Министерство науки и высшего образования Российской Федерации

ФГБОУ ВО «Волгоградский государственный технический университет»

Кафедра «Системы автоматизированного проектирования  
и поискового конструирования»

Пояснительная записка

к семестровой работе

Гребное судно

|  |  |  |  |
| --- | --- | --- | --- |
| Автор |  |  | Манукян Арсен Ваганович |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) |
| Группа | ИВТ-365 |  | |
|  | (шифр группы) |  | |
| Проверил |  |  | Парыгин Данила Сергеевич |
|  | (подпись и дата подписания) |  | (фамилия, имя, отчество) |

Волгоград 2020 г.

Содержание

[Введение 3](#__RefHeading___Toc41001461)

[1 Систематизация знаний об объекте исследования 4](#__RefHeading___Toc41001462)

[1.1Существующие типы рассматриваемого объекта 4](#__RefHeading___Toc41001463)

[1.2Сценарии поведения 6](#__RefHeading___Toc41001464)

[2. Проектирование модели объекта 10](#__RefHeading___Toc41001465)

[3 Реализация и тестирование модели объекта 11](#__RefHeading___Toc41001466)

[Заключение 14](#__RefHeading___Toc41001467)

[Список использованных источников 15](#__RefHeading___Toc41001468)

[Приложение А 16](#__RefHeading___Toc41001469)

Введение

Современное общество характеризуется интенсивными коммуникационными процессами. Общественная среда предопределяет необходимость перемещения людей в пространстве как функцию их социального поведения, стимулирует прогрессивное расширение доступных территорий и скорости передвижения. Уже на ранних этапах развития людей общественные отношения требовали объединения их существования для осуществления процессов производства, развития культурных ценностей. Так стали образовываться населенные пункты. Но со временем территории расширялись, и у людей появилась потребность перемещения и через водные препятствия, в том числе и реки. Сообщения между различными населенными пунктами, постоянно развивались. Со временем стали налаживаться и водные пути, в том числе и через реки.

Актуальность выбранной мной темы подкрепляется тем, что при отсутствии у многих граждан личных транспортных средств проблема своевременного и качественного удовлетворения спроса на перевозки перерастает из чисто транспортной в социальную, определяющую отношение населения не только к качеству оказываемых транспортных услуг, но и в целом к тем процессам, которые происходят в регионах и стране.

В таких условиях необходимы совместные усилия специалистов отрасли, центральных и региональных органов управления, которые должны быть направлены на совершенствование функционирования транспортного комплекса. Иными словами, необходим системный подход к организации работы транспорта.

1 Систематизация знаний об объекте исследования

* 1. Существующие типы рассматриваемого объекта

**Плот** — конструкция для плаванья, из связанных совместно брёвен или стеблей [камыша](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D0%BC%D1%8B%D1%88), [тростника](https://ru.wikipedia.org/wiki/%D0%A2%D1%80%D0%BE%D1%81%D1%82%D0%BD%D0%B8%D0%BA), скрученных в пучки. Обычно ширина плота относится к длине как 1 к 2—3,5. Средняя скорость плотов при сплаве по рекам составляет 30—50 км в день, а при благоприятных условиях превосходит 100 км. Древнейшее несамоходное плавучее сооружение ([плавсредство](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B2%D1%81%D1%80%D0%B5%D0%B4%D1%81%D1%82%D0%B2%D0%BE" \o "Плавсредство)). Появился, вероятно, раньше лодки, так как для постройки простейшего плота не требуется специальных [инструментов](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BC%D0%B5%D0%BD%D1%82), а материалы общедоступны. Из [бальсового дерева](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%BB%D1%8C%D1%81%D0%B0_(%D0%B4%D0%B5%D1%80%D0%B5%D0%B2%D0%BE)) был построен знаменитый плот [Кон-Тики](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BD-%D0%A2%D0%B8%D0%BA%D0%B8), на котором норвежский путешественник [Тур Хейердал](https://ru.wikipedia.org/wiki/%D0%A5%D0%B5%D0%B9%D0%B5%D1%80%D0%B4%D0%B0%D0%BB,_%D0%A2%D1%83%D1%80) со своей командой совершил путешествие через [Тихий океан](https://ru.wikipedia.org/wiki/%D0%A2%D0%B8%D1%85%D0%B8%D0%B9_%D0%BE%D0%BA%D0%B5%D0%B0%D0%BD) к островам [Полинезии](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BD%D0%B5%D0%B7%D0%B8%D1%8F). В наше время [надувной спасательный плот](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%83%D0%B2%D0%BD%D0%BE%D0%B9_%D1%81%D0%BF%D0%B0%D1%81%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D0%BD%D1%8B%D0%B9_%D0%BF%D0%BB%D0%BE%D1%82) широко используются как спасательное средство наряду со [шлюпками](https://ru.wikipedia.org/wiki/%D0%A8%D0%BB%D1%8E%D0%BF%D0%BA%D0%B0). Применяется для транспортировки леса (брёвен) по реками и [водного туризма](https://ru.wikipedia.org/wiki/%D0%92%D0%BE%D0%B4%D0%BD%D1%8B%D0%B9_%D1%82%D1%83%D1%80%D0%B8%D0%B7%D0%BC). Также в водном туризме применяется особая разновидность плота — [честер](https://ru.wikipedia.org/wiki/%D0%A7%D0%B5%D1%81%D1%82%D0%B5%D1%80_(%D1%81%D1%83%D0%B4%D0%BD%D0%BE)). Его основными отличиями от плота являются способ гребли — для управления используются не [греби](https://ru.wikipedia.org/w/index.php?title=%D0%93%D1%80%D0%B5%D0%B1%D0%B8&action=edit&redlink=1), а обычные вёсла, которые гребцы держат в руках, а также посадка гребцов: гребцы сидят лицом друг к другу на гондолах, расположенных поперёк хода судна. Для переправы через реку строят небольшой плот — салик для переправ. Такой салик для одного-двух человек может состоять всего из двух бревен. Плоты также применяются для [лесосплава](https://ru.wikipedia.org/wiki/%D0%9B%D0%B5%D1%81%D0%BE%D1%81%D0%BF%D0%BB%D0%B0%D0%B2" \o "Лесосплав).

**Катамара́н**  (от [тамильского](https://ru.wikipedia.org/wiki/%D0%A2%D0%B0%D0%BC%D0%B8%D0%BB%D1%8C%D1%81%D0%BA%D0%B8%D0%B9_%D1%8F%D0%B7%D1%8B%D0%BA) கட்டு மரம் *каттумарам* букв. «связанные брёвна»): Двухкорпусное судно; корпуса судна соединяются сверху *мостом* ([ферменного](https://ru.wikipedia.org/wiki/%D0%A4%D0%B5%D1%80%D0%BC%D0%B0_(%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F)) или [палубного](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BB%D1%83%D0%B1%D0%B0) типа); разновидность катамарана с вынесенным за борт балансирным поплавком называется [*проа*](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%B0). Суда, состоящие из двух корпусов, соединены между собой [палубным](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BB%D1%83%D0%B1%D0%B0) или [ферменным](https://ru.wikipedia.org/wiki/%D0%A4%D0%B5%D1%80%D0%BC%D0%B0_(%D0%BA%D0%BE%D0%BD%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%86%D0%B8%D1%8F)) мостом.

Конструктивной особенностью большинства [парусных](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D1%83%D1%81%D0%BD%D0%BE%D0%B5_%D1%81%D1%83%D0%B4%D0%BD%D0%BE) катамаранов является их бес[палубная](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D0%BB%D1%83%D0%B1%D0%B0) конструкция — корпуса соединены [балками](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%BB%D0%BA%D0%B0_(%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D0%BA%D0%B0)), пространство между которыми закрыто сеткой, по которой передвигаются члены экипажа. Подобное решение существенно ограничивает условия обитаемости, но с этим мирятся на небольших судах, предназначенных для участия в соревнованиях.

**Ло́дка** — небольшое [судно](https://ru.wikipedia.org/wiki/%D0%A1%D1%83%D0%B4%D0%BD%D0%BE). В русском Военно-морском словаре[[1]](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B4%D0%BA%D0%B0#cite_note-1) лодка определяется, как судно длиной до девяти [метров](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D1%80), шириной до трёх метров и [грузоподъёмностью](https://ru.wikipedia.org/wiki/%D0%93%D1%80%D1%83%D0%B7%D0%BE%D0%BF%D0%BE%D0%B4%D1%8A%D1%91%D0%BC%D0%BD%D0%BE%D1%81%D1%82%D1%8C) до пяти [тонн](https://ru.wikipedia.org/wiki/%D0%A2%D0%BE%D0%BD%D0%BD%D0%B0). Как правило, лодки — гребные суда, но бывают и [парусные](https://ru.wikipedia.org/wiki/%D0%9F%D0%B0%D1%80%D1%83%D1%81) лодки, однако небольшие [спортивные](https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%BE%D1%80%D1%82) парусные суда принято называть [яхтами](https://ru.wikipedia.org/wiki/%D0%AF%D1%85%D1%82%D0%B0). Лодка может быть оборудована [мотором](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%82%D0%BE%D1%80) (мотор может быть как вспомогательным, так и основным источником энергии). В соответствии с принятой в большинстве стран классификацией, [моторными лодками](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D1%82%D0%BE%D1%80%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B4%D0%BA%D0%B0) называют суда с [подвесным мотором](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B4%D0%B2%D0%B5%D1%81%D0%BD%D0%BE%D0%B9_%D0%BC%D0%BE%D1%82%D0%BE%D1%80)[[2]](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B4%D0%BA%D0%B0#cite_note-2). Небольшие суда со стационарным мотором чаще называют [катером](https://ru.wikipedia.org/wiki/%D0%9A%D0%B0%D1%82%D0%B5%D1%80).

Основным материалом для производства лодок служат лёгкие сплавы (как правило, [дюралюминий](https://ru.wikipedia.org/wiki/%D0%94%D1%8E%D1%80%D0%B0%D0%BB%D1%8E%D0%BC%D0%B8%D0%BD%D0%B8%D0%B9)), [древесина](https://ru.wikipedia.org/wiki/%D0%94%D1%80%D0%B5%D0%B2%D0%B5%D1%81%D0%B8%D0%BD%D0%B0), [композиционные материалы](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B5_%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D0%B8%D0%B0%D0%BB%D1%8B) и [полимеры](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D0%B8%D0%BC%D0%B5%D1%80). Лодки, изготавливаемые из композитных материалов, обычно называют [пластиковыми](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D1%81%D1%82%D0%B8%D0%BA%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B4%D0%BA%D0%B0). Особый тип лодок — [надувные лодки](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B4%D1%83%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B4%D0%BA%D0%B0). Лодки РИБ (RIB — rigid inflatable boat) — [надувные лодки с жёстким днищем и надувными бортами](https://ru.wikipedia.org/wiki/%D0%96%D1%91%D1%81%D1%82%D0%BA%D0%BE-%D0%BA%D0%BE%D1%80%D0%BF%D1%83%D1%81%D0%BD%D0%B0%D1%8F_%D0%BD%D0%B0%D0%B4%D1%83%D0%B2%D0%BD%D0%B0%D1%8F_%D0%BB%D0%BE%D0%B4%D0%BA%D0%B0), обеспечивающими непотопляемость.

В настоящее время лодки используются для отдыха, [рыболовства](https://ru.wikipedia.org/wiki/%D0%A0%D1%8B%D0%B1%D0%BE%D0%BB%D0%BE%D0%B2%D1%81%D1%82%D0%B2%D0%BE) и перевозки грузов с судна на берег там, где нет [причалов](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B8%D1%87%D0%B0%D0%BB), передвижения по населённым пунктам во время наводнений и как спасательное средство ([шлюпки](https://ru.wikipedia.org/wiki/%D0%A8%D0%BB%D1%8E%D0%BF%D0%BA%D0%B0)).

1.2 Сценарии поведения

Диаграмма прецендентов содержит акторов: Владелец, проверяющий состояние, диаграмма прецендентов. Владелец может купить, пользоваться и продать судно. Проверяющий состояние диагностировать, мыть и заниматься починкой. Туристы — арендовать. Диаграмма представлена на рисунке 1.

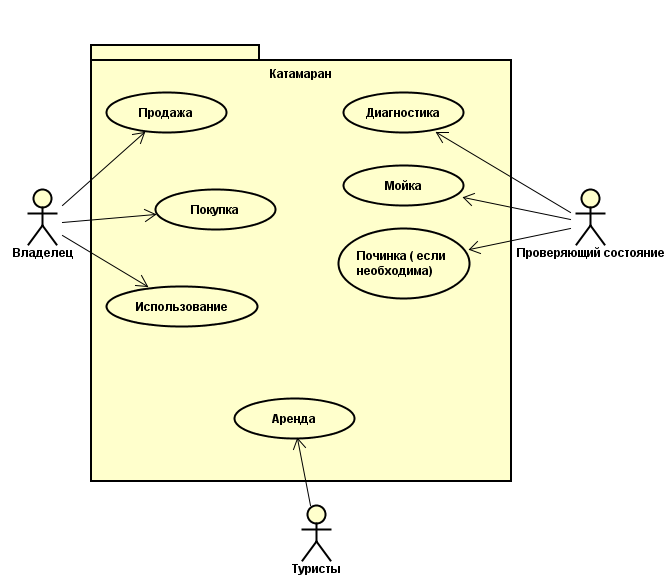


Рисунок 1 – Диаграмма прецедентов

На диаграмме классов описаны гребных суден и параметры, которыми они обладают. Диаграмма отображается на рисунке 2.

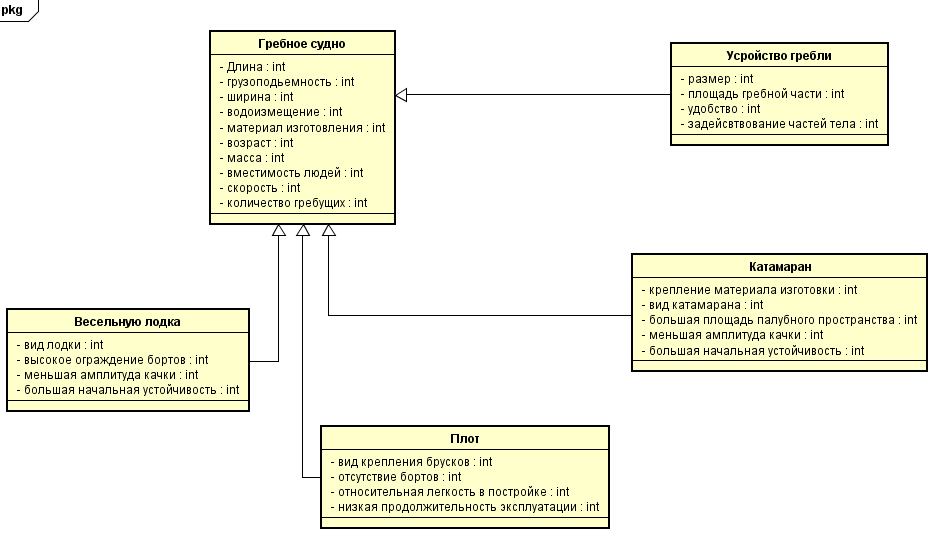


Рисунок 2 – Диаграмма классов

На диаграмме пользователей на рисунке 3 описан жизненный цикл гребного судна, начиная с производителя(завода или просто людей) и заканчивая поломкой и дальнейшим утилем.

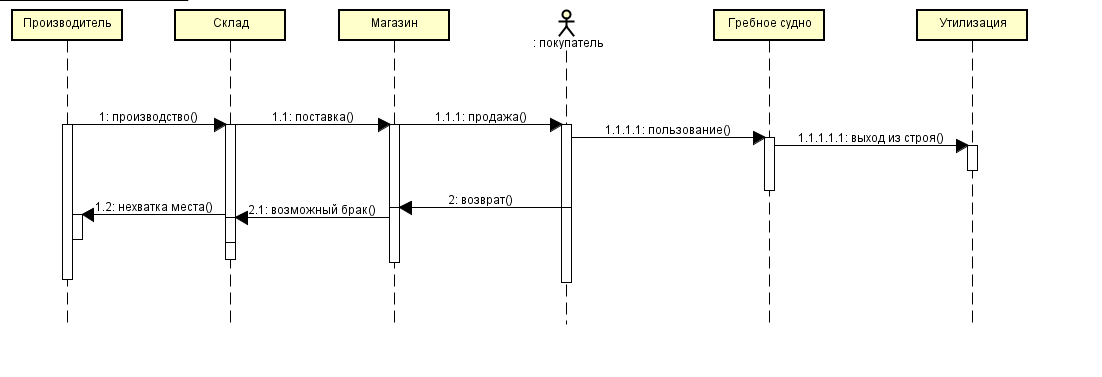


Рисунок 3 – Диаграмма последовательностей

Диаграмма активностей на рисунке 4 описывает возможные переходы жизненного цикла скоростной лодки между системами: создание, склад, магазин, владелец, утиль.

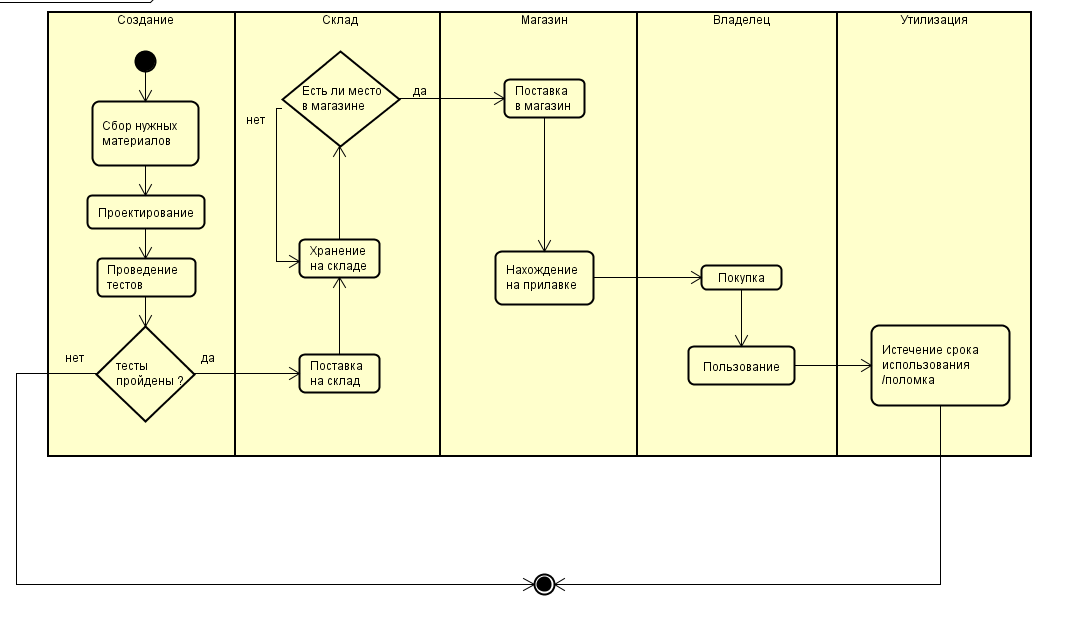


Рисунок 4 – Диаграмма активностей

Диаграмма состояний, изображенная на рисунке 5, показывает общие состояния системы.

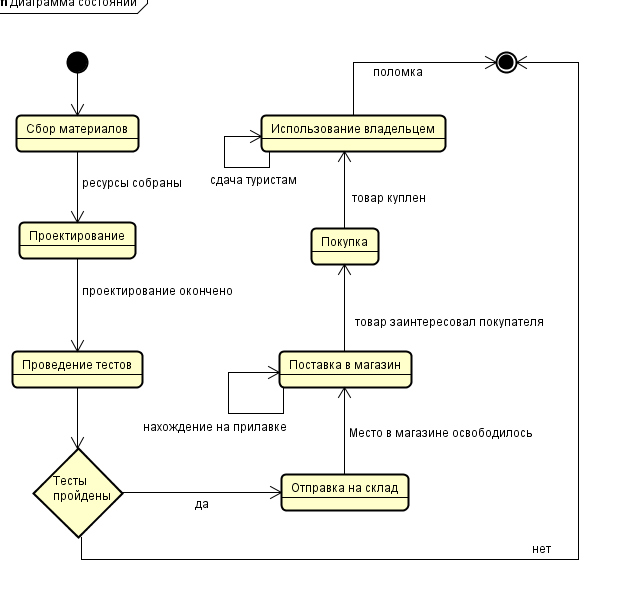


Рисунок 5 – Диаграмма состояний

2. Проектирование модели объекта

В данной блок-схеме (рисунок 6) наиболее подробно описывается алгоритм пользования гребным судном человеком. Цикл пользования и жизни судна в целом может продолжаться бесконечно долго до выполнения условия серьёзной поломки, или же отсутствия необходимости в судне и дальнейшей его продажи

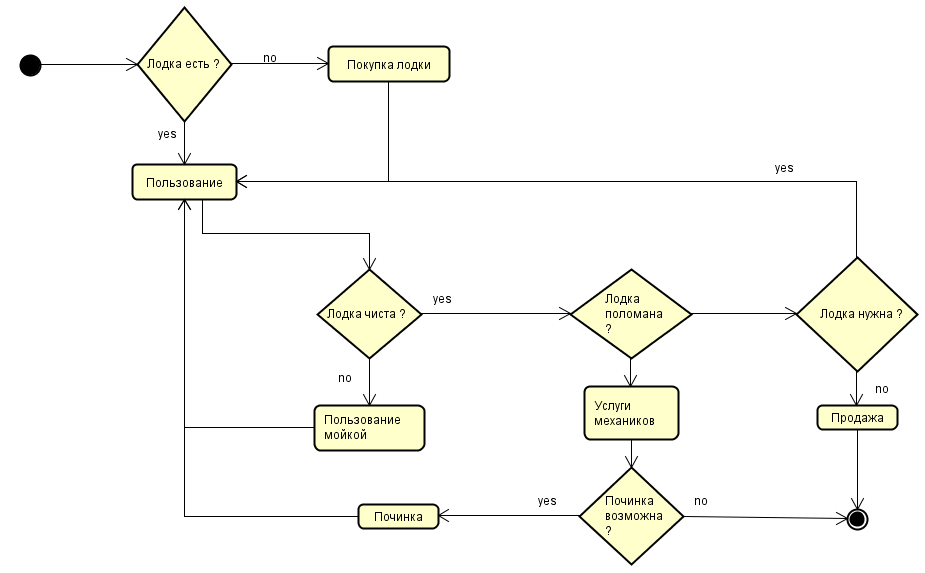
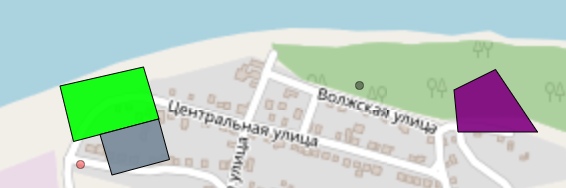


Рисунок 6 – Блок-схема сценариев

3 Реализация и тестирование модели объекта

Алгоритм начинается с появления первого покупателя (розовая точка) около входа в магазин (зеленая зона), и появления грузовика (черно-серая точка), поставлящего груз с лодками из завода (фиолетовая зона), в склад (серая зона). Далее, после достижения покупателем входа в магазин, он покупает самую дорогую лодку, на которую ему хватает денег (которые генерируются в рандомном количестве у покупателя от 0 до 1000) и сразу же приступает к использованию. Вышеперечисленное можно увидеть на рисунке 6.



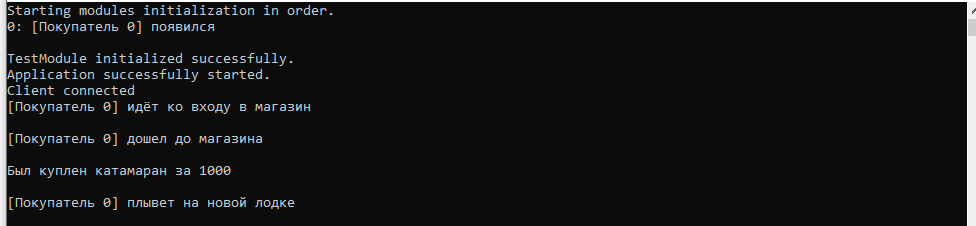
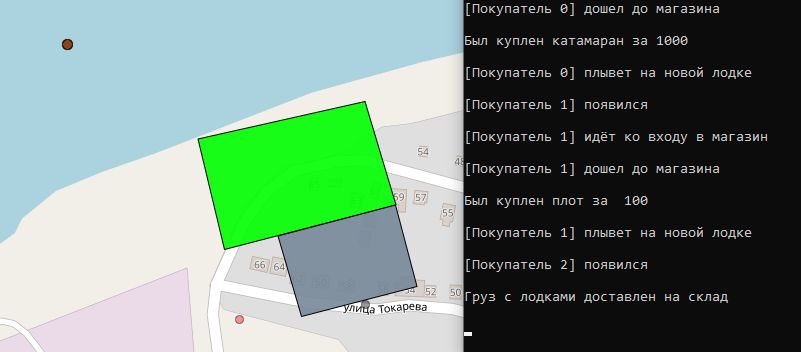


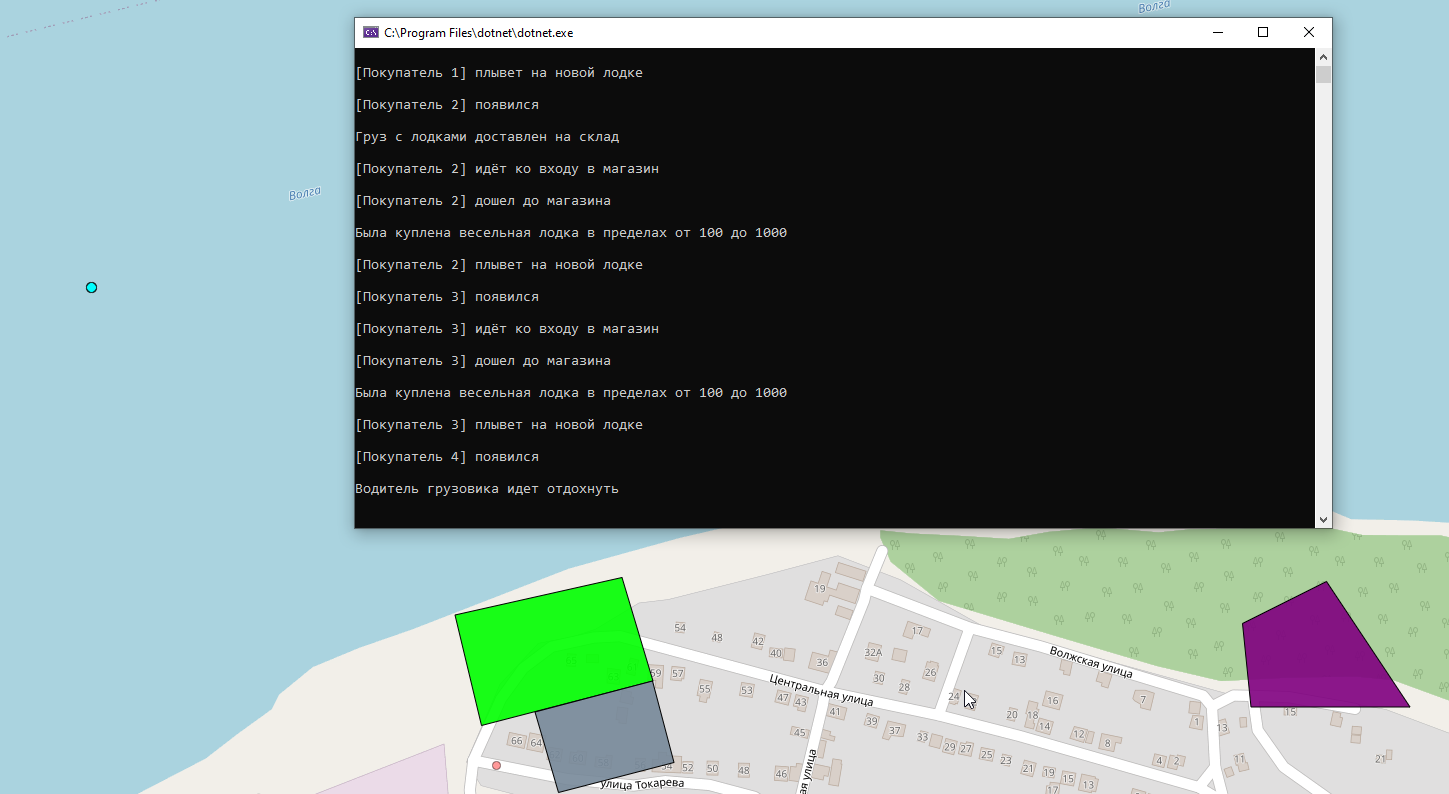
Рисунок 6 – Начало сценария

Как видно на рисунке 7, генерация покупателей происходит бесконечно, также видно, когда грузовик достигает склада, в консоль выводится сообщение о разгрузке грузовика. По сценарию, он должен делать это около 5-10 секунд реального времени, а затем отправиться обратно на завод.

Рисунок 7 – Сценарий действий грузовика

На следующем сценарии (рисунок 8) можно увидеть, что при покупке лодки происходит генерация новой точки, цвет которой зависит от типа лодки. Также тип гребного судна: плот (коричневая точка), весельная лодка (голубая точка), катамаран (синяя точка) определяется разными характеристиками, такими как: цена, скорость, шанс поломки. Соответственно, при дальнейшей эксплуатации лодки, имеется шанс на ее поломку, и дальнейшее исчезновение лодки с карты.

Также в данном рисунке можно увидеть, что грузовик, по достижению производственной зоны, исчезает из карты с выводом сообщения в консоль.

Рисунок 8 – Уход на отдых и цвета лодок

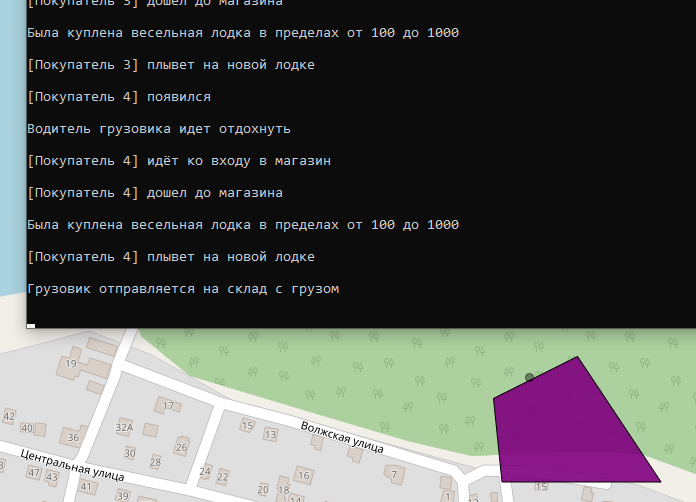
По истечению некоторого времени грузовик снова появляется карте (рисунок 9) и отправляется с грузом на склад.

Рисунок 9 – Появление грузовика

Заключение

Таким образом, из всего вышесказанного можно сделать вывод, что в настоящее время жизнь практически каждого человека связана с необходимостью постоянного перемещения и довольно немаловажную роль в этом играет гребное судно. Имея значение в таких разнообразных сферах общества, как туризм, спорт, гребное судно при всех своих преимуществах и недостатках никогда не потеряет актуальности в современном мире.

Рассмотренная в данной работе система помогает определить сильные и слабые стороны таких суден, как плот, весельная лодка, катамаран, и тем самым собственноручно оценить важность данных транспортных средств.

Список использованных источников

1. Разработка компонентов аналитических и ориентированных на добычу знаний программных средств поддержки принятия решений: методические указнания/ сост. Д.С. Парыгин, С.С. Буров, А.А. Пьянникова; ВолгГТУ. – Волгоград, 2020.
2. Установка, запуск программ для работы с OSMLS – [электронный ресурс] – Режим доступа: https://docs.google.com/document/d/1ku00EoGw6S3cMoC-EM1d2Z-hHY0rq8\_XB4Lj6CGS25E/edit – Дата обращения 29.05.2020.
3. Гребное\_судно – [электронный ресурс] – Режим доступа: [https://ru.wikipedia.org/wiki/](https://ru.wikipedia.org/wiki/Маломерное_судно)Гребное\_судно – Дата обращения 29.05.2020.
4. NTS Topology Suite – [электронный ресурс] – Режим доступа: https://nettopologysuite.github.io/ – Дата обращения 29.05.2020.

Приложение А

Программный код модуля

using NetTopologySuite.Geometries;

using OSMLSGlobalLibrary;

using OSMLSGlobalLibrary.Map;

using OSMLSGlobalLibrary.Modules;

using System;

using System.Collections.Generic;

namespace TestModule

{

public class TestModule : OSMLSModule

{

Shop shop;

Factory factory;

Storage storage;

List<Customer> customers;

Coordinate customerStartCoord = new Coordinate(4964618, 6197949);

//Coordinate customerStartCoord = new Coordinate(4960405, 6195000);

Truck truck;

Coordinate factoryStart = new Coordinate(4965377, 6198207);

//Coordinate factoryStart = new Coordinate(4961377, 6195258);

Random random;

int time = 0;

protected override void Initialize()

{

random = new Random(10);

// создание магазина

shop = new Shop(4964768, 6198228, random);

//shop = new Shop(4960555, 6195279, random);

MapObjects.Add(shop);

//создание склада

storage = new Storage(4964664, 6198067);

//storage = new Storage(4960451, 6195118);

MapObjects.Add(storage);

//создание завода

factory = new Factory(4965509, 6198173);

//factory = new Factory(4961296, 6195224);

MapObjects.Add(factory);

//создание грузовика

truck = new Truck(factoryStart, 2);

MapObjects.Add(truck);

customers = new List<Customer>();

//создание первого покупателя

var customer = new Customer(customerStartCoord.Copy(), 1, 1000, customers.Count);

MapObjects.Add(customer);

customers.Add(customer);

}

/// <summary>

/// Вызывается постоянно, здесь можно реализовывать логику перемещений и всего остального, требующего времени.

/// </summary>

/// <param name="elapsedMilliseconds">TimeNow.ElapsedMilliseconds</param>

public override void Update(long elapsedMilliseconds)

{

truck.delivering(MapObjects, truck); // делаем поставки грузовиком

time++;

if (time % 320 == 0) //периодически создаем новых покупателей

{

time = 0;

int money = random.Next(0, 1000);

var customer = new Customer(customerStartCoord.Copy(), 1, money, customers.Count);

customers.Add(customer);

MapObjects.Add(customer);

}

foreach (Customer customer in customers)

{

customer.doingSmthg(shop); //в зависимости от разных условий, покупатель делает разные вещи

if (customer.boat != null && !customer.BoatIsAdded)//Если лодка есть, но она ещё не была добавлена на карту

{

MapObjects.Add(customer.boat);

customer.BoatIsAdded = true;

}

if (customer.BoatIsBroken)//Если лодка сломалась

{

MapObjects.Remove(customer.boat);

MapObjects.Remove(customer);

}

}

}

}

#region Truck

[CustomStyle(

@"new ol.style.Style({

image: new ol.style.Circle({

opacity: 1.0,

scale: 1.0,

radius: 4,

fill: new ol.style.Fill({

color: 'rgba(0, 0, 0, 0.4)'

}),

stroke: new ol.style.Stroke({

color: 'rgba(0, 0, 0, 0.4)',

width: 1

}),

})

});

")]

class Truck : Point //класс грузовика, поставляющего лодки на склад

{

private int time = 0;

private bool delivered = false;

private Coordinate storageStart = new Coordinate(4964752, 6197981);

//private Coordinate wareStart = new Coordinate(4960539, 6195032);

private Coordinate factoryStart = new Coordinate(4965590, 6198207);

//private Coordinate plantStart = new Coordinate(4961377, 6195258);

public double Speed { get; }

public Truck(Coordinate coordinate, double speed) : base(coordinate)

{

//Необходимо переопределять в переменные класса(Speed) для дальнейшего хранения в объекте подобных параметров

Speed = speed;

}

public void delivering(IInheritanceTreeCollection<Geometry> MapObjects, Truck truck)

{

Point closestObj;

// в зависимости от bool delivered определяем, куда поедет грузовик

//delivered меняется на противоположный при достижении какого-илбо пункта назначения(склад или завод)

if (delivered) closestObj = new Point(factoryStart);

else closestObj = new Point(storageStart);

Coordinate relativeCoord = new Coordinate(closestObj.X - X, closestObj.Y - Y);

double curSpeedX = Math.Abs(relativeCoord.X) > Speed ? Speed : Speed / 2;

double curSpeedY = Math.Abs(relativeCoord.Y) > Speed ? Speed : Speed / 2;

//двигаем грузовик

X += curSpeedX \* Math.Sign(relativeCoord.X);

Y += (curSpeedY \* Math.Sign(relativeCoord.Y)) / 4;

if (Coordinate.Distance(closestObj.Coordinate) < 2) // если добрался до точки назначения

{

time++;

if (time > 200 && Coordinate.Distance(factoryStart) < 3) // если немного постоял около завода

{

if (time == 201)

{

Console.WriteLine("Водитель грузовика идет отдохнуть \n");

MapObjects.Remove(truck);

}

if (time == 400)

{

Console.WriteLine("Грузовик отправляется на склад с грузом \n");

MapObjects.Add(truck);

delivered = false;

time = 0;

}

}

else if (time > 200 && Coordinate.Distance(storageStart) < 3) // если немного постоял около склада

{

Console.WriteLine("Груз с лодками доставлен на склад \n");

delivered = true;

time = 0;

}

}

}

}

#endregion

#region Boat

[CustomStyle(

@"new ol.style.Style({

image: new ol.style.Circle({

opacity: 1.0,

scale: 1.5,

radius: 4,

fill: new ol.style.Fill({

color: 'rgba(255, 255, 0, 0.4)'

}),

stroke: new ol.style.Stroke({

color: 'rgba(0, 0, 0, 0.4)',

width: 1

}),

})

});

")]

class Boat : Point //родительский класс для видов лодок

{

public double Speed { get; }//Скорость передвижения

public double FailureChance { get; }//Шанс поломки

public double Price { get; set; }//Цена

public Random Random { get; }

public Boat(Coordinate coordinate, double speed, double price, double failureChance, Random random) : base(coordinate)

{

Speed = speed;

Price = price;

Random = random;

FailureChance = failureChance;

}

public bool CheckFailure()//Проверка поломки

{

if (Random.NextDouble() <= FailureChance)//Если полученное значение меньше шанса поломки

{

return true;

}

else

{

return false;

}

}

}

#endregion

#region Raft

[CustomStyle(

@"new ol.style.Style({

image: new ol.style.Circle({

opacity: 1.0,

scale: 1.0,

radius: 5,

fill: new ol.style.Fill({

color: 'rgba(139, 69, 19, 1)'

}),

stroke: new ol.style.Stroke({

color: 'rgba(0, 0, 0, 1)',

width: 1

}),

})

});

")]

class Raft : Boat//Скорость 1, цена 100, шанс поломки 0,001

{

public Raft(Coordinate coordinate, Random random) : base(coordinate, 1, 100, 0.001, random)

{

}

}

#endregion

#region RowingBoat

[CustomStyle(

@"new ol.style.Style({

image: new ol.style.Circle({

opacity: 1.0,

scale: 1.0,

radius: 5,

fill: new ol.style.Fill({

color: 'rgba(0, 255, 255, 1)'

}),

stroke: new ol.style.Stroke({

color: 'rgba(0, 0, 0, 1)',

width: 1

}),

})

});

")]

class RowingBoat : Boat//Скорость 2, цена 500, шанс поломки 0,0001

{

public RowingBoat(Coordinate coordinate, Random random) : base(coordinate, 2, 500, 0.0001, random)

{

}

}

#endregion

#region Catamaran

[CustomStyle(

@"new ol.style.Style({

image: new ol.style.Circle({

opacity: 1.0,

scale: 1.0,

radius: 5,

fill: new ol.style.Fill({

color: 'rgba(0, 0, 255, 1)'

}),

stroke: new ol.style.Stroke({

color: 'rgba(0, 0, 0, 1)',

width: 1

}),

})

});

")]

class Catamaran : Boat //Скорость 2, цена 1000, шанс поломки 0,00001

{

public Catamaran(Coordinate coordinate, Random random) : base(coordinate, 2, 1000, 0.00001, random)

{

}

}

#endregion

#region Customer

[CustomStyle(

@"new ol.style.Style({

image: new ol.style.Circle({

opacity: 1.0,

scale: 1.5,

radius: 4,

fill: new ol.style.Fill({

color: 'rgba(255, 0, 0, 0.4)'

}),

stroke: new ol.style.Stroke({

color: 'rgba(0, 0, 0, 0.4)',

width: 1

}),

})

});

")]

class Customer : Point

{

public Boat boat { get; set; }//Купленный плот

private Coordinate shopEntr = new Coordinate(4964618, 6198080);

//private Coordinate shopEntr = new Coordinate(4960405, 6195141);

private int Number { get; }

private int Money { get; set; }

public bool BoatIsAdded { get; set; }

public bool BoatIsBroken { get; set; }

public double Speed { get; }

private int time = 0;

public Customer(Coordinate coordinate, double speed, int money, int number) : base(coordinate)

{

Money = money;

Number = number;

BoatIsAdded = false;

BoatIsBroken = false;

Speed = speed;

Console.WriteLine($"[Покупатель {number}] появился\n");

}

public void doingSmthg(Shop shop)

{

if (boat == null) //лодка ещё не куплена

{

if (!Coordinate.Equals(shopEntr)) //Ещё не подошёл ко входу

{

moveToShop();

}

else if (BoatIsBroken)

{

//лодка сломлена

}

else

{

Console.WriteLine($"[Покупатель {Number}] дошел до магазина\n");

BuyRaft(shop);

}

}

else

{

if (boat.CheckFailure())//Если лодка не прошла проверку

{

BoatIsBroken = true;//Сломать лодку

Console.WriteLine($"[У Покупателя {Number}] сломалась лодка\n");

}

else

{

//двигаем покупателя и лодку

boat.Coordinate.X -= boat.Speed;

boat.Coordinate.Y += boat.Speed;

Coordinate.X = boat.Coordinate.X;

Coordinate.Y = boat.Coordinate.Y;

}

}

}

public void moveToShop()

{

time++;

if (time % 100 == 0)

{

Console.WriteLine($"[Покупатель {Number}] идёт ко входу в магазин\n");

}

Y += Speed;

}

public void BuyRaft(Shop shop)//Купить мотоцикл в магазине

{

boat = shop.sellBoat(Money);

Console.WriteLine($"[Покупатель {Number}] плывет на новой лодке\n");

Money -= (int)boat.Price; //Отнять от денег цену лодки

}

}

#endregion

#region Shop

[CustomStyle(

@"new ol.style.Style({

fill: new ol.style.Fill(

{

color: 'rgba(0, 255, 0, 0.9)'

}),

stroke: new ol.style.Stroke(

{

color: 'rgba(0, 0, 0, 1)',

width: 1

}),

});

")]

class Shop : Polygon

{

private Random Random { get; }

public Shop(int X, int Y, Random random)

: base(new LinearRing(

new Coordinate[] {

new Coordinate(X, Y),

new Coordinate(X - 200, Y - 45),

new Coordinate(X - 168, Y - 177),

new Coordinate(X + 37, Y - 124),

new Coordinate(X, Y)

}))

{

Random = random;

}

public Boat sellBoat(int money)//Продаём лодку

{

var boatCoord = new Coordinate(4964618, 6198090);

//var boatCoord = new Coordinate(4960405, 6195141);

//Смотрим на какую лодку хватит денег у покупателя

if (money >= 1000)

{

Console.WriteLine("Был куплен катамаран за 1000\n");

return new Catamaran(boatCoord, this.Random);

}

else if (money >= 100 && money < 1000)

{

Console.WriteLine("Была куплена весельная лодка в пределах от 100 до 1000\n");

return new RowingBoat(boatCoord, this.Random);

}

else

{

Console.WriteLine("Был куплен плот за 100\n");

return new Raft(boatCoord, this.Random);

}

}

}

#endregion

#region Factory

[CustomStyle(

@"new ol.style.Style({

fill: new ol.style.Fill(

{

color: 'rgba(128, 0, 128, 0.9)'

}),

stroke: new ol.style.Stroke(

{

color: 'rgba(0, 0, 0, 1)',

width: 1

}),

});

")]

class Factory : Polygon

{

public Factory(int X, int Y)

: base(new LinearRing(

new Coordinate[] {

new Coordinate(X, Y),

new Coordinate(X + 100, Y + 50),

new Coordinate(X + 200, Y - 100),

new Coordinate(X + 10, Y - 100),

new Coordinate(X, Y)

}))

{

}

}

#endregion

#region Storage

[CustomStyle(

@"new ol.style.Style({

fill: new ol.style.Fill(

{

color: 'rgba(119, 136, 153, 0.9)'

}),

stroke: new ol.style.Stroke(

{

color: 'rgba(0, 0, 0, 1)',

width: 1

}),

});

")]

class Storage : Polygon

{

public Storage(int X, int Y)

: base(new LinearRing(

new Coordinate[] {

new Coordinate(X, Y),

new Coordinate(X + 140, Y + 37),

new Coordinate(X + 166, Y - 60),

new Coordinate(X + 28, Y - 96),

new Coordinate(X, Y)

}))

{

}

}

#endregion

}