенства). Хотя реальный уровень развития информационного общества в России невысок, страна находится на самом динамично отрезке пути.

Кроме того, отмечен высокий уровень информационного неравенства - как для различных групп населения, так и для отдельных регионов. Результаты расчета показателей могут быть представлены в виде пентаграммы, представленной на рисунке 5.1.

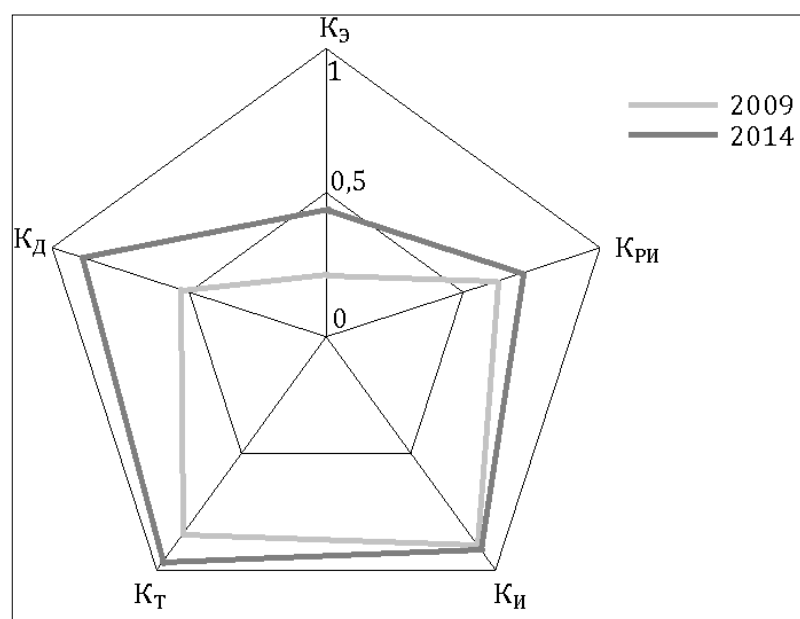


Рисунок 5.1 – Движение по пути к информационному обществу

Из сделанных расчетов можно сделать вывод, что быстро развивающаяся область телекоммуникаций приближает Россию к третьему уровню готовности к информационному обществу, что подразумевает доступ значительной части населения к качественным телекоммуникационным услугам.

Заключение

Выбор программного обеспечения в области спелеологии обуславливается поставленными задачами и особенностью рельефа местности. Подобрать универсальный инструмент достаточно сложно. С другой стороны, многие из существующих систем, обладающие достаточным функционалом слишком дороги и их не могут себе позволить организации, не имеющие сторонней финансовой поддержки.

В ходе работы над дипломным проектом был проанализирован рынок аналогичных продуктов. На основании этого исследования разработан функционал приложения, необходимый для работы в Свердловской городской спелеосекции с учётом пожеланий самих потенциальных пользователей приложения.

На этапе разработки был сделан выбор средств разработки, соответствующий поставленной задаче. Проведены исследования в соответствующих областях математики и геологии для разработки необходимых алгоритмов. Определены задачи для каждого из модулей приложения и разработаны необходимые тесты, подтверждающие правильность работы данного модуля. Разработано приложение для автоматизации обработки результатов топосъёмки. Проведено исследование по определению минимальных системных требований для работы приложения.

Одним из преимуществ этого продукта является ориентированность на конкретного заказчика. Дополнительно к этому снижаются затраты на обработку результатов, снижается порог вхождения новых специалистов.

В процессе работы было проведено исследование экономической эффективности разрабатываемого программного продукта, вычислен срок его окупаемости и, на основании полученных результатов, принимая во внимание факт окупаемости менее 1,5 лет, сделан вывод об эффективности разработки.

Была разработана справочная документация, облегчающая работу с приложением конечного пользователя, разработаны рекомендации для предотвращения нанесения вреда здоровью при работе с ПК. Было проведено обучение участников спелеосекции по работе с приложением.

Разработана документация, необходимая для дальнейшего сопровождения приложения. В процессе обучения потенциальных пользователей был накоплен необходимый материал для дальнейшей разработки. Были определены приоритеты развития приложения с учётом специфики работы спелеосекции и географическими предпочтениями работы её участников.

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		51

Библиография

- 1 Бахвалов Н.С. Численные методы / Н.С.Бахвалов, Н.П.Жидков, Г.М.Кобельков. – М.: Бином. 2007 – 627 с.
- 2 Геодезия. Изд. 2-е переработанное / Данилов В.В. и др. – М.: Недра. 1976 – 488 с.
- 3 Гидрогеология и карстоведение. Методика изучения карста: Межвузовский сборник научных трудов / под ред. И.А. Печеркин. – Пермь: Пермский университет, 1987 – 184 с.
- 4 Грачёв А.П. Топографо-геодезические работы в горизонтальных пещерах. Практические рекомендации для спелотопографа / А.П. Грачёв. – Киев: Самиздат, 2010 – 48 с.
- 5 Дублянский В.Н. Путешествия под землёй / В.Н. Дублянский, В.В. Илюхин. – М.: Физкультура и спорт, 1968 – 80 с.
- 6 Дьяконов В.П. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ / В.П. Дьяконов. – М.: Наука, 1989 – 240 с.
- 7 Киреев В.И. Полиномиальные интегродифференциальные одномерные и двумерные сплайны / В. И. Киреев, Т. К. Бирюкова. – М.: Московский государственный авиационный институт, 1998 – 16 с.
- 8 Компьютерная графика и стандарт OpenGL, 3-е издание / пер. с англ. – М.: Издательский дом «Вильямс», 2005 – 1168 с.
- 9 Макконнел С. Совершенный код. Мастер-класс / С. Макконнел; пер. с англ. М.: Издательско-торговый дом «Русская редакция». – СПб.: Питер, 2005 – 896 с.
- 10 Мартин Р. Чистый код: создание, анализ и рефакторинг. Библиотека программиста / Р. Мартин. – СПб.: Питер, 2010 – 464 с.
- 11 Носач В.В. Решение задач аппроксимации с помощью персональных компьютеров / В.В. Носач. – М.: МИКАП, 1994 – 382 с.
- 12 Смолич Б.А. Уравнительные вычисления / Б.А. Смолич – М.: Недра. 1989 – 245 с.
- 13 Введение в топосъемку пещер [Электронный ресурс] – Режим доступа: <http://barrier.marshruty.ru/Arts/Biblio.aspx>
- 14 ГОСТ 19.504-79. Руководство программиста. Требования к содержанию и оформлению [Электронный ресурс] – Режим доступа: http://www.rugost.com/index.php?option=com_content&view=article&id=64:19504-79 &catid=19&Itemid=50
- 15 Интерполяция сплайнами [Электронный ресурс] – Режим доступа: <http://alglib.sources.ru/interpolation/spline3.php>
- 16 Методика описания пещер [Электронный ресурс] – Режим доступа: http://sablino.narod.ru/library/met_opisan.htm
- 17 Ссылки по вопросам топосъемки [Электронный ресурс] – Режим доступа: <http://www.rgo-speleo.ru/topo/topolinks.htm>

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		52

18 Уроки OpenGL + C# [Электронный ресурс] – Режим доступа:
<http://esate.ru/uroki/OpenGL/uroki-OpenGL-c-sharp/?rv1>

19 Posts Tagged «Tao Framework» [Электронный ресурс] – Режим доступа:
<https://slizerboy.wordpress.com/tag/tao-framework>

20 Programming: theory and practice by alexBlack [Электронный ресурс] –
Режим доступа: <http://alex-black.ru/article.php?content=109>

					230105.000009П.308.ПЗ	Лист
						53
Изм.	Лист	№ докум.	Подп.	Дата		

Приложение А (справочное) Листинг программы

```
using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Windows.Forms;
using System.Xml;
using ETopo.Properties;
using Ionic.Zip;
using Tao.FreeGlut;

namespace ETopo
{
    public partial class FrMain : Form
    {
        private List<Piquet> _piquetLst = new List<Piquet>();
        private List<Trace> _traceList = new List<Trace>();
        private string _name;
        private string _date;
        private List<string> _autor;

        public FrMain()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            Glut.glutInit();
            Glut.glutInitDisplayMode(Glut.GLUT_RGB | Glut.GLUT_DOUBLE | Glut.GLUT_DEPTH);
            dgTopo.BackgroundImage = Resources.cell;
        }

        private void LoadMenu_Click(object sender, EventArgs e)
        {
            if (odLoad.ShowDialog() != DialogResult.OK) return;
            try
            {
                using (var zip = ZipFile.Read(odLoad.FileName))
                {
                    using (var fl = new MemoryStream())
                    {
                        var ent = zip["trace.xml"];
                        ent.Extract(fl);
                        fl.Position = 0;
                        using (var reader = XmlReader.Create(fl))
                        {
                            var trc = new Trace();
                        }
                    }
                }
            }
        }
    }
}
```

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		54

```

while (reader.Read())
{
    if (reader.NodeType == XmlNodeType.Element)
    {
        switch (reader.Name)
        {
            case "team":
                _autor = new List<string>();
                break;
            case "name":
                reader.Read();
                _autor.Add(reader.Value);
                break;
            case "date":
                reader.Read();
                _date = reader.Value;
                break;
            case "title":
                reader.Read();
                _name = reader.Value;
                break;
            case "segment":
                dgTopo.Rows.Add();
                trc = new Trace();
                break;
            case "from":
                reader.Read();
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[0].Value = reader.Value;
                trc.From = reader.Value;
                break;
            case "to":
                reader.Read();
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[1].Value = reader.Value;
                trc.To = reader.Value;
                break;
            case "tape":
                reader.Read();
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[2].Value = reader.Value;
                trc.Tape = Convert.ToDouble(reader.Value.Replace('.', ','));
                break;
            case "compass":
                reader.Read();
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[3].Value = reader.Value;
                trc.Azimuth = Convert.ToDouble(reader.Value.Replace('.', ','));
                break;
            case "clino":
                reader.Read();
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[4].Value = reader.Value;
                trc.Clino = Convert.ToDouble( reader.Value .Replace ('.', ','));
                break;
            case "left":
                reader.Read();
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[5].Value = reader.Value;
                trc.Left = Convert.ToDouble(reader.Value.Replace('.', ','));
                break;
        }
    }
}

```

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		55

```

        case "right":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[6].Value = reader.Value;
            trc.Right = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "up":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[7].Value = reader.Value;
            trc.Up = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "down":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[8].Value = reader.Value;
            trc.Down = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "f_left":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[5].Value = reader.Value +
                "\\\" + dgTopo.Rows[dgTopo.RowCount - 2].Cells[5].Value;
            trc.FromLeft = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "f_right":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[6].Value = reader.Value +
                "\\\" + dgTopo.Rows[dgTopo.RowCount - 2].Cells[6].Value;
            trc.FromRight = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "f_up":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[7].Value = reader.Value + "\\\" +
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[7].Value;
            trc.FromUp = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "f_down":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[8].Value = reader.Value + "\\\" +
                dgTopo.Rows[dgTopo.RowCount - 2].Cells[8].Value;
            trc.FromDown = Convert.ToDouble(reader.Value.Replace('.', ','));
            break;
        case "note":
            reader.Read();
            dgTopo.Rows[dgTopo.RowCount - 2].Cells[9].Value = reader.Value;
            trc.Note = reader.Value;
            break;
    }

}

else if (reader.NodeType == XmlNodeType.EndElement
    && reader.Name == "segment")
{
    _traceList.Add(trc);
}
} //while(reader.Read())
}

```

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		56

```

}
using (var fl = new MemoryStream())
{
    var pqEnt = zip["piquet.xml"];
    pqEnt.Extract(fl);
    fl.Position = 0;
    using (var pqReader = XmlReader.Create(fl))
    {

        var piquet = new Piquet();
        while (pqReader.Read())
        {
            if (pqReader.NodeType == XmlNodeType.Element)
            {
                switch (pqReader.Name)
                {
                    {
                        case "piquet":
                            if (pqReader.IsEmptyElement) break;
                            piquet = new Piquet();
                            break;
                        case "name":
                            if (pqReader.IsEmptyElement) break;
                            pqReader.Read();
                            piquet.Name = pqReader.Value;
                            break;
                        case "note":
                            if (pqReader.IsEmptyElement) break;
                            pqReader.Read();
                            piquet.Note = pqReader.Value;
                            break;
                        case "x":
                            if (pqReader.IsEmptyElement) break;
                            pqReader.Read();
                            piquet.X = Convert.ToDouble(pqReader.Value.Replace('.', ','));
                            break;
                        case "y":
                            if (pqReader.IsEmptyElement) break;
                            pqReader.Read();
                            piquet.Y = Convert.ToDouble(pqReader.Value.Replace('.', ','));
                            break;
                        case "z":
                            if (pqReader.IsEmptyElement) break;
                            pqReader.Read();
                            piquet.Z = Convert.ToDouble(pqReader.Value.Replace('.', ','));
                            break;
                        case "step":
                            pqReader.Read();
                            piquet.Step = Convert.ToInt32(pqReader.Value);
                            break;
                    }
                }

            }
            else if (pqReader.NodeType == XmlNodeType.EndElement
                && pqReader.Name == "piquet")
            {

```

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		57

```

        _piquetLst.Add(piquet);
    }
} //while(reader.Read())
}
}
}

}
catch (Exception)
{
    MessageBox.Show(Resources.FileReadError, Resources.ETopo, MessageBoxButtons.OK,
        MessageBoxIcon.Error);
}
finally
{
    var min = 0.0;
    var max = 0.0;
    foreach (var piquet in _piquetLst)
    {
        min = Math.Min(min, piquet.Z);
        max = Math.Max(max, piquet.Z);
    }
}
}

private void SaveMenu_Click(object sender, EventArgs e)
{
    if (sdSave.ShowDialog() != DialogResult.OK) return;
    _traceList = new List<Trace>();
    _piquetLst = new List<Piquet>();
    foreach (DataGridViewRow row in dgTopo.Rows)
    {
        if (row.Cells[0].Value == null) continue;
        var trace = new Trace
        {
            From = row.Cells["ClFrom"].Value.ToString(),
            To = row.Cells["ClTo"].Value.ToString(),
            Tape = Convert.ToDouble(row.Cells["ClLen"].Value.ToString())
                .Replace('.', ','),
            Azimuth = Convert.ToDouble(row.Cells["ClAz"].Value.ToString())
                .Replace('.', ','),
            Clino = Convert.ToDouble(row.Cells["ClClino"].Value.ToString())
                .Replace('.', ','),
            Note = row.Cells["ClNote"].Value == null ? "" : row.Cells["ClNote"]
                .Value.ToString()
        };
        var down = row.Cells["ClDown"].Value.ToString().Replace('.', ',');
        var dVals = down.Split('\\');
        if (dVals.Count() == 2)
        {
            trace.FromDown = Convert.ToDouble(dVals[0]);
            trace.Down = Convert.ToDouble(dVals[0]);
        }
        else
        {

```

```

    trace.Down = Convert.ToDouble(down);
}
var up = row.Cells["ClUp"].Value.ToString().Replace('.', ',');
var uVals = up.Split('\\');
if (uVals.Count() == 2)
{
    trace.FromUp = Convert.ToDouble(uVals[0]);
    trace.Up = Convert.ToDouble(uVals[0]);
}
else
{
    trace.Up = Convert.ToDouble(up);
}
var left = row.Cells["ClLeft"].Value.ToString().Replace('.', ',');
var lVals = left.Split('\\');
if (lVals.Count() == 2)
{
    trace.FromLeft = Convert.ToDouble(lVals[0]);
    trace.Left = Convert.ToDouble(lVals[0]);
}
else
{
    trace.Left = Convert.ToDouble(left);
}
var right = row.Cells["ClRight"].Value.ToString().Replace('.', ',');
var rVals = right.Split('\\');
if (rVals.Count() == 2)
{
    trace.FromRight = Convert.ToDouble(rVals[0]);
    trace.Right = Convert.ToDouble(rVals[0]);
}
else
{
    trace.Right = Convert.ToDouble(right);
}
_traceList.Add(trace);
}
var start = new Piquet
{
    Delta = 0D,
    Name = _traceList[0].From,
    X = 0D,
    Y = 0D,
    Z = 0D,
    Step = 0,
    Note = ""
};
_piquetLst.Add(start);
TopoLib.GetTrace(_traceList, _piquetLst, start);
var min = 0.0;
var max = 0.0;
foreach (var piquet in _piquetLst)
{
    min = Math.Min(min, piquet.Z);
    max = Math.Max(max, piquet.Z);
}

```

					230105.000009П.308.ПЗ	Лист
Изм.	Лист	№ докум.	Подп.	Дата		59


```

}

var rings = TopoLib.GetAllRing(_tracelist, _piquetLst, start);

MessageBox.Show(string.Format("Длина: {0:N0}м. Глубина: {1:N0}м. Неточность: {2:F3}%",
    _tracelist.Sum(t => t.Tape), max - min, rings.Count > 0
    ? (rings.Sum(r => r.Offset.Length)/rings.Sum(r => r.Length)*100)
    : 0));

foreach (var ring in rings)
{
    foreach (var point in ring.Points)
    {
        TopoLib.PiquetsCorrection(ring, _tracelist, _piquetLst, point, point.Offset);
    }
}
using (var fl = new FileStream(sdSave.FileName, FileMode.Create))
{
    using (var zipp = new ZipOutputStream(fl))
    {
        zipp.PutNextEntry("trace.xml");
        using (var wr = new MemoryStream())
        {
            var writer = XmlWriter.Create(wr);
            writer.WriteStartDocument();
            writer.WriteStartElement("cave");
            writer.WriteElementString("title", _name);
            writer.WriteStartElement("survey");
            writer.WriteElementString("date", _date);
            writer.WriteStartElement("team");
            foreach (var item in _autor)
            {
                writer.WriteElementString("name", item);
            }
            writer.WriteEndElement();
            foreach (var trace in _tracelist)
            {
                writer.WriteStartElement("segment");
                writer.WriteElementString("from", trace.From);
                writer.WriteElementString("to", trace.To);
                writer.WriteElementString("tape", trace.Tape.ToString());
                writer.WriteElementString("compass", trace.Azimuth.ToString());
                writer.WriteElementString("clino", trace.Clino.ToString());
                writer.WriteElementString("left", trace.Left.ToString());
                writer.WriteElementString("right", trace.Right.ToString());
                writer.WriteElementString("up", trace.Up.ToString());
                writer.WriteElementString("down", trace.Down.ToString());
                if (trace.FromLeft > 0)
                    writer.WriteElementString("f_left", trace.FromLeft.ToString());
                if (trace.FromRight > 0)
                    writer.WriteElementString("f_right", trace.FromRight.ToString());
                if (trace.FromUp > 0)
                    writer.WriteElementString("f_up", trace.FromUp.ToString());
                if (trace.FromDown > 0)

```

```

        writer.WriteElementString("f_down", trace.FromDown.ToString());
        if (!string.IsNullOrEmpty(trace.Note))
            writer.WriteElementString("note", trace.Note);
        writer.WriteEndElement();
    }
    writer.WriteEndElement();
    writer.WriteEndElement();
    writer.WriteEndDocument();
    writer.Flush();
    zipp.Write(wr.ToArray(), 0, (int)wr.Length);
}
zipp.PutNextEntry("piquet.xml");
using (var wr = new MemoryStream())
{
    var writer = XmlWriter.Create(wr);
    writer.WriteStartDocument();
    writer.WriteStartElement("cave");
    foreach (var piquet in _piquetLst)
    {
        writer.WriteStartElement("piquet");
        writer.WriteElementString("name", piquet.Name);
        writer.WriteElementString("x", piquet.X.ToString());
        writer.WriteElementString("y", piquet.Y.ToString());
        writer.WriteElementString("z", piquet.Z.ToString());
        writer.WriteElementString("step", piquet.Step.ToString());
        writer.WriteElementString("note", piquet.Note);
        writer.WriteEndElement();
    }
    writer.WriteEndElement();
    writer.WriteEndDocument();
    writer.Flush();
    zipp.Write(wr.ToArray(), 0, (int)wr.Length);
}
}
}

MessageBox.Show(Resources.Saved);
}

private void dToolStripMenuItem_Click(object sender, EventArgs e)
{
    if(_piquetLst==null||_piquetLst.Count==0||
        _traceList==null||_traceList.Count==0)
        return;
    var fr = new Graph {PqList = _piquetLst, TrcList = _traceList};
    var d = Math.Max(_traceList.Max(t => t.Left), _traceList.Max(t => t.Right));
    fr.top = _piquetLst.Max(p => p.Y)+d;
    fr.bottom = _piquetLst.Min(p => p.Y)-d;
    fr.left = _piquetLst.Min(p=>p.X)-d;
    fr.right = _piquetLst.Max(p=>p.X)+d;
    fr.ShowDialog();
}

private void mData_Click(object sender, EventArgs e)
{

```

```

    var fr = new FrTopoData {name = _name, date = _date, autor = _autor};
    fr.ShowDialog();
    _name = fr.name;
    _date = fr.date;
    _autor = fr.autor;
}

private void ExitMenu_Click(object sender, EventArgs e)
{
    Close();
}
}
}

```

					230105.000009П.308.ПЗ	Лист
						62
Изм.	Лист	№ докум.	Подп.	Дата		