**Государственное бюджетное профессиональное образовательное учреждение**

**«Нижегородский Губернский колледж»**

Методическая комиссия «Информатика и вычислительная техника»

Допущен к защите:

преподаватель

Л.В. Мухина,

Е.П. Голубева,

«21»\_\_марта\_\_2025 г.

**ОТЧЕТ** **ПО**

**УЧЕБНОЙ ПРАКТИКЕ**

ПМ.01 РАЗРАБОТКА МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

Руководитель: \_\_\_\_\_\_\_\_\_\_ Мухина Л.В., Голубева Е.П. 21.03.2025 г.

Студент: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Сухарева А.К. 21.03.2025 г.

Специальность, группа: 09.02.07, 43П

Нижний Новгород

2025 г.

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc192209662)

[ОСНОВНАЯ ЧАСТЬ 4](#_Toc192209663)

[1. Разработка, администрирование и защита баз данных 4](#_Toc192209664)

[2. Разработка и сопровождение модулей программного обеспечения для компьютерных системы 5](#_Toc192209665)

[3. Осуществление интеграции программных модулей 9](#_Toc192209666)

[4. Проектирование 9](#_Toc192209667)

[ЗАКЛЮЧЕНИЕ 12](#_Toc192209668)

[ПРИЛОЖЕНИЕ А 13](#_Toc192209669)

[ПРИЛОЖЕНИЕ Б 29](#_Toc192209670)

[ПРИЛОЖЕНИЕ В 34](#_Toc192209671)

# ВВЕДЕНИЕ

Учебная практика по ПМ.01 Разработка модулей программного обеспечение для компьютерных систем проходила в ГБПОУ «Нижегородский Губернский колледж» в период с 01.03.2025г. по 22.03.2025г.

Целью данной практики является приобретение навыков проектирования, разработки и тестирования программных модулей, а также интеграции их в существующие информационные системы. В процессе выполнения практических заданий изучаются методы и технологии разработки модулей программного обеспечение, анализируются требования к системам и проектируются решения, соответствующие современным стандартам и требованиям.

Для достижения цели поставлены следующие задачи:

1. Спроектировать базу данных;
2. Разработать программное обеспечение с интеграцией БД;
3. Разработать библиотеку;
4. Разработать модульные тесты для библиотеки.

Практическим результатом прохождения практики является отчёт по выполнению заданий с представленными в нём рисунками и листингами кодов.

# ОСНОВНАЯ ЧАСТЬ

## Разработка, администрирование и защита баз данных

Производственная компания «Мастер пол» выпускает напольные покрытия. Компания занимается производством и реализует свою продукцию через партнеров, которые доставляют продукцию компании до конечных потребителей.

С целью оптимизации деятельности компании разрабатывается система, в которой заказчик выделил несколько подсистем:

− продукция;

− склад и материалы;

− производство;

− сотрудники;

− партнеры.

Для данной работы были разработаны таблицы и поля, которые относятся к разрабатываемой подсистеме и её функционалу. ER-диаграмма для предметной области «Мастер пол» представлена на рисунке 1. Также в приложение А представлены скрипты для создания таблиц БД и наполнения данными в листингах А1 и А2.

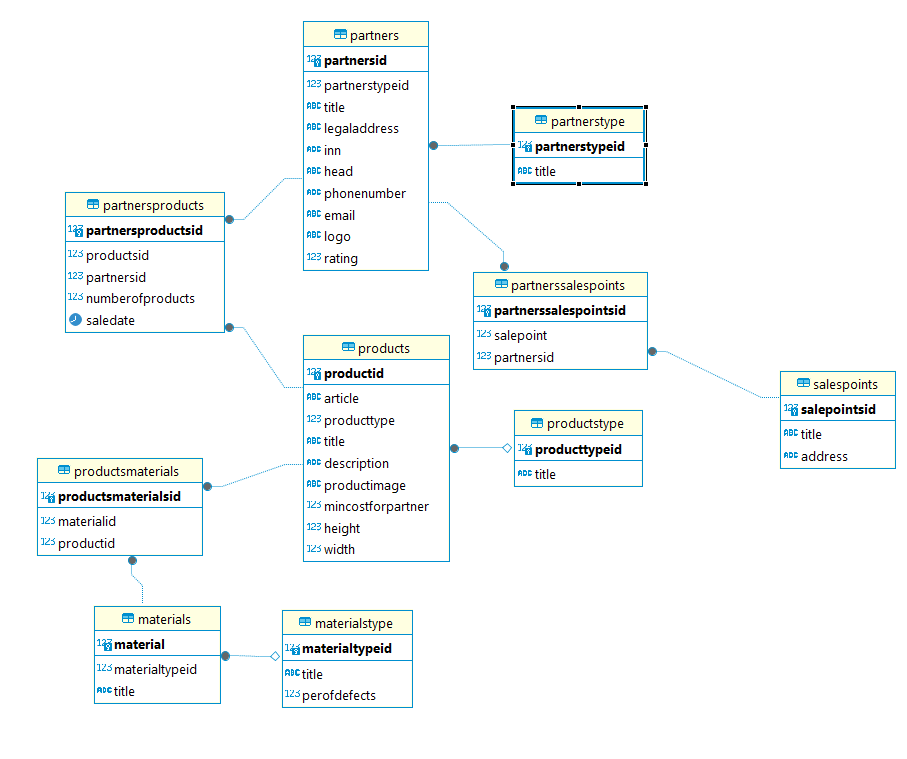


Рисунок №1 – ER-диаграмма к предметной области «Мастер пол»

## Разработка и сопровождение модулей программного обеспечения для компьютерных системы

Для данной предметной области было разработано программное обеспечение для учёта партнёров компании и истории их взаимодействий с продукцией. Основными функциями данной системы являются просмотр существующих партнёров компании, изменение данных существующих партнёров, добавление новых партнёров, просмотр истории реализации продукции партнёрами.

Для разработки программного обеспечения для этой предметной области были использованы следующие инструменты разработки:

1. Visual Studio 2022
2. .NET Framework 8.0
3. DBeaver Community 24.0.5

Всего программное обеспечение включает в себя 4 страницы:

1. Список партнёров;

На данной странице пользователь может изучить список партнёров компании, а также:

* Перейти на страницу добавления нового партнёра по кнопке "Добавить";
* Перейти на страницу редактирования данных существующего партнёра по кнопке "Редактировать";
* Просмотреть историю реализации продукции партнёром по кнопке "История реализации продукции".

Интерфейс данной страницы представлен на рисунке 2.

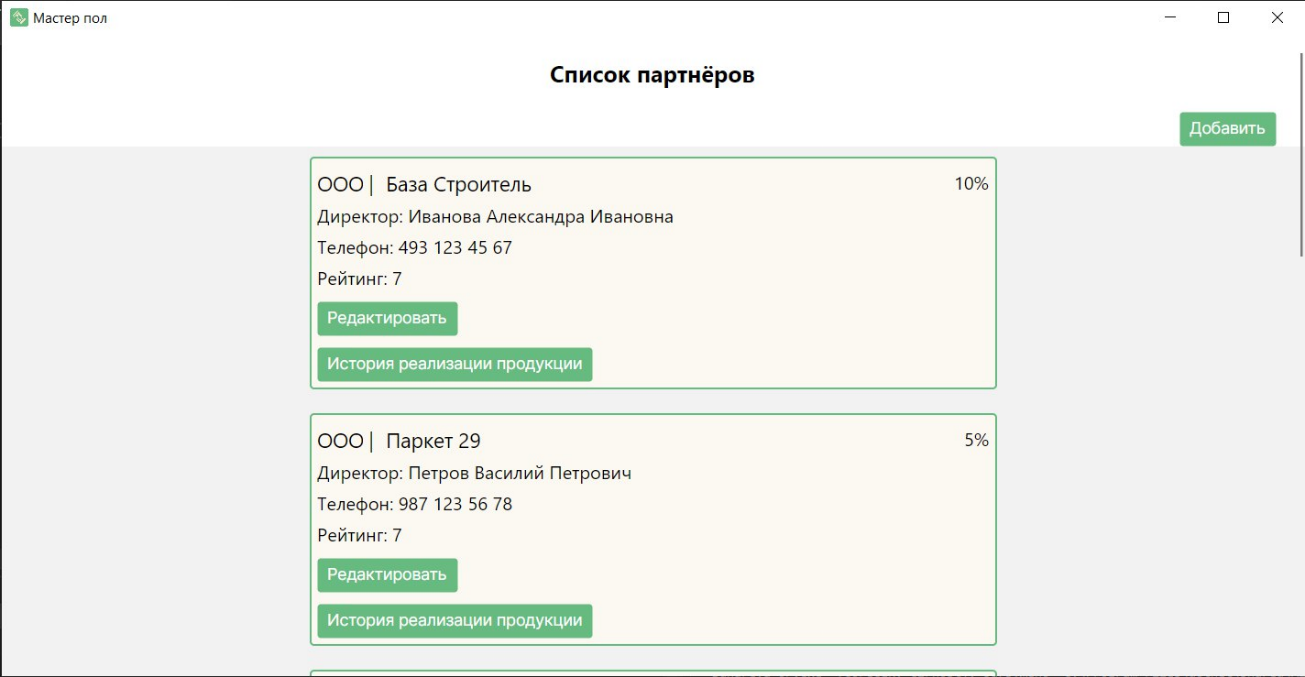


Рисунок №2 - Список партнёров

1. История реализации продукции партнёром

На данной странице пользователь может изучить историю реализации продукции партнёром, а также:

* Вернуться на главную страницу по кнопке "Назад".

Интерфейс данной страницы представлен на рисунке 3.

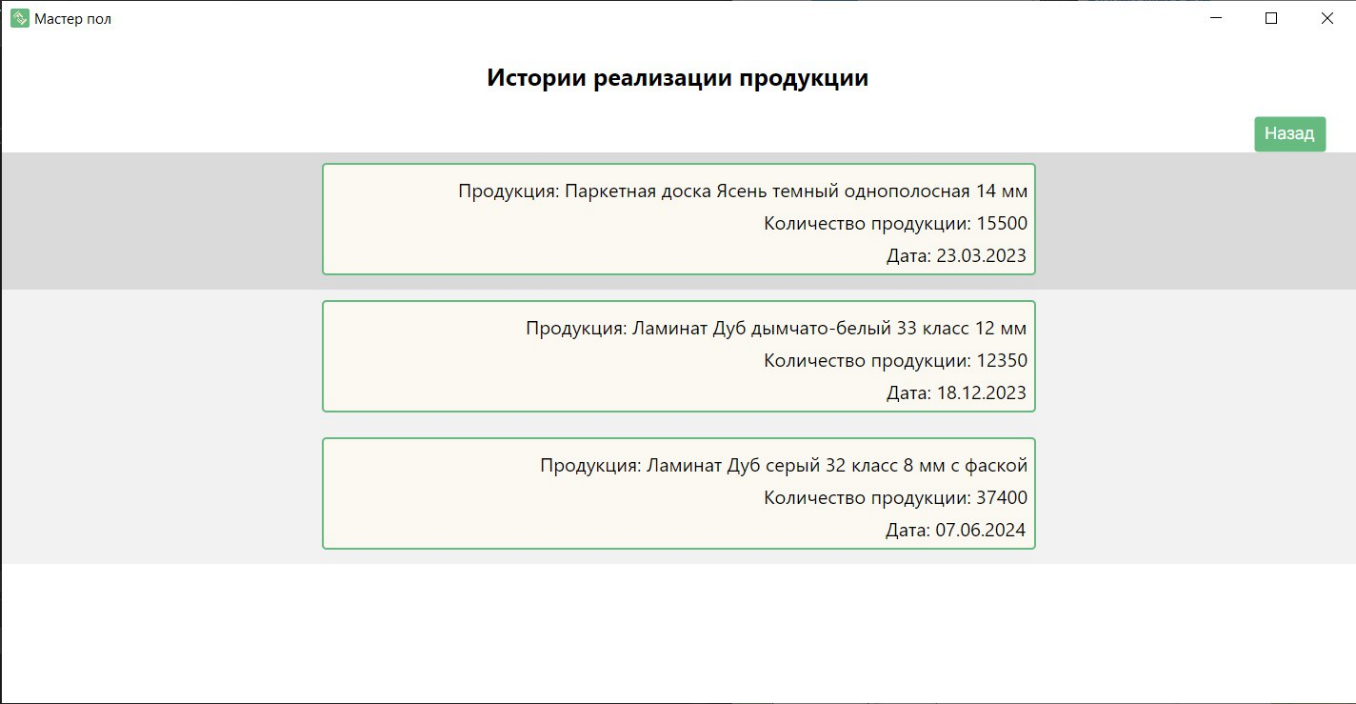


Рисунок №3 - История реализации продукции партнёром

1. Редактирование данных партнёра

На данной странице пользователь может изменить данные уже существующего партнёра, а также:

* Вернуться на главную страницу по кнопке "Назад".

Интерфейс данной страницы представлен на рисунке 4.

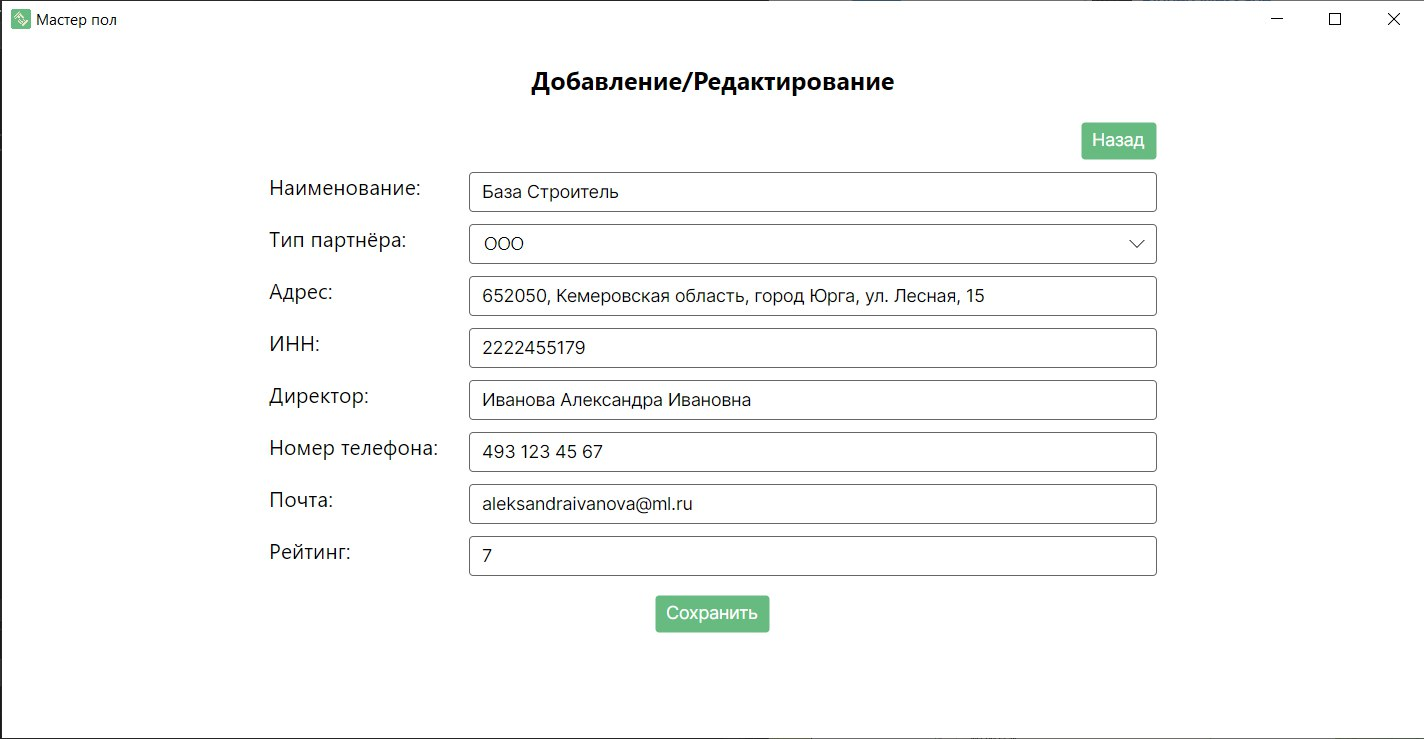


Рисунок №4 - Редактирование данных партнёра

1. Добавления нового партнёра

На данной странице пользователь может добавить нового партнёра, а также:

* Вернуться на главную страницу по кнопке "Назад".

Интерфейс данной страницы представлен на рисунке 5.

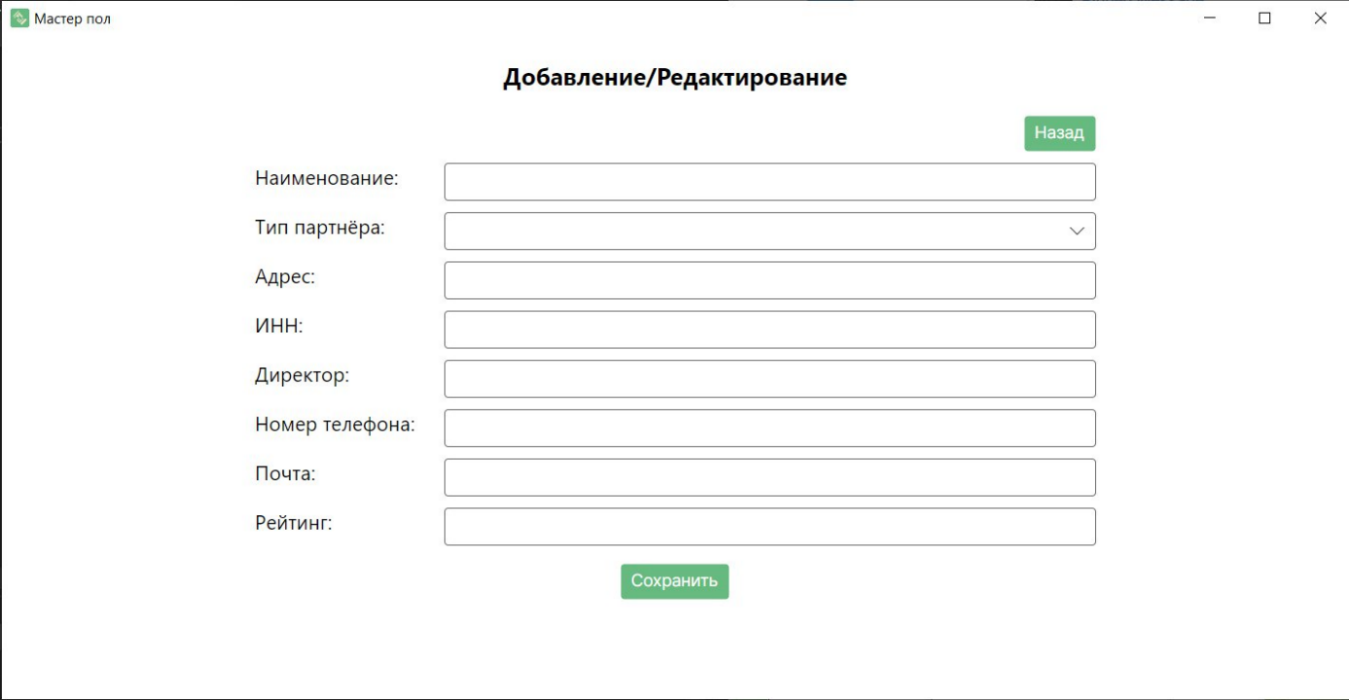


Рисунок №5 - Добавления нового партнёра

Также для функции редактирования были разработана тестовая документация, состоящая из пяти тест-кейсов. Рассмотреть её можно в приложении Б.

## Осуществление интеграции программных модулей

С целью обеспечить одинаковый расчет количества материала, требуемого для производства продукции, был разработан метод.

Метод принимает идентификатор типа продукции, идентификатор типа материала, количество получаемой продукции – целые числа, параметры продукции (два параметра) – вещественные, положительные числа, а возвращать целое число – количество необходимого материала с учетом возможного брака материала.

Метод рассчитывает целое количество материала, необходимого для производства указанного количества продукции, учитывая возможный брак материала. Количество необходимого материала на одну единицу продукции рассчитывается как произведение параметров продукции, умноженное на коэффициент типа продукции. Кроме того, нужно учитывать процент брака материала в зависимости от его типа: с учетом возможного брака материала необходимое количество материала должно быть увеличено.

Коэффициент типа продукции и процент брака – вещественные числа.

Код метода представлен в приложении А в листинге А3.

Также для данной библиотеки были разработаны 10 простых и 5 сложных юнит-тестов. Они представлены в приложении А в листинге А4.

## Проектирование

В дополнение к основным задачам были спроектированы две UML-диаграммы для следующей предметной области: программа для фитнес-центра по распределению фитнес – расписания и контроля его соблюдения.

Диаграмма использования:

Эта диаграмма помогает визуализировать основные функции приложения и взаимодействие между участниками, улучшая пользовательский опыт.

Диаграмма использования для фитнеса приложения включает в себя несколько ключевых участников: пользователя, тренера и администратора. Пользователь регистрируется в приложении, ищет тренеров, просматривает их профили, выполняет тренировки и просматривает свою статистику по занятиям. Тренер, в свою очередь, может принимать заявки от пользователей и отклонять их, просматривать результаты тренировок своих учеников. Администратор регистрирует тренеров и предоставляет скидки постоянным клиентам.

Данная диаграмма представлена в приложении В на рисунке В1.

Диаграмма деятельности:

Диаграмма деятельности визуализирует последовательность действий и взаимодействие между подсистемами.

Диаграмма деятельности для ситуации, когда администратор регистрирует тренера в приложении, включает три подсистемы: администратор, приложение и база данных.

Процесс начинается с того, что администратор инициирует регистрацию тренера в приложении. Затем администратор вводит необходимые данные. После этого приложение проверяет корректность введенных данных. Если данные корректны, приложение отправляет запрос на сохранение информации в базу данных. База данных обрабатывает этот запрос и сохраняет данные о тренере. После успешного сохранения база данных возвращает подтверждение приложению, которое, в свою очередь, уведомляет администратора о завершении регистрации. Если данные некорректны, приложение сообщает об ошибке администратору, который может внести изменения и повторить процесс.

Данная диаграмма представлена в приложении В на рисунке В2

# ЗАКЛЮЧЕНИЕ

# ПРИЛОЖЕНИЕ А

Листинг А1 – Скрипт создания таблиц базы данных

|  |
| --- |
| -- public.materialstype определение  -- Drop table  -- DROP TABLE public.materialstype;  CREATE TABLE public.materialstype (  materialtypeid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  title varchar(30) NOT NULL,  perofdefects numeric NOT NULL,  CONSTRAINT materialstype\_pkey PRIMARY KEY (materialtypeid)  );  -- public.partnerstype определение  -- Drop table  -- DROP TABLE public.partnerstype;  CREATE TABLE public.partnerstype (  partnerstypeid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  title varchar(100) NOT NULL,  CONSTRAINT partnerstype\_pkey PRIMARY KEY (partnerstypeid)  );  -- public.productstype определение  -- Drop table  -- DROP TABLE public.productstype;  CREATE TABLE public.productstype (  producttypeid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  title varchar(50) NOT NULL,  CONSTRAINT productstype\_pkey PRIMARY KEY (producttypeid)  );  -- public.salespoints определение  -- Drop table  -- DROP TABLE public.salespoints;  CREATE TABLE public.salespoints (  salepointsid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  title varchar(100) NOT NULL,  address varchar(300) NOT NULL,  CONSTRAINT salespoints\_pkey PRIMARY KEY (salepointsid)  );  -- public.materials определение  -- Drop table  -- DROP TABLE public.materials;  CREATE TABLE public.materials (  material int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  materialtypeid int4 NULL,  title varchar(300) NOT NULL,  CONSTRAINT materials\_pkey PRIMARY KEY (material),  CONSTRAINT materials\_materialtypeid\_fkey FOREIGN KEY (materialtypeid) REFERENCES public.materialstype(materialtypeid)  );  -- public.partners определение  -- Drop table  -- DROP TABLE public.partners;  CREATE TABLE public.partners (  partnersid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  partnerstypeid int4 NOT NULL,  title varchar(100) NOT NULL,  legaladdress varchar(300) NOT NULL,  inn varchar(100) NOT NULL,  head varchar(150) NOT NULL,  phonenumber varchar(30) NOT NULL,  email varchar(100) NOT NULL,  logo varchar(300) NULL,  rating int4 NULL,  CONSTRAINT partners\_pkey PRIMARY KEY (partnersid),  CONSTRAINT partners\_partnerstypeid\_fkey FOREIGN KEY (partnerstypeid) REFERENCES public.partnerstype(partnerstypeid)  );  -- public.partnerssalespoints определение  -- Drop table  -- DROP TABLE public.partnerssalespoints;  CREATE TABLE public.partnerssalespoints (  partnerssalespointsid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  salepoint int4 NOT NULL,  partnersid int4 NOT NULL,  CONSTRAINT partnerssalespoints\_pkey PRIMARY KEY (partnerssalespointsid),  CONSTRAINT partnerssalespoints\_partnersid\_fkey FOREIGN KEY (partnersid) REFERENCES public.partners(partnersid),  CONSTRAINT partnerssalespoints\_salepoint\_fkey FOREIGN KEY (salepoint) REFERENCES public.salespoints(salepointsid)  );  -- public.products определение  -- Drop table  -- DROP TABLE public.products;  CREATE TABLE public.products (  productid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  article varchar(30) NULL,  producttype int4 NULL,  title varchar(300) NOT NULL,  description varchar(600) NULL,  productimage varchar(300) NULL,  mincostforpartner numeric NOT NULL,  height numeric NULL,  width numeric NULL,  CONSTRAINT products\_pkey PRIMARY KEY (productid),  CONSTRAINT products\_producttype\_fkey FOREIGN KEY (producttype) REFERENCES public.productstype(producttypeid)  );  -- public.productsmaterials определение  -- Drop table  -- DROP TABLE public.productsmaterials;  CREATE TABLE public.productsmaterials (  productsmaterialsid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  materialid int4 NOT NULL,  productid int4 NOT NULL,  CONSTRAINT productsmaterials\_pkey PRIMARY KEY (productsmaterialsid),  CONSTRAINT productsmaterials\_materialid\_fkey FOREIGN KEY (materialid) REFERENCES public.materials(material),  CONSTRAINT productsmaterials\_productid\_fkey FOREIGN KEY (productid) REFERENCES public.products(productid)  );  -- public.partnersproducts определение  -- Drop table  -- DROP TABLE public.partnersproducts;  CREATE TABLE public.partnersproducts (  partnersproductsid int4 GENERATED ALWAYS AS IDENTITY( INCREMENT BY 1 MINVALUE 1 MAXVALUE 2147483647 START 1 CACHE 1 NO CYCLE) NOT NULL,  productsid int4 NOT NULL,  partnersid int4 NOT NULL,  numberofproducts int4 NOT NULL,  saledate date NOT NULL,  CONSTRAINT partnersproducts\_pkey PRIMARY KEY (partnersproductsid),  CONSTRAINT partnersproducts\_partnersid\_fkey FOREIGN KEY (partnersid) REFERENCES public.partners(partnersid),  CONSTRAINT partnersproducts\_productsid\_fkey FOREIGN KEY (productsid) REFERENCES public.products(productid)  ); |

Листинг А2 - Скрипт заполнения данными таблиц базы данных

|  |
| --- |
| INSERT INTO public.materialstype (title,perofdefects) VALUES  ('Тип материала 1',0.1),  ('Тип материала 2',0.95),  ('Тип материала 3',0.28),  ('Тип материала 4',0.55),  ('Тип материала 5',0.34);  INSERT INTO public.partners (partnerstypeid,title,legaladdress,inn,head,phonenumber,email,logo,rating) VALUES  (2,'Паркет 29','164500, Архангельская область, город Северодвинск, ул. Строителей, 18','3333888520','Петров Василий Петрович','987 123 56 78','vppetrov@vl.ru','',7),  (3,'Стройсервис','188910, Ленинградская область, город Приморск, ул. Парковая, 21','4440391035','Соловьев Андрей Николаевич','812 223 32 00','ansolovev@st.ru','',7),  (4,'Ремонт и отделка','143960, Московская область, город Реутов, ул. Свободы, 51','1111520857','Воробьева Екатерина Валерьевна','444 222 33 11','ekaterina.vorobeva@ml.ru','',5),  (1,'МонтажПро','309500, Белгородская область, город Старый Оскол, ул. Рабочая, 122','5552431140','Степанов Степан Сергеевич','912 888 33 33','stepanov@stepan.ru','',10),  (3,'Ромашка','353340, Тульская область, город Балашиха, ул. Будапештсткая, 65','2811022383','Цветков Даниил Филиппович','783 133 21 79','ruxaddissedu-2686@yopmail.com',NULL,9),  (2,'База Строитель','652050, Кемеровская область, город Юрга, ул. Лесная, 15','2222455179','Иванова Александра Ивановна','493 123 45 67','aleksandraivanova@ml.ru','',7),  (1,'123','12','1','1','1','1',NULL,1);  INSERT INTO public.partnersproducts (productsid,partnersid,numberofproducts,saledate) VALUES  (1,1,15500,'2023-03-23'),  (3,1,12350,'2023-12-18'),  (4,1,37400,'2024-06-07'),  (2,2,35000,'2022-12-02'),  (5,2,1250,'2023-05-17'),  (3,2,1000,'2024-06-07'),  (1,2,7550,'2024-07-01'),  (1,3,7250,'2023-01-22'),  (2,3,2500,'2024-07-05'),  (4,4,59050,'2023-03-20');  INSERT INTO public.partnersproducts (productsid,partnersid,numberofproducts,saledate) VALUES  (3,4,37200,'2024-03-12'),  (5,4,4500,'2024-05-14'),  (3,5,50000,'2023-09-19'),  (4,5,670000,'2023-11-10'),  (1,5,35000,'2024-04-15'),  (2,5,25000,'2024-06-12');  INSERT INTO public.partnerstype (title) VALUES  ('ЗАО'),  ('ООО'),  ('ПАО'),  ('ОАО');  INSERT INTO public.products (article,producttype,title,description,productimage,mincostforpartner,height,width) VALUES  ('8758385',1,'Паркетная доска Ясень темный однополосная 14 мм','','',4456.90,NULL,NULL),  ('8858958',1,'Инженерная доска Дуб Французская елка однополосная 12 мм','','',7330.99,NULL,NULL),  ('7750282',2,'Ламинат Дуб дымчато-белый 33 класс 12 мм','','',1799.33,NULL,NULL),  ('7028748',2,'Ламинат Дуб серый 32 класс 8 мм с фаской','','',3890.41,NULL,NULL),  ('5012543',3,'Пробковое напольное клеевое покрытие 32 класс 4 мм','','',5450.59,NULL,NULL);  INSERT INTO public.productstype (title) VALUES  ('Паркетная доска'),  ('Ламинат'),  ('Пробковое покрытие'); |

Листинг А3 – Метод для рассчёта кол-ва материала

|  |
| --- |
| public int GetQuantityForProduct(int count, float width, float length, int product\_type, int material\_type)  {  if (width <= 0 || length <= 0)  {  throw new ArgumentOutOfRangeException("Длина и ширина должны быть больше 0");  }  if (product\_type <= this.product\_type.Length && product\_type >= 1)  {  if (material\_type <= waste\_material\_type.Length && material\_type >= 1)  {  float result = ((width \* length) \* this.product\_type[product\_type - 1]) \* Convert.ToSingle(count);  result = result + result \* (waste\_material\_type[material\_type - 1] + 1);  return Convert.ToInt32(Math.Ceiling(result));  }  else  {  return -1;  }  }  else  {  return -1;  }  } |

Листинг А4 – Юнит-тесты

|  |
| --- |
| [TestMethod]  public void NonExistentProductType()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 4;  int material\_type = 1;  int except = -1;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(except, actual);  }  [TestMethod]  public void NonExistentMaterialType()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 6;  int except = -1;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(except, actual);  }  [TestMethod]  public void CalculationsAreEqual()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int except = 238991;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(except, actual);  }  [TestMethod]  public void CalculationsIsNotNull()  {  int count = Convert.ToInt32(null);  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int except = 238991;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.IsNotNull(actual);  }  [TestMethod]  public void CalculationsIsInt()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int except;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.IsInstanceOfType(actual, typeof(int));  }  [TestMethod]  public void CalculationsIsNotDouble()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int except;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.IsNotInstanceOfType(actual, typeof(double));  }  [TestMethod]  public void CalculationsIsNotBool()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int except;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.IsNotInstanceOfType(actual, typeof(bool));  }  [TestMethod]  public void CalculationsIsTrue()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  bool except = false;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  if (actual != 0 && actual > 0)  {  except = true;  }  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.IsTrue(except);  }  [TestMethod]  public void CalculationsInRange()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int low = 238990;  int high = 238993;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  NUnit.Framework.Assert.That(actual, Is.InRange(low, high));  }  [TestMethod]  public void CalculationsNotInRange()  {  int count = 15;  float width = 20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  int low = 114145;  int high = 114146;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  NUnit.Framework.Assert.That(actual, Is.Not.InRange(low, high));  }  [TestMethod]  public void CalculationsWidthExeption()  {  int count = 15;  float width = 0;  float length = 45;  int product\_type = 3;  int material\_type = 1;  Calc calc = new Calc();  NUnit.Framework.Assert.Throws<ArgumentOutOfRangeException>(() => calc.GetQuantityForProduct(count, width, length, product\_type, material\_type));  }  [TestMethod]  public void CalculationsLengthExeption()  {  int count = 15;  float width = 20;  float length = 0;  int product\_type = 3;  int material\_type = 1;  Calc calc = new Calc();  NUnit.Framework.Assert.Throws<ArgumentOutOfRangeException>(() => calc.GetQuantityForProduct(count, width, length, product\_type, material\_type));  }  [TestMethod]  public void CalculationsCheckMaterialType1()  {  int count = 10;  float width = 1;  float length = 1;  int product\_type = 1;  int material\_type = 1;  Calc calc = new Calc();  int actual = calc.GetQuantityForProduct(count, width, length, product\_type, material\_type);  int except = 12;  Microsoft.VisualStudio.TestTools.UnitTesting.Assert.AreEqual(except, actual);  }  [TestMethod]  public void CalculationsNegativeLengthExeption()  {  int count = 15;  float width = 20;  float length = -45;  int product\_type = 3;  int material\_type = 1;  Calc calc = new Calc();  NUnit.Framework.Assert.Throws<ArgumentOutOfRangeException>(() => calc.GetQuantityForProduct(count, width, length, product\_type, material\_type));  }  [TestMethod]  public void CalculationsNegativeWidthExeption()  {  int count = 15;  float width = -20;  float length = 45;  int product\_type = 3;  int material\_type = 1;  Calc calc = new Calc();  NUnit.Framework.Assert.Throws<ArgumentOutOfRangeException>(() => calc.GetQuantityForProduct(count, width, length, product\_type, material\_type));  } |

# ПРИЛОЖЕНИЕ Б

Аннотация теста

|  |  |
| --- | --- |
| **Название проекта** | Suhareva\_UP |
| **Рабочая версия** | 1.0 |
| **Имя тестирующего** | Сухарева Анжелика Кирилловна |
| **Дата(ы) теста** | 05.03.2025 |

Таблица № 1 - Тестовый пример #1

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_1 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Ввод отрицательного рейтинга |
| **Краткое изложение теста** | При редактировании данных о партнёре замена рейтинга на отрицательное число. |
| **Этапы теста** | 1. Запуск программы; 2. Переход с главной страницы на страницу редактирования; 3. Замена рейтинга на отрицательное число; 4. Сохранение данных. |
| **Тестовые данные** | Наименование: «Ромашка»  Юр. адрес: 353340, Тульская область, город Балашиха, ул. Будапештсткая, 65  ИНН: 2811022383  Директор: Цветков Даниил Филиппович  Номер телефона: 783 133 21 79  Почта: ruxaddissedu-2686@yopmail.com  Рейтинг: -1 |
| **Ожидаемый результат** | Выводится ошибка о том, что рейтинг не может быть отрицательным. |
| **Фактический результат** | Выводится ошибка о том, что рейтинг не может быть отрицательным. |
| **Статус** | Зачёт |

Продолжение таблицы №1

|  |  |
| --- | --- |
| **Предварительное условие** | Существующий партнер должен быть добавлен в базу данных. Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица № 2 - Тестовый пример #2

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_2 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Ввод рейтинга, который больше 10 |
| **Краткое изложение теста** | При редактировании данных о партнёре замена рейтинга на число, большее 10. |
| **Этапы теста** | 1. Запуск программы; 2. Переход с главной страницы на страницу редактирования; 3. Замена рейтинга на число, большее 10; 4. Сохранение данных. |
| **Тестовые данные** | Наименование: «Ромашка»  Юр. адрес: 353340, Тульская область, город Балашиха, ул. Будапештсткая, 65  ИНН: 2811022383  Директор: Цветков Даниил Филиппович  Номер телефона: 783 133 21 79  Почта: ruxaddissedu-2686@yopmail.com  Рейтинг: 11 |
| **Ожидаемый результат** | Выводится ошибка о том, что рейтинг не может быть больше 10. |
| **Фактический результат** | Выводится ошибка о том, что рейтинг не может быть больше 10. |
| **Статус** | Зачёт |
| **Предварительное условие** | Существующий партнер должен быть добавлен в базу данных. Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица № 3 - Тестовый пример #3

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_3 |
| **Приоритет тестирования** | Средний |
| **Заголовок/название теста** | Ввод ИНН, который больше 10 |
| **Краткое изложение теста** | При редактировании данных о партнёре замена рейтинга на отрицательное число. |
| **Этапы теста** | 1. Запуск программы; 2. Переход с главной страницы на страницу редактирования; 3. Замена рейтинга на число, большее 10; 4. Сохранение данных. |
| **Тестовые данные** | Наименование: «Ромашка»  Юр. адрес: 353340, Тульская область, город Балашиха, ул. Будапештсткая, 65  ИНН: 28110223831  Директор: Цветков Даниил Филиппович  Номер телефона: 783 133 21 79  Почта: ruxaddissedu-2686@yopmail.com  Рейтинг: 9 |
| **Ожидаемый результат** | Выводится ошибка о том, что ИНН не может быть больше 10. |
| **Фактический результат** | Выводится ошибка о том, что ИНН не может быть больше 10. |
| **Статус** | Зачёт |
| **Предварительное условие** | Существующий партнер должен быть добавлен в базу данных. Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица № 4 - Тестовый пример #4

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_4 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Редактирование существующих данных о партнере |

Продолжение таблицы №4

|  |  |
| --- | --- |
| **Краткое изложение теста** | Проверка возможности редактирования существующих данных о партнере. |
| **Этапы теста** | 1. Запуск программы; 2. Переход с главной страницы на страницу редактирования; 3. Редактирование данных; 4. Сохранение данных. |
| **Тестовые данные** | Наименование: «Ромашка»  Юр. адрес: 353340, Тульская область, город Балашиха, ул. Будапештсткая, 65  ИНН: 2811022383  Директор: Цветков Даниил Филиппович  Номер телефона: 783 133 21 79  Почта: ruxaddissedu-2686@yopmail.com  Рейтинг: 10 |
| **Ожидаемый результат** | Появляется сообщение, уточняющее действие пользователя. После подтверждения изменения успешно сохранены, происходит переход на главную страницу, где отображаются обновленные данные. В случае отмены действий, данные не сохраняются. Пользователь остаётся на странице редактирования. |
| **Фактический результат** | Появляется сообщение, уточняющее действие пользователя. После подтверждения изменения успешно сохранены, происходит переход на главную страницу, где отображаются обновленные данные. В случае отмены действий, данные не сохраняются. Пользователь остаётся на странице редактирования. |
| **Статус** | Зачёт |
| **Предварительное условие** | Существующий партнер должен быть добавлен в базу данных. Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

Таблица № 5 - Тестовый пример #5

|  |  |
| --- | --- |
| **Тестовый пример #** | TC\_UI\_5 |
| **Приоритет тестирования** | Высокий |
| **Заголовок/название теста** | Проверка полей на пустые значения |

Продолжение таблицы №5

|  |  |
| --- | --- |
| **Краткое изложение теста** | Проверка валидации полей при редактировании данных о партнере. |
| **Этапы теста** | 1. Запуск программы; 2. Переход с главной страницы на страницу редактирования; 3. Редактирование данных; 4. Сохранение данных. |
| **Тестовые данные** | Наименование: «»  Юр. адрес: «»  ИНН: «»  Директор: «»  Номер телефона: «»  Почта: «»  Рейтинг: «» |
| **Ожидаемый результат** | Выводится ошибка о том, что данные строки не могут быть заполнены пустыми значениями. Сообщение, уточняющее действие пользователя не появляется. Переход на другие страницы не происходит. |
| **Фактический результат** | Выводится ошибка о том, что данные строки не могут быть заполнены пустыми значениями. Сообщение, уточняющее действие пользователя не появляется. Переход на другие страницы не происходит. |
| **Статус** | Зачёт |
| **Предварительное условие** | Перед выполнением программы IDE не выдаёт сообщений об ошибках, соблюдена вся логика проектирования библиотеки. |
| **Постусловие** | После завершения тестирования система не должна выдавать сообщений об ошибках или исключений. При следующих запусках сохраняется корректная работа программы. |
| **Примечания/комментарии** | - |

# ПРИЛОЖЕНИЕ В

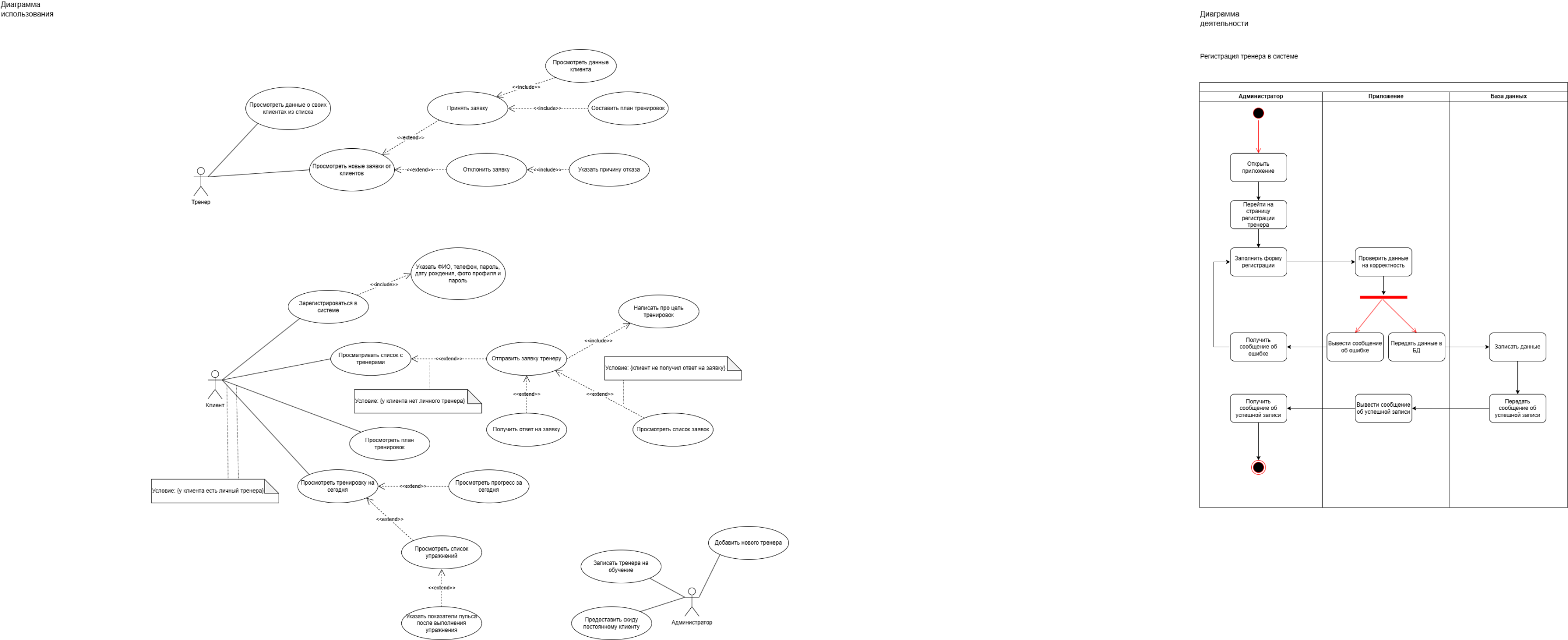


Рисунок № В1 – Диаграмма использования

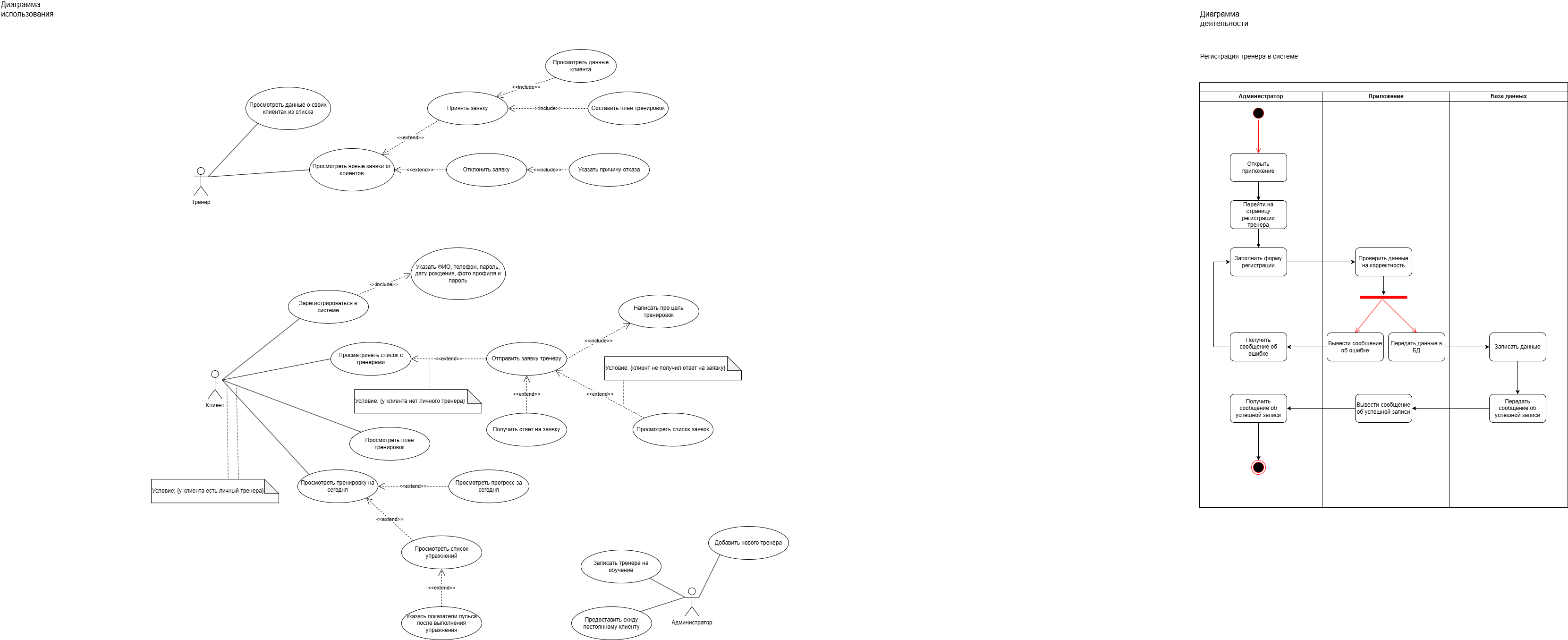


Рисунок № В2 – Диаграмма деятельности