**Слайд 1.** Здравствуйте, меня зовут Горшков Никита Александрович, и я хочу представить Вам свою квалификационную работу бакалавра на тему «Метод построения велосипедных маршрутов».

В последнее время популярность использования велосипедов растет. Поэтому многие велосипедисты нуждаются в сервисах, которые помогут им построить оптимальный, безопасный или маршрут для тренировки на шоссейном велосипеде. Существующие сервисы маршрутизации работают, в основном, только для автомобилей. Отсутствие хороших сервисов построения велосипедных маршрутов можно объяснить несколькими проблемами, связанными с данной задачей:

1) Отсутствие данных о дорожках, по которым можно проехать на велосипеде;

2) Для автоматического построения велосипедного маршрута необходимо учитывать множество факторов, относящих к отдельному велосипедисту.

В данной работе разрабатывается метод, который позволяет решить первую и, частично, вторую проблему. Этот метод основан на построении не одного, а нескольких возможных маршрутов, из которых выбирается наиболее подходящий по определенным метрикам.

**Слайд 2.** Целью данной работы является разработка метода построения велосипедных маршрутов и его реализация в виде Web-приложения. Для достижения этой цели необходимо выполнить ряд задач:

* Анализ и выбор алгоритмов для построения оптимальных и субоптимальных маршрутов в графах;
* Разработка метода построения веломаршрутов по различным критериям;
* Построение графа из исходных данных о карте;
* Реализация разработанного метода;
* Исследование полученных результатов.

**Слайд 3.** Основными алгоритмами поиска кратчайших путей являютсяпоиск в ширину, алгоритм Беллмана — Форда, алгоритм Флойда — Уоршелла, Алгоритм Дейкстры, Алгоритм А\*. Поиск в ширину работает с невзвешенными графами. Алгоритмы Беллмана-Форда и Флойда-Уоршелла могут работать с графами с отрицательными весами ребер, но в данной работе это не нужно, поэтому были выбраны алгоритм Дейкстры и А\* из-за меньшей сложности. При этом А\* использует эвристическую функцию, поэтому является наиболее предпочтительным. Алгоритмов поиска альтернативных путей на данный момент известно не так много, поэтому и алгоритм Йена, и алгоритм поиска Е-близких путей выбраны для реализации в данной работе.

**Слайд 4.** Рассмотрим функциональную модель работы метода. Работу метода можно разделить на 3 этапа:

1. Построение графа дорог;
2. Построение маршрута;
3. Отображение маршрута на карте.

Причем первый этап “Построение графа дорог” выполняется только один раз. На вход первого этапа метода подаются исходные данные об электронной карте OSM и область, в которой должны строится маршруты. Результатом выполнения первого этапа является готовый граф, который больше не будет изменяться во времени, поэтому он передается в качестве из одного входных данных на этап построения маршрута, который кроме него в качестве входных данных принимает вид маршрута и начальную и конечную точку. Результатом работы этого этапа является последовательность точек построенного маршрута, которая передается на вход последнему этапу, отображающему маршрут на карте.

**Слайд 5.** Сеть дорог представляется в виде связного ориентированного графа*G* = (𝑉, 𝐸), где 𝑉 – множество вершин графа, 𝐸 – множество ребер графа. Вершинами графа являются пересечения дорог. Каждая вершина имеет географические координаты. Ребрами графа являются дороги, соединяющие пару вершин. Каждое ребро имеет несколько свойств: длина, тип дороги, вид дорожного покрытия. Весом ребра является его длина. Длина ребра высчитывается по формуле расстояния между точками на поверхности Земли.

**Слайд 6.** Рассмотрим этап построения графа более подробно. Исходными данными является XML-файл с данными о карте OpenStreetMap. Из этого файла создается база данных со всей информацией об объектах карты. После этого из базы удаляются все данные о дорогах, по которым запрещен или невозможен проезд на велосипеде. Далее получаются все точки карты, являющиеся пересечениями дорог. На основе этиъ данных создается графовая БД, представляющая граф дорог. Большинство объектов OSM представлены 2 типами данных: точка и линия. Точка (node) является базовым типом, только этот тип данных хранит географическую информацию об объекте – координаты в виде широты и долготы. В XML нотации объект типа node выглядит следующим образом (на слайде). Линия (way) является совокупностью указателей на точки. Линия должна содержать ссылку минимум на один, уже объявленный, объект типа node.

**Слайд 7.** Теперь более подробно рассмотрим этап построения маршрута. Этот этап можно разделить на 3 задачи. Первой является построение кратчайшего маршрута. Построенный кратчайший маршрут поступает на вход задачи построения субоптимальных маршрутов, результатом работы которой является множество субоптимальных маршрутов. Это множество передается в этап выбора маршрута в зависимости от его вида.

**Слайд 8.** На каждом шаге алгоритмов уменьшаются метки 𝑙(𝑠,𝑢) вершин, смежных с текущей вершиной 𝑣. Из непосещенных вершин выбирается та, у которой метка 𝑓(𝑣) имеет наименьшее значение. Для алгоритма Дейкстры 𝑓(𝑣)=𝑙(𝑠,𝑣), где 𝑑(𝑠,𝑣) – расстояние от вершины-источника до вершины 𝑣. Для алгоритма А\* 𝑓(𝑣)=𝑙(𝑠,𝑣)+ℎ(𝑣,𝑡), где ℎ(𝑣,𝑡) – оценка расстояния от 𝑣 до конечной вершины 𝑡, вычисленное по формуле расстояния между точками на поверхности Земли.

**Слайд 9.** Алгоритм находит 𝐾 маршрутов, являющихся наиболее близкими к кратчайшему пути и не имеющих циклов. Уже найденный кратчайший путь обозначается P^1 и добавляется в список A, который в итоге будет содержать K найденных субоптимальных путей. На каждом шаге удалением одного ребра находятся новые маршруты, они добавляются в список B, в котором находятся все найденные на данный момент пути. Из списка B выбирается путь, имеющий минимальную длину, который добавляется в список A, удаляется из списка B. Алгоритм продолжает свою работу, пока в списке A не будет записано K путей.

**Слайд 10.** Алгоритм находит маршруты, которые длиннее кратчайшего не более, чем на величину Е. Алгоритм основан на алгоритме Дейкстры. Отличие его в том,что на каждом шаге он не уменьшает метки вершин, а добавляет новые, каждый раз сохраняя вершину из которой дошли до текущей. После достижения конечной вершины ищутся все пути, длина которых меньше длины кратчайшего + Е.

**Слайд 11.** Теперь рассмотрим последний этап поиска маршрута. На нем из найденного множества субоптимальных путей выбирается тот, который больше соотвествует выбранному виду (безопасному или спортивному). Этот выбор производится на основе типа дороги каждого ребра пути. Для каждого субоптимального маршруты вычисляется мера безопасности по данной формуле. Таким образом, безопасным маршрутом считается тот, мера безопасности которого максимальна, а спортивным – мера безопасности которого максимальна.

**Слайд 12.** На основе описанных алгоритмов был разработан программный продукт, структура которого представлена на данном слайде.

**Слайд 13.** На этом слайде показаны результаты работа метода в зависимости от выбранного вида маршрута.

**Слайд 14.** По полученным результатам было проведено исследование с использованием разных комбинаций алгоритмов для поиска безопасного маршрута. На первом графике представлена зависимость времени поиска маршрута от коэффициентов К и Е. На втором графике проиллюстрована зависимость меры безопасности найденного маршрута от коэффициентов К и Е. Проанализировав полученные результаты можно сказать, что комбинация алгоритма Дейкстры и алгоритма поиска Е-близких путей дает лучшие результаты не только в скорости работы, но и в качестве полученного результата.

**Слайд 15.** В результате выполнения данной работы были решены следующие задачи:

Проведен анализ и выбор алгоритмов для построения оптимальных и субоптимальных маршрутов в графах; Разработан метод построения веломаршрутов по различным критериям; Построен граф из исходных данных о карте; Реализован разработанный метод в виде web-приложения; Проведено исследование результатов работы метода.

В качестве перспективы развития данной работы планируется добавить просмотр подробной информации о построенном маршруте, ручное редактирования пути и возможность отмечать зоны, нежелательные для проезда.