

ОЛИМПИАДА ШКОЛЬНИКОВ «ШАГ В БУДУЩЕЕ»
ЧЕТЫРНАДЦАТАЯ НАУЧНАЯ КОНФЕРЕНЦИЯ МОЛОДЫХ ИССЛЕДОВАТЕЛЕЙ
«ШАГ В БУДУЩЕЕ, МОСКВА»

регистрационный номер

Информатика и системы управления

название факультета

Программное обеспечение ЭВМ и информационные технологии

название кафедры

Путеводитель по зданию

«Нить Ариадны»

название работы

Автор:

Инфлянскас Роман Вацловович

фамилия, имя, отчество

Люберцы, лицей №12, 11А класс

наименование учебного заведения, класс

Научный руководитель:

Трусов Борис Георгиевич

фамилия, имя, отчество

МГТУ им. Н.Э. Баумана

место работы

д.т.н., профессор,

заведующий кафедрой ИУ-7

звание, должность

Москва - 2011

Оглавление

Введение	3
Представление здания в программе	4
Графическое представление здания.....	4
Поиск пути	5
Алгоритм нахождения пути.....	7
Выбор алгоритма	7
Описание алгоритма.....	8
Разработка программы	10
Выбор средств разработки	10
Выбор структуры ПО	10
Диаграмма классов.....	11
Описание пользовательского интерфейса	12
Программа Ариадна	12
Центральная панель	12
Действия.....	13
Создание областей.....	14
Создание дверей.....	16
Создание графа	16
Боковые панели	18
Программа Тесей	23
Заключение	24
Достигнутые результаты	24
Возможности использования	24
Список литературы	25

Введение

*Ковыляющий по прямой дороге опередит бегущего,
который сбился с пути.*

Ф. Бэкон

Проблема поиска пути была актуальна всегда. Но во все времена решалась она по-разному. Так в Древней Греции, согласно мифологии, Тесей для того, чтобы выбраться Лабиринта Минотавра, воспользовался клубком красных ниток, данным ему его возлюбленной Ариадной. А в Китае (предположительно при династии Сун) для определения движения по пустыням был изобретён компас. В эпоху Великих географических открытий стали широко использоваться различные карты. В начале XVIII века появились секстанты, – навигационные измерительные инструменты, используемые для измерения высоты светила над горизонтом, с целью определения географических координат той местности, в которой проводится измерение.

В наше время проблема поиска пути особенно актуальна. Постоянно растет количество автомобилистов, которым нужно ориентироваться на дороге, для чего они используют GPS-навигаторы, которые определяют текущее местоположение транспортного средства и рассчитывают пути до заданной точки. Многие пешеходы пользуются различными картографическими онлайн-сервисами (например, *Яндекс.Карты*, *WikiMapia*, *OpenStreetMap*, *Yahoo! Maps* и др.). Наиболее функциональным из них является сайт *Карты Google*: помимо просмотра спутниковых снимков и карт, пробок, он позволяет прокладывать маршруты для автомобилистов, пользователей общественного транспорта и пешеходов на открытой местности.

Однако проблема ориентирования и поиска пути иногда возникает при перемещении в здании, особенно со сложной конфигурацией. Широко известных программ для зданий не существует.

Было решено реализовать программу-путеводитель по зданию и назвать её *Нить Ариадны*.

Представление здания в программе

Графическое представление здания

Архитектурное сооружение физически является трёхмерной структурой. Однако создание 3D модели здания представляется излишним, так как практика показывает, что человек лучше ориентируется по двумерным картам и планам (именно по этой причине большинство нынешних карт, в том числе и в картографических онлайн-сервисах, двумерны).

Поэтому было решено изображать здание в виде поэтажного плана. Для обозначения на поэтажном плане объектов, находящихся в здании, было решено ввести такое понятие, как область.

Область – это многоугольник, принадлежащий определённому этажу. Если область B лежит внутри области A , то считается, что область A содержит область B , а область B содержится в области A . Область A непосредственно содержит область B , если не существует никакой области C , содержащейся в области A и содержащей область B . Если область A непосредственно содержит область B , то область B непосредственно содержится в области A . Область может непосредственно содержать любое количество областей, но сама может непосредственно содержаться лишь в одной области. Пересечение областей не допускается. Областям сопоставляются текстовые поля: номер, название, описание и надпись. Поле надписи отображается на плане этажа.

На границе области может лежать любое количество дверей. Так как размеры двери невелики, её можно принять за точку (на плане этажа представляется в виде окружности небольшого радиуса).

Приведём примеры областей: кабинет, отдел, в котором содержится несколько кабинетов, банкомат, лифтовый холл, лестничная площадка. Отметим, что понятие дверь в некоторых случаях используется не в прямом значении: например, для того чтобы указать окно бюро пропусков, обозначить место входа на лестницу.

Поиск пути

Одной из основных задач путеводителя по зданию является поиск пути. Для реализации данной возможности было решено добавить в поэтажный план здания смешанный взвешенный граф, обладающий данными свойствами:

1. Вершины представляют собой точки на поэтажном плане.
2. Часть вершин графа связана с дверьми (их координаты равны).
3. Минимальная степень прочих (*свободных*) вершин равна двум.
4. Веса всех ребер положительны.

Граф задаётся пользователем. Все рёбра, обе вершины которых лежат на одном этаже, ненаправлены, вес каждого ребра равен расстоянию между вершинами этого ребра.

Нельзя рассматривать этажи здания отдельно, ведь в таком случае невозможно прокладывать пути, проходящие по нескольким этажам. Для того чтобы связать области, представляющие собой лестницы (или лифты), находящиеся на разных этажах, введём понятие вертикаль. Вертикаль – объединение областей, находящихся на различных этажах. В вертикали может содержаться только одна область от каждого этажа. Так как в зданиях для перемещения между этажами наиболее часто используются лестницы и лифты (в некоторых зданиях, например в торговых центрах, могут использоваться эскалаторы), было решено присваивать вертикалям один из типов: лестница, лифт или аудитория (см. ниже). Поскольку обычно лестницы и лифты имеют единственное место входа на данном этаже, будем считать, что их области содержат только одну дверь. Если области $A_1, A_2, A_3, \dots, A_{k-1}, A_k$, упорядоченные по возрастанию номера этажа, объединены в вертикаль типа лестница или лифт, то в графе здания есть ориентированные рёбра $(N_1, N_2), (N_2, N_1), (N_2, N_3), (N_3, N_2), \dots, (N_{k-1}, N_k), (N_k, N_{k-1})$, где N_i – вершина, связанная с дверью области A_i . Вес ребра (N_i, N_{i+1}) равен расстоянию, соответствующему подъёму с этажа, на котором расположена область A_i , на этаж, на котором расположена область A_{i+1} , а вес ребра (N_{i+1}, N_i) равен расстоянию, соответствующему спуску с этажа, на котором расположена область A_{i+1} , на этаж, на котором расположена область A_i :

$$S = vnt,$$

где v – средняя скорость пешехода на горизонтальной поверхности (5 км/ч), S – расстояние, соответствующее подъёму (спуску) на n этажей за t – время, требуемое на подъём (спуск) на один этаж.

Однако помимо лестниц и лифтов в некоторых зданиях, например в УЛК МГТУ им. Н.Э. Баумана, есть двухэтажные аудитории. Для того чтобы объединить части аудитории, расположенные на разных этажах, требуется ввести тип вертикали: аудитория. В отличие от вертикалей типа лестница и лифт, при объединении областей в вертикаль дополнительные рёбра в граф не добавляются, ведь нет необходимости прокладывать пути через аудитории.

Алгоритм нахождения пути

– Не подскажите, как пройти в библиотеку?

– В три часа ночи?!

Фильм «Операция “Ы”»

Выбор алгоритма

Пусть дан взвешенный граф $G = (V, E)$, причем $V < E$. Для определенности будем считать, что граф является связным (обычно в здании не существует недостижимых участков).

Было решено реализовать поиск пути, при котором в качестве начального и конечного объектов могут выступать область, дверь и вершина графа. Поиск пути до (от) двери равносильно поиску пути до (от) вершины, связанной с ней. Поиск до (от) области можно определить как поиск пути до (от) всех дверей, принадлежащих области. Таким образом, поиск пути должен вестись от некоторого количества (N) начальных вершин до некоторого количества (M) конечных вершин, где N и M невелики ($1 \leq N, M \ll V$), после чего из полученных NM путей должен быть отобран путь наименьшей длины. Поскольку нет общеизвестного алгоритма, решающего эту задачу, было решено реализовать алгоритм путем изменения уже существующих, которые можно разделить на две группы:

1. Алгоритмы, решающие задачу поиска кратчайших путей из одной вершины до всех остальных (1 стр. 479-508).
2. Алгоритмы, ищущие кратчайшие пути для всех пар вершин (1 стр. 509-534).

К алгоритмам первой группы относятся алгоритмы Дейкстры (стоимость $O(V^2)$) и Беллмана-Форда (стоимость $O(VE)$); в отличие от алгоритма Дейкстры, граф может иметь рёбра отрицательного веса). Их можно преобразовать к необходимому алгоритму путём добавления внешнего цикла, в котором будет происходить перебор по N начальным вершинам. В данном случае стоимость полученного алгоритма будет в N раз больше стоимости исходного.

К алгоритмам второй группы относятся алгоритмы Флойда-Уоршола (стоимость $O(V^3)$) и Джонсона (стоимость $O(V^2 \log V + VE)$). В обоих алгоритмах граф может иметь рёбра отрицательного веса. При их модификации стоимость не изменится, поскольку множество полученных путей включает в себя множество искомым.

В данном случае для модификации более всего подходит алгоритм Дейкстры (1 стр. 489-493) по следующим причинам:

- При изменении данного алгоритма будет достигнута высокая скорость работы при заданных ограничениях: стоимость алгоритма будет $O(NV^2)$, что меньше, чем при модификации алгоритма Беллмана-Форда (стоимость которого $O(NVE)$), Флойда-Уоршола ($O(V^3)$) и Джонсона ($O(V^2 \log V + VE)$).
- Он наиболее прост в реализации из всех вышеперечисленных.

Описание алгоритма

Процедура FindWay находит кратчайший путь от множества начальных до множества конечных вершин. Результатом работы алгоритма является массив *way*, который содержит вершины кратчайшего пути, следующие от конечной к начальной.

В процедуре FindWay внутри цикла, в котором осуществляется перебор по всем начальным вершинам, происходит вызов процедуры Dijkstra, в которую передаются: текущая начальная вершина, множество конечных вершин и текущее значение длины пути, наименьшей длины.

В процедуре Dijkstra поддерживается множество L вершин, для которых уже найден кратчайший путь. Алгоритм выбирает вершину $c \notin L$ с наименьшим $d[c]$, добавляет её во множество L и производит релаксацию (1 стр. 483-489) для всех рёбер, выходящих из c , после чего цикл повторяется.

Приведём исходный код алгоритма на псевдокоде (1 стр. 20-21). Дополнительно отметим, что если какой-то элемент массива не был инициализирован, то он считается несуществующим (проверка на существование выполняется следующим образом: `IsExist(Array, index)`).

Dijkstra(s, F, min)

```

1    $L \leftarrow \emptyset$ 
2    $d[0] \leftarrow \infty$ 
3    $d[s] \leftarrow 0$ 
4    $c \leftarrow s$ 
5   while  $c \neq 0$  do
6        $c \leftarrow 0$ 
7       for (для) всех элементов  $d[v]$  do
8           if  $v \notin V$  then
9               if  $d[v] < d[c]$  then
```



```

10           $c \leftarrow v$ 
11      if  $c \neq 0$  then
12           $L \leftarrow L \cup c$ 
13          for (для) всех вершин  $v$ , смежных с  $c$  do
14               $distThroughC \leftarrow d[c] + \text{arc}(c, v)$ 
15               $update \leftarrow \text{FALSE}$ 
16              if  $\text{IsExist}(d, v)$  then
17                   $update \leftarrow (dist < d[v])$ 
18              else
19                   $update \leftarrow \text{TRUE}$ 
20              if  $update$  then
21                   $d[v] \leftarrow dist$ 
22                   $\pi[v] \leftarrow c$ 
23   $c \leftarrow 0$ 
24  for (для) всех вершин  $v \in F$  do
25      if  $d[v] < c$  then
26           $c \leftarrow d[v]$ 
27  if  $d[c] < min$  then
28       $way \leftarrow 0$ 
29       $i \leftarrow 0$ 
30      while  $c \neq 0$  do
31           $way[i] \leftarrow c$ 
32           $c \leftarrow \pi[c]$ 
33           $i \leftarrow i + 1$ 
34      return  $d[c]$ 
35  else
36      return  $min$ 

```

FindWay(S, F)

```

1       $min \leftarrow \infty$ 
2      for (для) всех вершин  $v \in S$  do
3           $min \leftarrow \text{Dijkstra}(v, F, min)$ 

```

Разработка программы

Выбор средств разработки

В качестве языка программирования для реализации ПО был выбран C++. Язык был выбран в связи с тем, что он позволяет эффективно реализовывать алгоритмы, обеспечивает необходимую функциональность и читаемость кода программы. Программы, написанные на C++, работают быстрее, чем, например, программы, написанные на Delphi, C# или Java. В качестве набора библиотек был выбран Qt, так как он позволяет создавать кроссплатформенное ПО.

В качестве комплекта средств разработки был использован Qt SDK, который специально создан для программирования на Qt. Входящая в его состав среда разработки Qt Creator предоставляет удобный интерфейс для работы с кодом, контекстно-зависимую систему справки и интегрированный отладчик.

Выбор структуры ПО

Поскольку чаще всего пользователю не нужна возможность редактирования поэтажного плана здания (аналогично картографическим онлайн-сервисам: создание карты производится заранее, пользователь не имеет средств для её изменения), было решено разбить путеводитель по зданию *Нить Ариадны* на две независимые программы: *Ариадна* и *Тесей*. Программа *Ариадна* предназначена для создания пользователем и отображения поэтажного плана здания, поиска нужных областей и нахождения между объектами кратчайших путей. Программа *Тесей* является сокращённой версией программы *Ариадна*, в которой исключена функция изменения поэтажного плана здания (см. Таблица 1).

Таблица 1 – сравнение программ *Тесей* и *Ариадна*

	Тесей	Ариадна
Просмотр поэтажного плана здания	✓	✓
Поиск областей	✓	✓
Нахождение кратчайших путей	✓	✓
Редактирование поэтажного плана здания	✗	✓

Диаграмма классов

Ниже приведена диаграмма классов (см. Рисунок 1) путеводаителя по зданию *Нить Ариадны*.

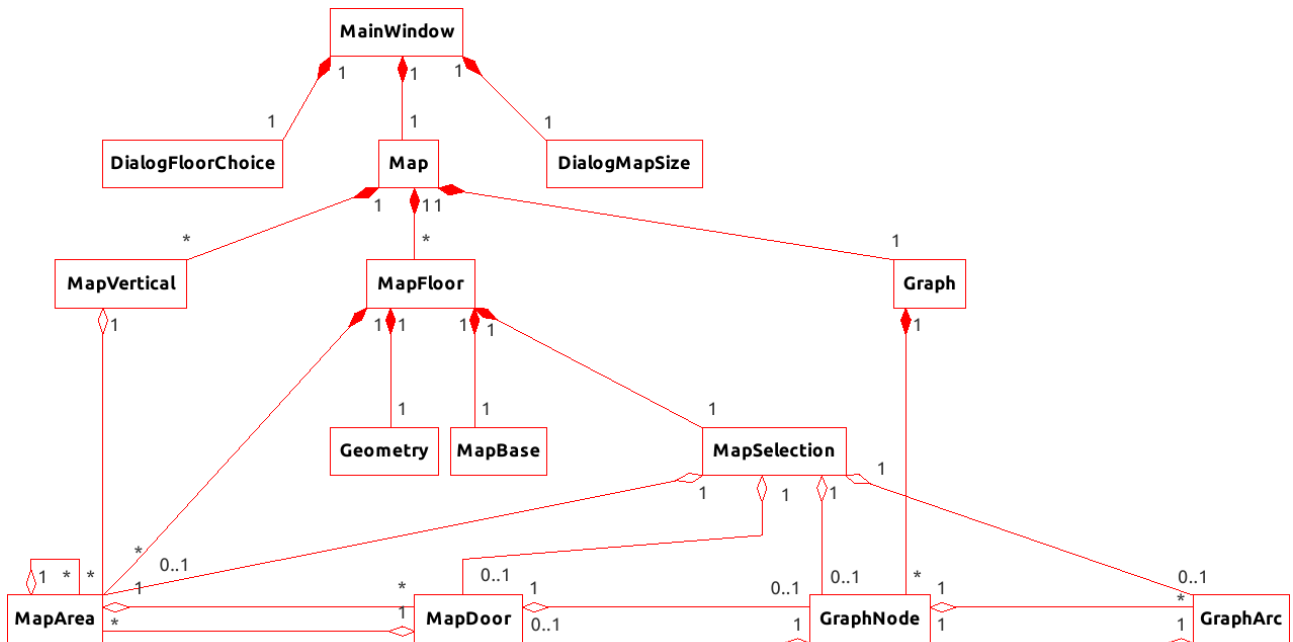


Рисунок 1 – диаграмма классов.

Описание пользовательского интерфейса

Кто так строит, а? Кто так строит?
С.Л. Фарада, выпускник МВТУ им. Н.Э. Баумана

Программа Ариадна

Основную часть окна программы *Ариадна* (см. Рисунок 2) занимает центральная панель. В верхней части окна находится главное меню и панели инструментов *Файл*, *Вид*, *Перейти* и *Добавить*. Слева располагается панель инструментов *Панели*, при помощи которой осуществляется управление видимостью боковых панелей: *Этажи*, *Вертикали*, *Свойства области*, *Поиск* и *Пути*. При отображении боковой панели, видимая панель (если таковая имеется) скрывается.

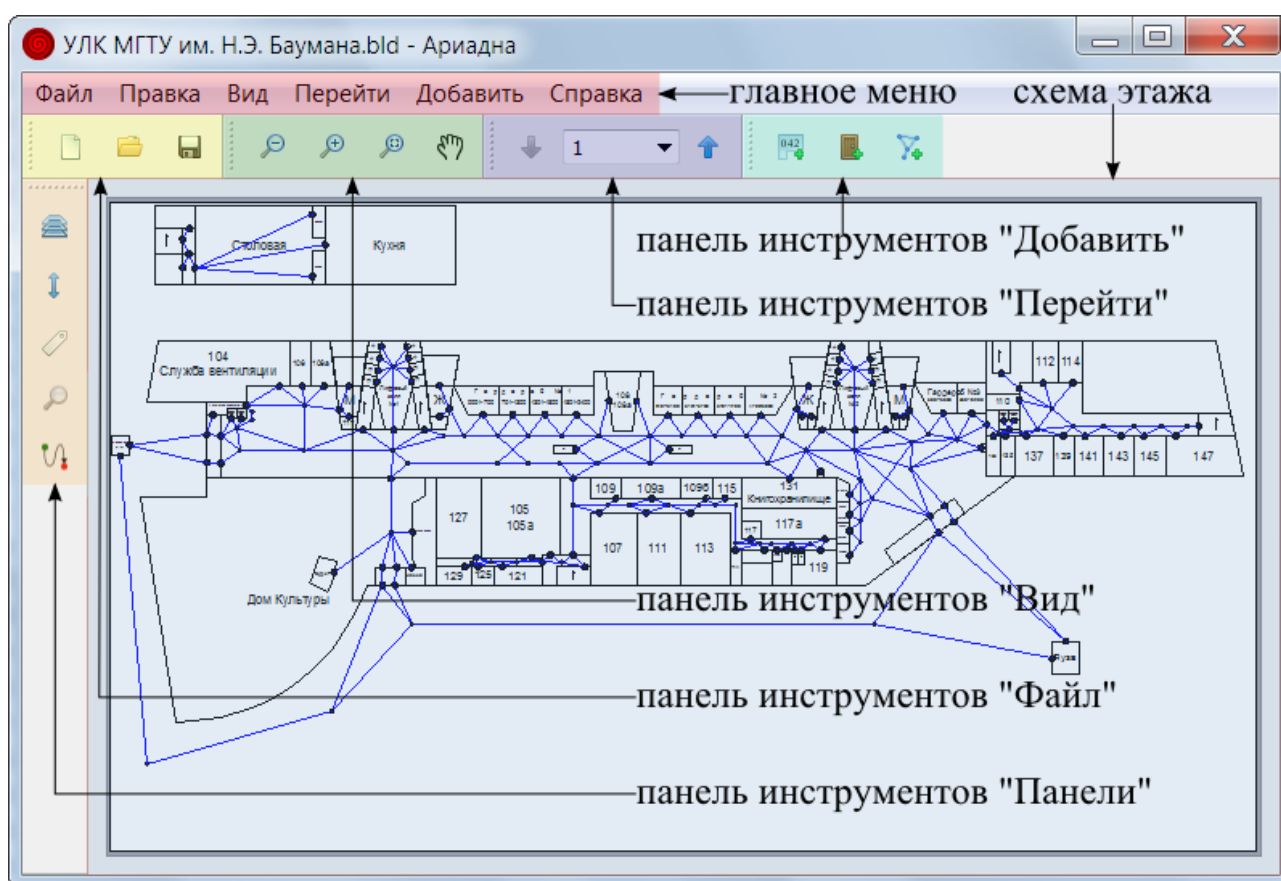


Рисунок 2 – главное окно программы *Ариадна*.

Центральная панель

На центральной панели этажа могут отображаться слои:

1. *План этажа*. Отображается по умолчанию, не может быть скрыт.
2. *Подложка*. В качестве подложки может быть выбран файл графического формата JPEG, BMP и PNG. Отображается по умолчанию.

3. *Сетка*. Не отображается по умолчанию.











4. *Граф*. Отображается по умолчанию.


















На каждом этаже может быть выделен один объект. При нажатии клавиши Delete выделенный объект будет удалён. Если выделенный объект является областью, то возможен ввод информации об этой области. При выделении другого объекта или при щелчке на пустом месте выделенный объект теряет выделение.

Действия


Ниже (см. Таблица 2) приведены действия, которые может осуществлять пользователь с помощью главного меню, панелей инструментов и горячих клавиш. При переходе в режим создания областей, дверей или графа некоторые действия становятся недоступны.



Таблица 2 – действия.

Действие	Сочетание клавиш	Описание
 Создать...	Ctrl+N	Создание нового поэтажного плана здания.
 Открыть...	Ctrl+O	Открытие существующего поэтажного плана здания.
 Сохранить	Ctrl+S	Сохранение текущего поэтажного плана здания.
 Сохранить как...		Сохранение текущего поэтажного плана здания как нового поэтажного плана здания. Последующие изменения вносятся в новый поэтажный план здания.
 Выход	Ctrl+Q	Завершение работы с программой. В случае, если открытый файл был изменён, но не был сохранён, показывается предложение сохранить поэтажный план здания.
 Копировать область...	Ctrl+C	Копирование выделенной области со всеми внутренними областями и дверьми на другой этаж с сохранением местоположения на плоскости.
 Копировать граф...	Ctrl+Shift+C	Копирование графа части графа, находящегося на данном этаже, на другой этаж.
 Добавить подложку		Добавление подложки для данного этажа.
 Слой «Подложка»	Ctrl+B	Управление видимостью слоя «Подложка».
 Слой «Сетка»	Ctrl+#	Управление видимостью слоя «Сетка».
 Слой «Граф»		Управление видимостью слоя «Граф».
 Уменьшить	Ctrl+-	Уменьшение масштаба центральной панели.

 Увеличить	Ctrl++	Увеличение масштаба центральной панели.
 Вместить	Ctrl+*	Изменение масштаба центральной панели так, чтобы план этажа целиком поместился на экране.
 Панорамирование	Ctrl+H	Включение/выключение панорамирования.
 Прилипнуть к продолжениям отрезков	F5	Включение/выключение прилипания курсора рисования к продолжениям отрезков областей.
 Этаж выше	Page Up	Отображение на экране этажа, находящегося выше текущего.
 Этаж ниже	Page Down	Отображение на экране этажа, находящегося ниже текущего.
 Самый верх	Home	Отображение на экране верхнего этажа.
 Самый низ	End	Отображение на экране нижнего этажа.
 Создание областей	Ctrl+A	Переключение в режим создания областей.
 Создание дверей	Ctrl+D	Переключение в режим создания дверей.
 Создание графа	Ctrl+G	Переключение в режим создания графа.
 О программе		Отображение информации о программе.
 Этажи	Ctrl+1	Управление видимостью боковой панели <i>Этажи</i>
 Вертикали	Ctrl+2	Управление видимостью боковой панели <i>Вертикали</i>
 Свойства области	Ctrl+3	Управление видимостью боковой панели <i>Свойства области</i>
 Поиск	Ctrl+4	Управление видимостью боковой панели <i>Поиск</i>
 Пути	Ctrl+5	Управление видимостью боковой панели <i>Пути</i>

Создание областей

При переходе в режим создания областей, дверей и графа курсор меняет свой вид на перекрестие, возникает курсор рисования в виде окружности. Обычно центр окружности (точка ) совпадает с курсором.

Рисование области осуществляется посредством обозначения вершин многоугольника. Первая вершина помечается красным цветом. При перемещении окружности рисуется отрезок, соединяющий последнюю добавленную вершину и точку . При щелчке левой клавишей мыши происходит добавление вершины (точки ) в область. Для того чтобы удалить последнюю добавленную вершину, требуется нажать клавишу Delete. Чтобы замкнуть первую точку с последней и завершить область, необходимо нажать клавишу Enter или дважды

щёлкнуть мышью. После этого начинается создание следующей области. Для того чтобы выйти из режима создания областей, не завершая область, требуется нажать клавишу Esc.

Так как большинство отрезков в областях должны быть вертикальны или горизонтальны, при создании области рисуются только такие отрезки. Чтобы нарисовать произвольный отрезок, необходимо нажать и удерживать клавишу Ctrl.

В подавляющем числе случаев области соприкасаются сторонами, поэтому курсор рисования прилипает к отрезкам и вершинам областей, содержащихся на плане текущего этажа, а также к горизонтальной или вертикальной прямой, проходящим через первую вершину создаваемой области (см. Рисунок 3). Для того чтобы временно отключить данное поведение, необходимо нажать и удерживать клавишу Shift. Чтобы отключить прилипание только к вершинам областей, необходимо нажать и удерживать клавишу Alt. Иногда встаёт необходимость прилипнуть не только к отрезкам областей, но и к их продолжениям. Для включения/выключения данного режима было решено использовать одиночное нажатие клавиши F5 (или выбор пункта меню *Прилипнуть к продолжениям отрезков*).

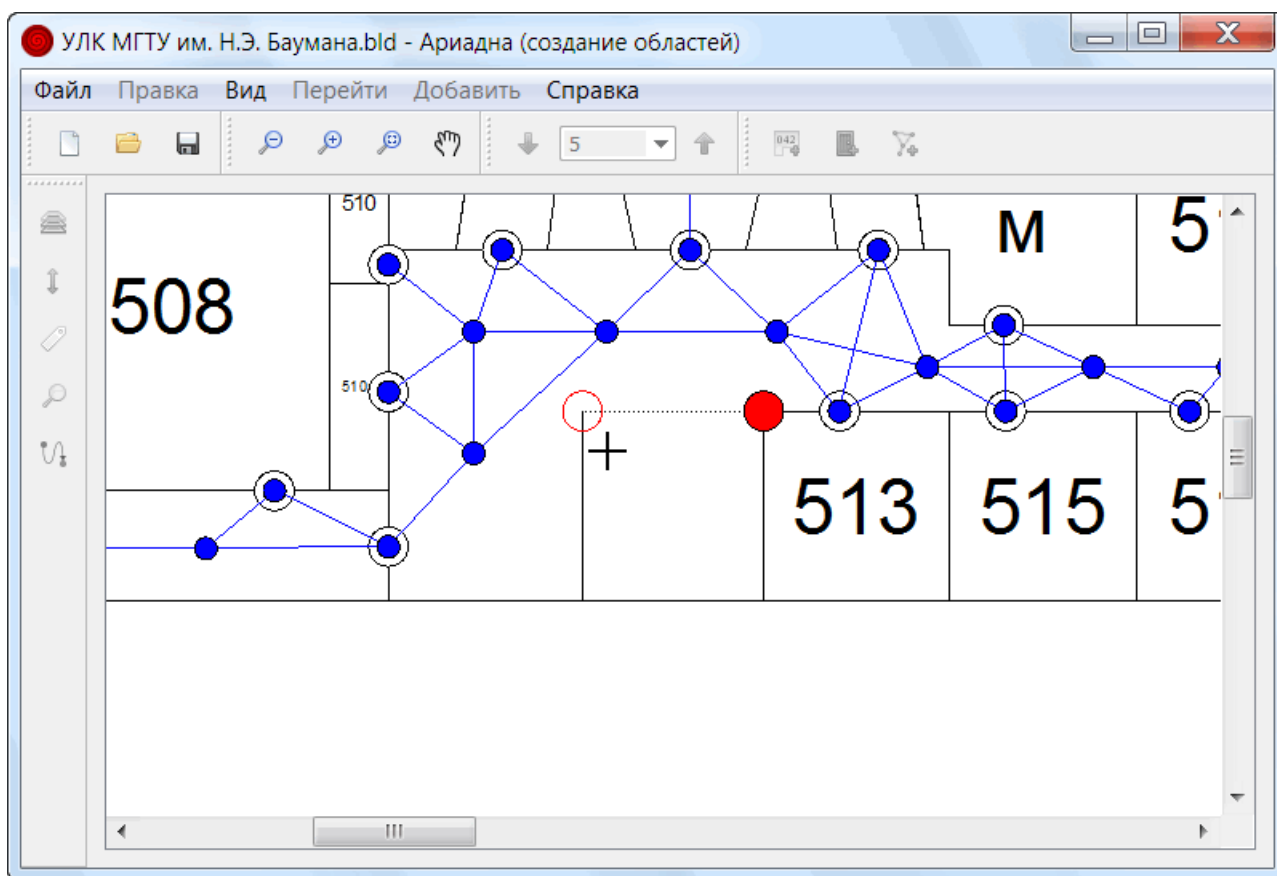


Рисунок 3 – создание областей.

Создание дверей

Для того чтобы поставить дверь, нужно навести курсор рисования на место, в котором должна быть поставлена дверь, и щёлкнуть левой кнопкой мыши или нажать клавишу Enter. Двери ставятся только на контуры областей, к которым они прилипают (см. Рисунок 4). Для того чтобы выйти из режима создания дверей, требуется нажать клавишу Esc.

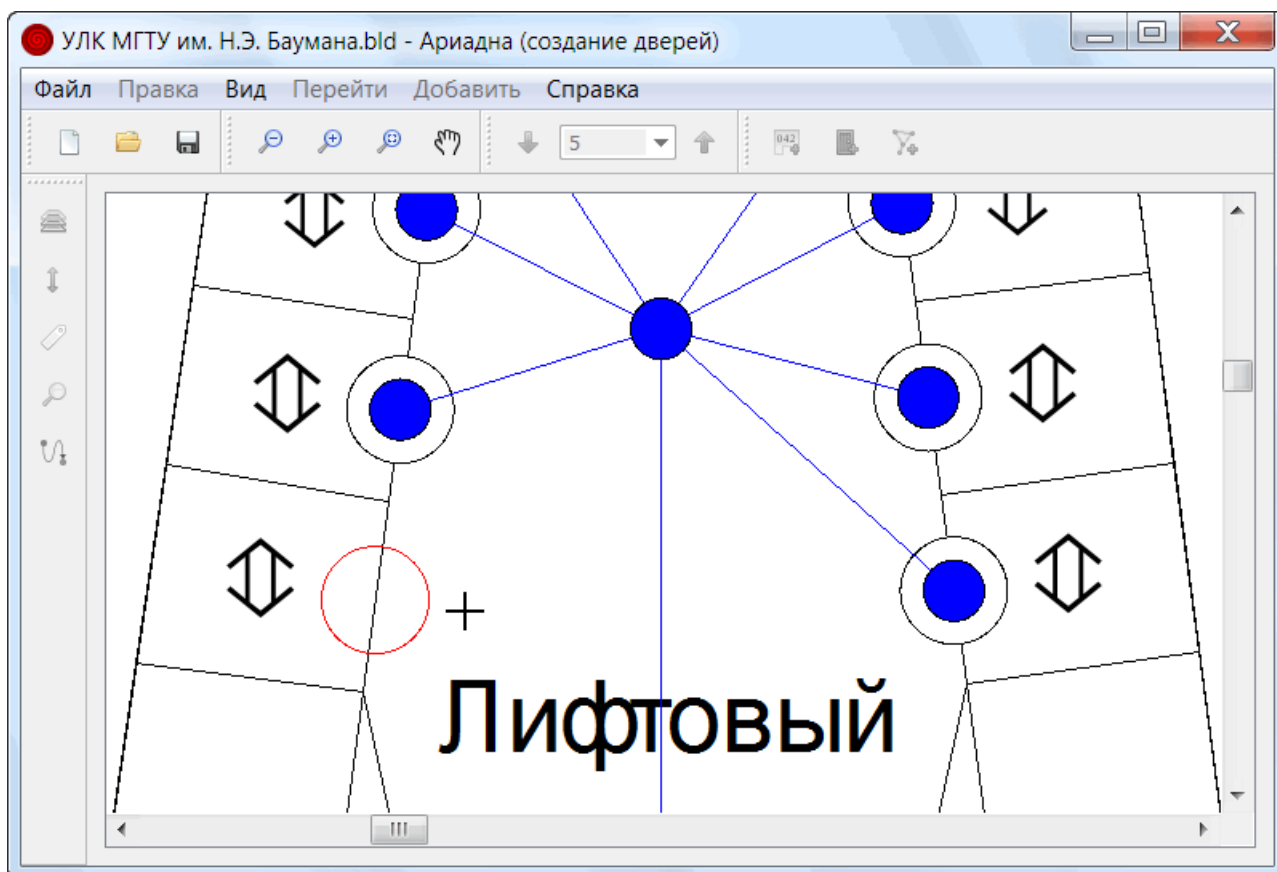


Рисунок 4 – создание дверей.

Создание графа

Создание графа производится путём последовательного добавления вершин графа. Для создания вершины графа нужно навести курсор рисования на место, в котором она должна быть поставлена, и щёлкнуть левой клавишей мыши или нажать клавишу Enter. Если в данном месте уже была вершина графа, то добавление новой вершины не произойдёт, однако эта вершина графа будет считаться последней добавленной вершиной. После этого будет создано ребро, инцидентное предпоследней и последней добавленным вершинам. Для того чтобы удалить последнюю добавленную вершину, требуется нажать клавишу Delete.

При создании графа курсор рисования движется так, чтобы все образующиеся рёбра были вертикальными или горизонтальными. Чтобы временно отключить данное поведение, необходимо нажать и удерживать клавишу Ctrl.

Также во время создания графа курсор рисования липнет ко всем дверям и вершинам графа (см. Рисунок 5). Чтобы временно отключить это поведение, нажмите и удерживайте клавишу Shift. Если необходимо прилипнуть к горизонтальным и вертикальным прямым, проходящим через вершины графа, требуется нажать клавишу F5.

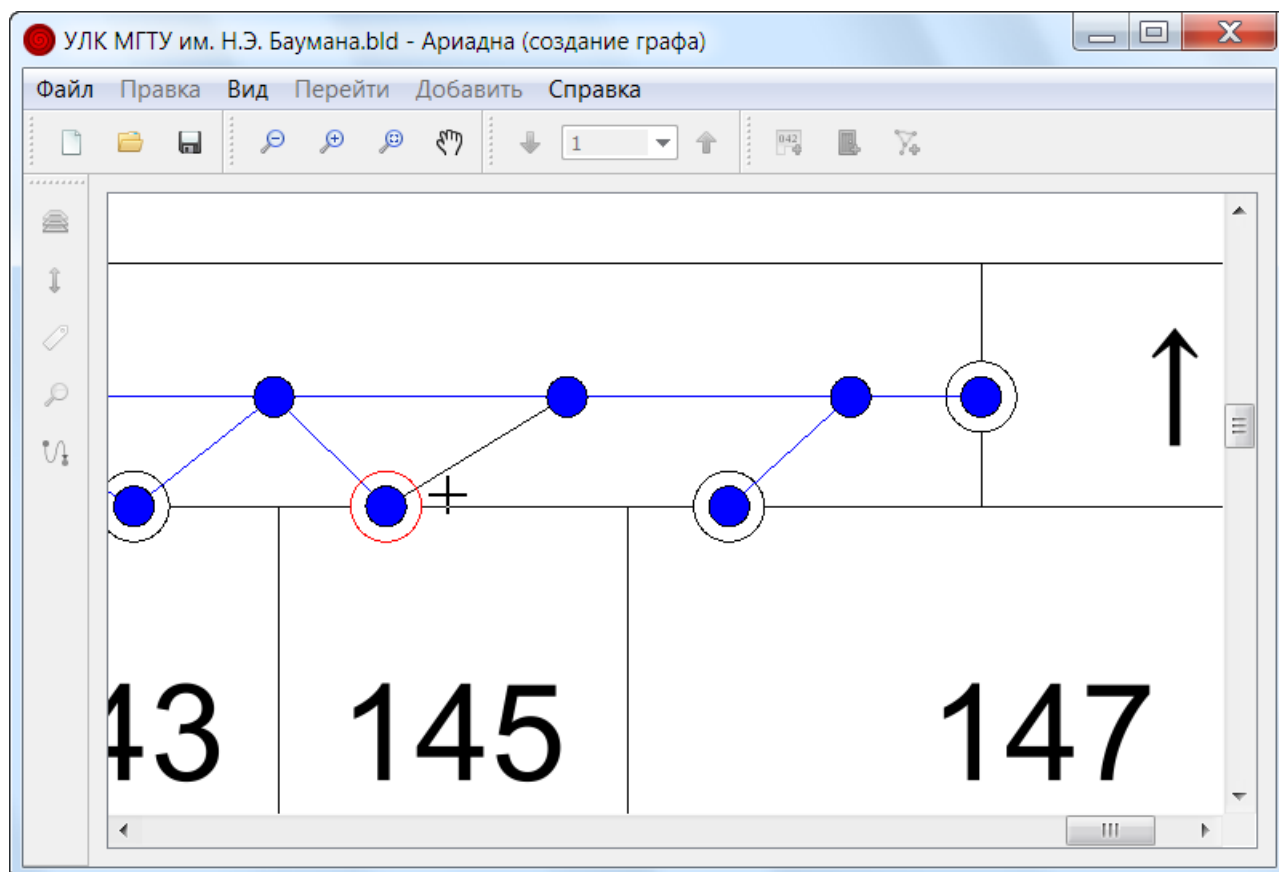


Рисунок 5 – создание графа.

Боковые панели

Этажи

Боковая панель *Этажи* (см. Рисунок 6) служит для создания, удаления, изменения порядка этажей. Также возможно задание этажа, отображаемого при открытии поэтажного плана здания. Все действия производятся над текущим этажом (выделенным в списке и отображаемом в области плана этажа). Управляющие кнопки находятся над списком этажей, в котором этажи расположены в естественном порядке (в начале (вверху) списка расположены этажи, верхние физически, в конце – нижние этажи).

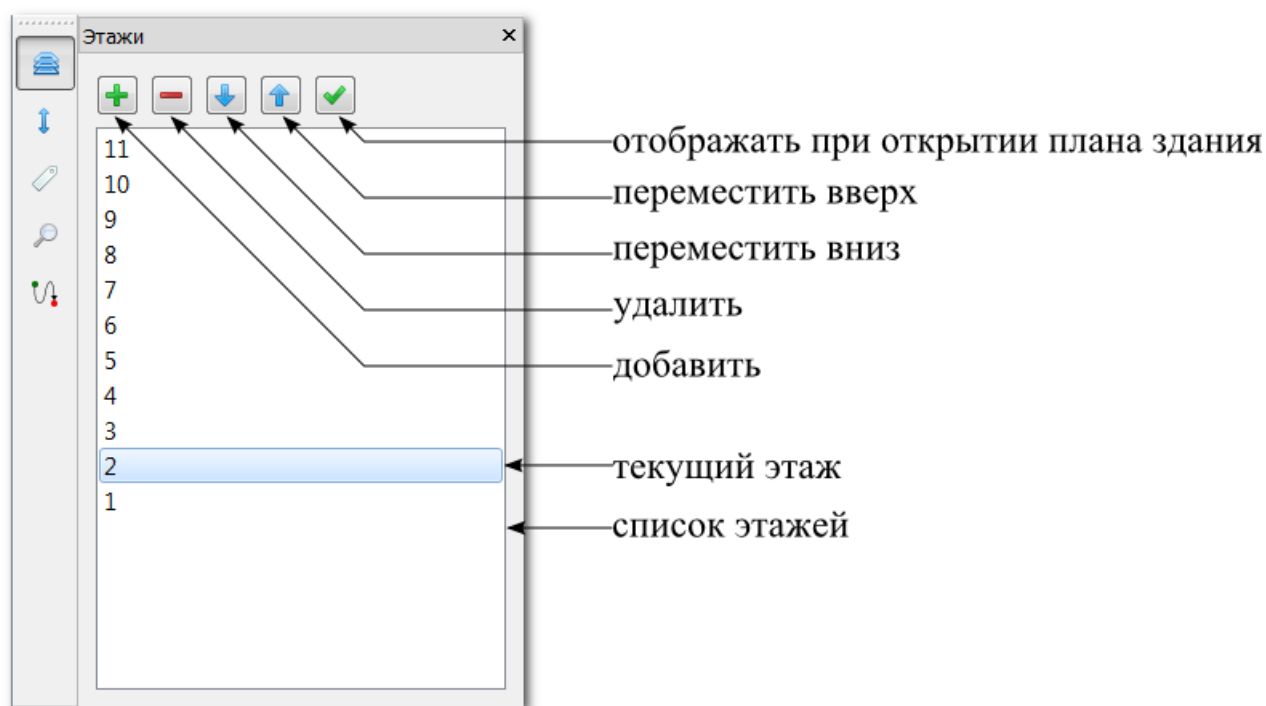


Рисунок 6 – боковая панель *Этажи*.

Вертикали

Боковая панель *Вертикали* (см. Рисунок 7) служит для создания, удаления, изменения порядка вертикалей. Для указания типа вертикали используется выпадающий список. Управляющие кнопки находятся над списком вертикалей, в котором, в отличие от списка этажей, вертикали расположены в произвольном порядке.

При выделении вертикали в списке на поэтажном плане выделяются все области, входящие в её состав. После этого возможно добавление и удаление областей из вертикалей. Для того чтобы добавить область в вертикаль, необходимо активировать этаж, содержащий область, и выделить область. Для того чтобы удалить область из вертикали, необходимо снять выделение с нужной области.

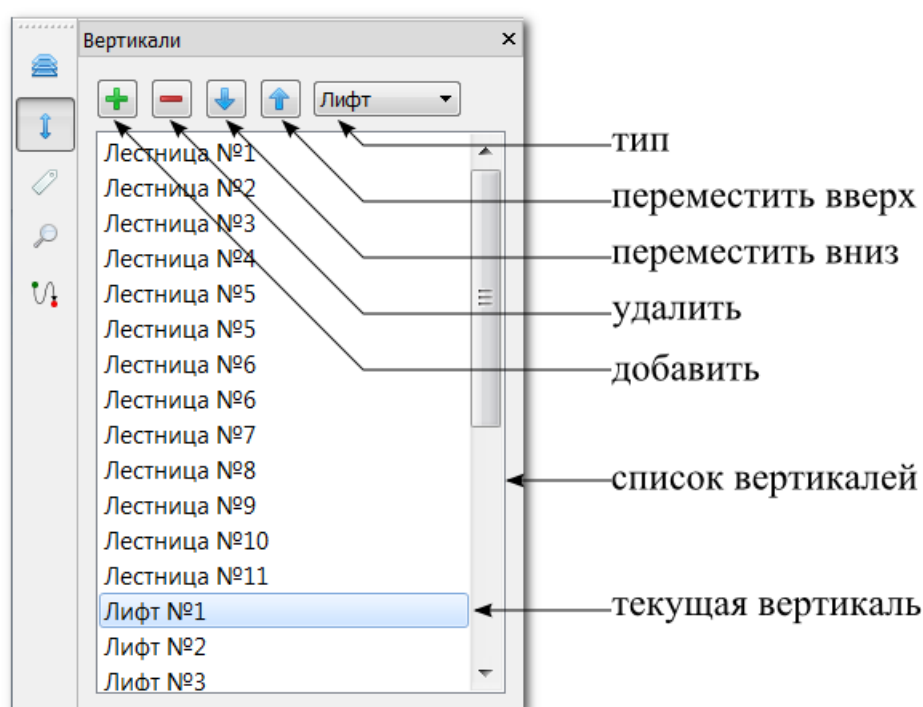


Рисунок 7 – боковая панель *Лестницы*

Свойства области

Боковая панель *Свойства области* (см. Рисунок 8) служит для внесения пользователем номера, названия, описания и надписи выделенной области. Все поля являются текстовыми; *Номер* и *Название* являются однострочными, а *Описание* и *Надпись* – многострочными полями. Для того чтобы ускорить процесс набора надписи, при первом вводе свойств области или при нажатии кнопки *Обновить надпись*, все введенные поля склеиваются в поле *Надпись*.

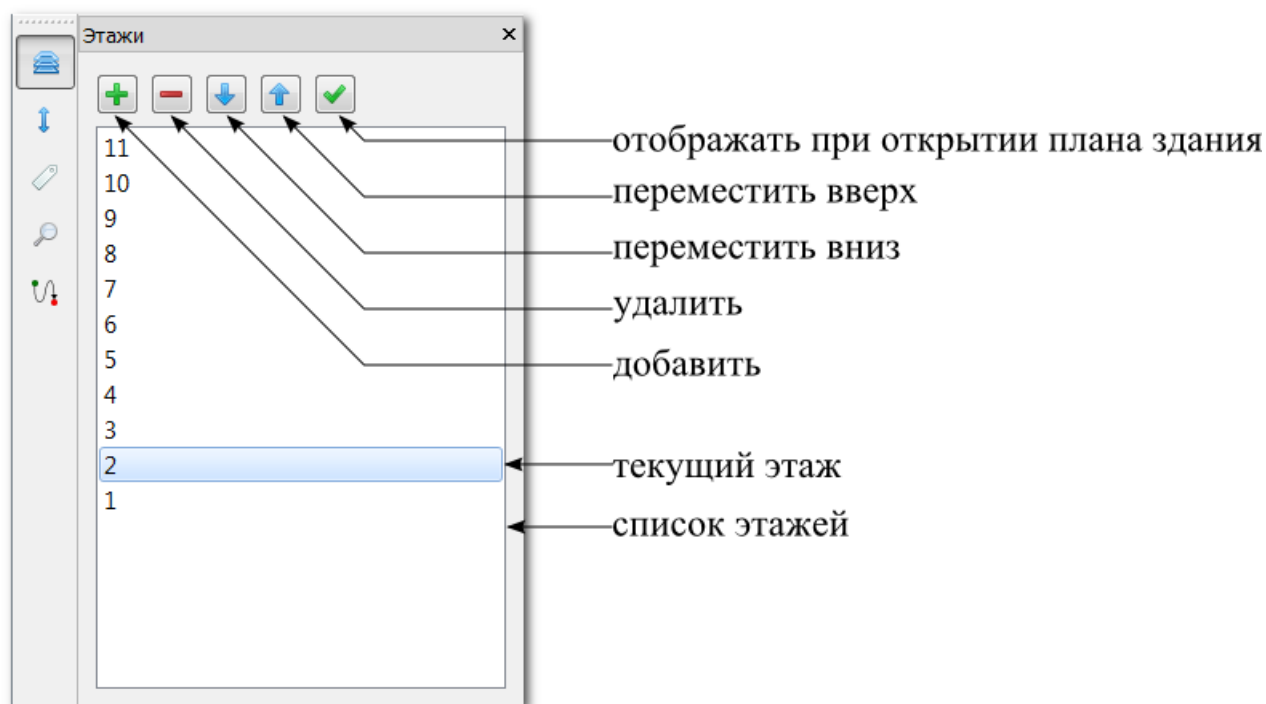


Рисунок 8 – боковая панель *Свойства области*

Поиск

Боковая панель *Поиск* (см. Рисунок 9) служит поиска областей на поэтажном плане здания. Поиск ведётся по мере ввода пользователем запроса (возможен запрос с использованием регулярных выражений). Под полем для ввода отображаются результаты поиска. При щелчке по результату на центральной панели отображается и выделяется данная область.

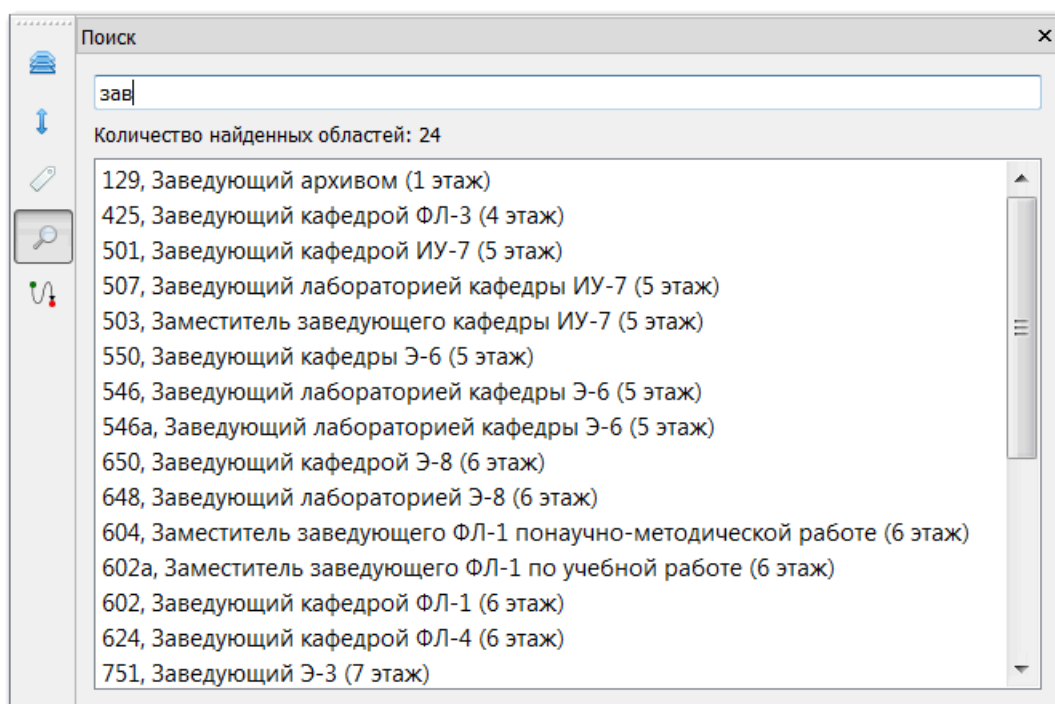


Рисунок 9 – боковая панель *Поиск*.

Пути

Боковая панель *Пути* (см. Рисунок 10) служит поиска пользователем кратчайших путей. При помощи кнопок *Старт* и *Финиш* пользователь устанавливает начальный (выделяемый зелёным цветом) и конечный (выделяемый красным цветом) объекты пути. После этого снизу отображается список возможных вариантов путей. При щелчке на варианте путь отображается на карте зелёным цветом.

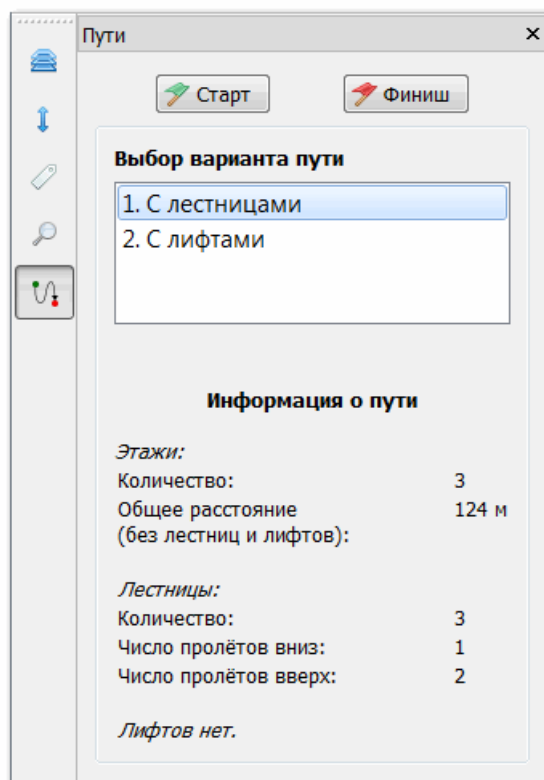


Рисунок 10 –боковая панель Пути.

Программа Тесей

Поскольку программа *Тесей* является урезанной версией программы *Ариадна*, не будем приводить описание пользовательского интерфейса программы. Ниже (см. Рисунок 11) приведено главное окно программы *Тесей*.

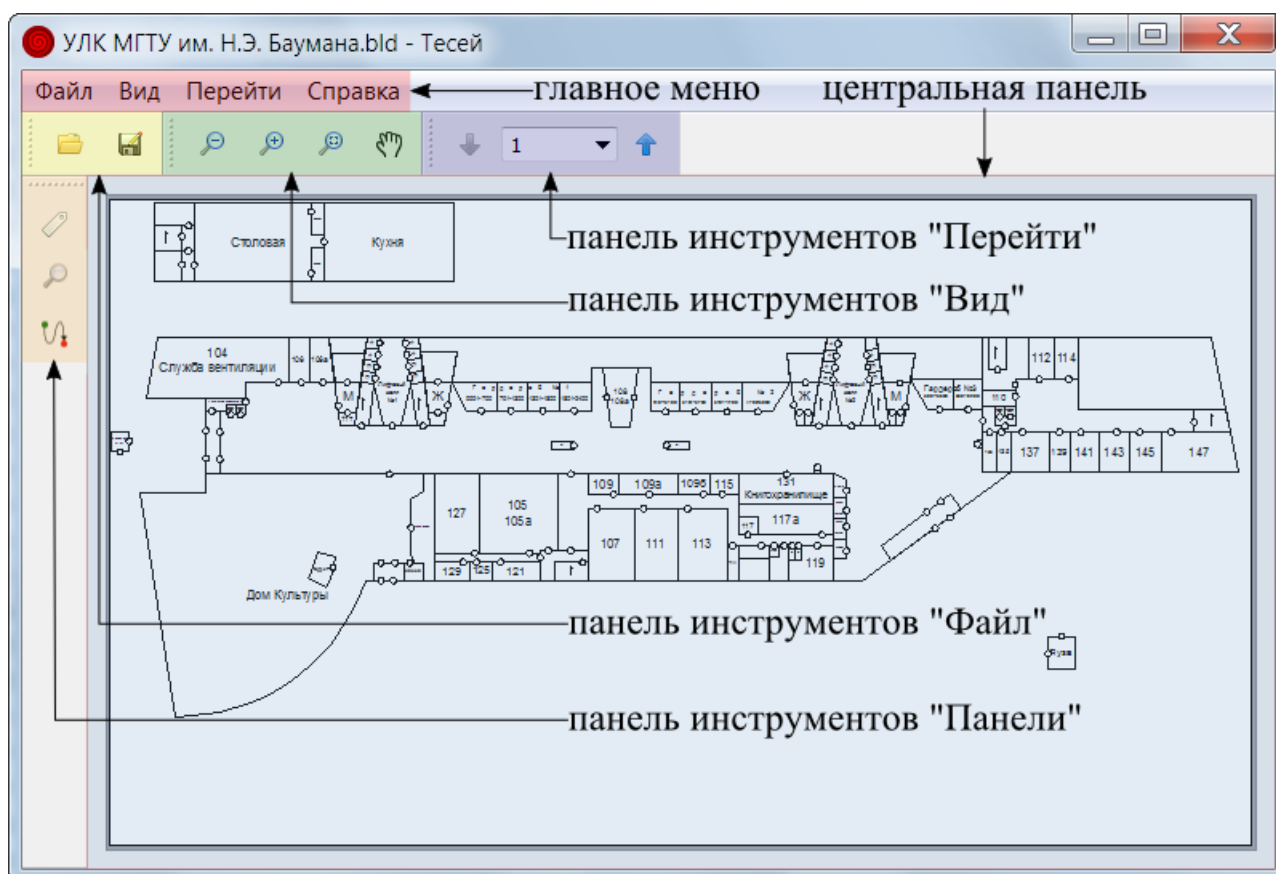


Рисунок 11 – главное окно программы *Тесей*.

Заключение

Клянусь найти аудиторию 501ю... в двухмесячный срок
Клятва бауманца

Достигнутые результаты

В процессе работы были достигнуты следующие результаты:

- проанализированы алгоритмы для нахождения кратчайших путей;
- показана актуальность разработки путеводителя по зданию;
- разработан путеводитель по зданию *Нить Ариадны*, состоящий из двух независимых программ: *Ариадна* и *Тесей*. Программа *Ариадна* предназначена для создания пользователем и отображения поэтажного плана здания, поиска нужных областей и нахождения между объектами кратчайших путей. Программа *Тесей* является сокращённой версией программы *Ариадна*, в которой исключена функция изменения поэтажного плана здания;
- созданы файлы зданий учебно-лабораторного корпуса МГТУ им. Н.Э. Баумана и Государственного Музея Изобразительных Искусств им. А.С. Пушкина для демонстрации работы путеводителя.

Возможности использования

Возможны следующие варианты использования путеводителя по зданию *Нить Ариадны*:

- создание поэтажных планов различных зданий;
- создание поэтажных планов всех строений МГТУ им. Н.Э. Баумана;
- установка программы *Тесей* на терминалы, расположенные в МГТУ им. Н.Э. Баумана.

Список литературы

1. **Кормен, Т., Лейзерсон, Ч. и Ривест, Р.** *Алгоритмы: построение и анализ*. М. : МЦНМО: БИНОМ. Лаборатория знаний, 2004.
2. **Кун, Н. А.** *Легенды и мифы Древней Греции*. М. : Просвещение, 1975.
3. **Корн, Г. и Корн, Т.** *Справочник по математике (для научных сотрудников и инженеров)*. М. : Наука, 1978.
4. **Дейтел, Х. М. и Дейтел, П. Дж.** *Как программировать на C++*. М. : ООО "Бином-Пресс", 2007.
5. **Шилдт, Г.** *Полный справочник по C++*. М. : Издательский дом "Вильямс", 2007.
6. **Шлее, М.** *Qt4.5. Профессиональное программирование на C++*. СПб. : БВХ-Петербург, 2010.