**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ Частное**

**учреждение ВЫСШЕГО образования**

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ**

**УНИВЕРСИТЕТ «СИНЕРГИЯ»**

**Кафедра цифровой экономики**

**КУРСОВАЯ РАБОТА ПО ДИСЦИПЛИНЕ “ТЕХНОЛОГИЯ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ”**

Выполнил:

Обучающийся учебной группы VДКИП-111прог Трухин Глеб Олегович

**(***фамилия, имя, отчество)*

Проверил:

руководитель практики от колледжа:

*(И.О. Фамилия)*

**Владивосток, Москва 2025 г.**

Содержание:

Введение……………………………………………………………………..3

1. Теоретическая часть……………………………………………………...5

Практическая часть………………………………………………………..10

Цель работы………………………………………………………………..10

Ход работы и схемы……………………………………………………….11

Вывод……………………………………………………………………….26

Список использованной литературы……………………………………..27

Введение

Процесс современной разработки программного обеспечения ориентирован на жизненный цикл программного продукта. Все существующие в настоящее время технологии, методики и стандарты напрямую или косвенно касаются или регламентируют этапы жизненного цикла, как по функциональному наполнению, так и по содержанию.

Процесс разработки программных систем тесно связан с областью управления проектами, потому что любой программный продукт является уникальным результатом. От организации этого процесса напрямую зависят основные характеристики выполнения программного проекта – сроки выполнения, запланированный бюджет, качество выпускаемого продукта.

Но профессиональное управление проектами само по себе не может обеспечить достижение указанных характеристик. Немаловажную роль в этом играет архитектура программной системы, опыт и квалификация участников команды разработки, а также правильное документирование всех процессов разработки программного обеспечения.

Существуют различные определения технологии разработки программного обеспечения. К наиболее распространенным относятся следующие.

**Технология разработки программного обеспечения (ПО) заключается в комплексе мер по созданию программных продуктов**. Она включает в себя этапы разработки, методологии, инструменты и примеры проектов.

**Основные задачи:**

* **Решение конкретных задач пользователей**. Например, создание приложения для доставки еды, которое позволяет быстро заказывать блюда.
* **Автоматизация процессов**. ПО может автоматизировать рутинные задачи, которые раньше выполнялись вручную.
* **Обработка и анализ данных**. ПО может обрабатывать большие объёмы данных, находить закономерности и предоставлять полезную информацию.
* **Обеспечение безопасности**. ПО используют для защиты данных и предотвращения кибератак.

**Этапы разработки**

* **Анализ требований**. Собираются данные о цели и функционале будущего продукта, создаётся техническое задание.
* **Проектирование**. Разрабатывается архитектура ПО, определяются компоненты и связи между ними, создаются схемы базы данных и логика приложения.
* **Разработка**. Программисты пишут код, создают отдельные модули и интегрируют их в единую систему.
* **Тестирование**. Проверяется работоспособность и качество ПО, проводятся функциональное, интеграционное, системное тестирование.
* **Развёртывание**. Разработанное ПО устанавливается и запускается на целевой платформе или инфраструктуре.
* **Обслуживание и поддержка**. Разработчики отслеживают работу ПО, исправляют ошибки, выпускают обновления и предоставляют поддержку пользователей.

**Технология разработки программного обеспечения (ПО)** — это комплекс мер по созданию программных продуктов, который включает в себя этапы, вспомогательные процессы и модели жизненного цикла.

Теоретическая часть.

* 1. Небольшая историческая справка.

**Технология разработки ПО возникла на основе развития программирования**. Первые алгоритмы создавались ещё в древности, а с появлением электронных компьютеров в XX веке началось развитие современного программирования.

Существует множество различных процессов для создания ПО. Тем не менее, технологий, рассматривающих полный жизненный цикл проекта разработки ПО, сочетающих в себе научный подход, серьезную базу исследований и имеющих историю реального использования и адаптации, относительно немного.

За несколько десятилетий эволюции аппаратное обеспечение значительно усовершенствовалось. Вычислительные мощности, которые еще десять-пятнадцать лет назад могли себе позволить лишь немногие научные учреждения и обслуживание которых требовало целого штата специалистов, сегодня доступны практически каждому инженеру.

Однако эти мощности требуют соответствующего программного обеспечения. И именно в этой области, несмотря на то что аппаратные ресурсы стали значительно более доступны, наблюдаются значительные проблемы.

Так, по данным американских исследователей, в 80-е только 14% проектов по созданию ПО завершались успешно. Но и сегодня - после нескольких десятилетий эволюции языков программирования, инструментальных средств разработки, при практически неограниченном (по сравнению с 70-ми и 80-ми) машинном времени - процент успешно завершенных проектов составляет всего 26%.

* 1. Технологии.

Термин «технология» – он подчеркивает аналогию между созданием программного продукта и промышленным производством. Он отражает современную тенденцию к вводу дисциплины, организации и инструментирования в такой творческий процесс, как программирование. Слово фиксирует ту точку зрения, что программирование, несмотря на интеллектуальность и творческий характер этой деятельности, нуждается в организации и регламентировании, наборе соглашений и правил, не говоря уже об инструментальном обеспечении. Сейчас это кажется тривиальным, но в 60-е годы такую точку приходилось отстаивать. Да и сейчас порой возникают трения на почве регламентирования деятельности разработчиков. Сам русский термин «технология программирования» был введен русским академиком Андреем Петровичем Ершовым. Он трактовал термин «программирование» в обобщенном виде и подразумевал все виды деятельности, выполняемые в ходе создания программных систем. На западе для определения этой деятельности использовался термин «engineering». Сейчас обобщённый термин, применимый к созданию программных средств, обозначают как «разработка» или «конструирование». Справедлива формула: разработка = анализ + проектирование + программирование (кодирование) + тестирование + отладка Иногда сюда также включают “сопровождение”. Чтобы подчеркнуть промышленно-производственный аспект, говорят о “технологии разработки” или “технологии конструирования”.

* 1. Этапы развития

Лишь в начале 90-х Британское сообщество вычислительной техники (British Computer Society) начало присваивать разработчикам программ звание инженера. В США только в 1998 году стало возможным хоть где-то зарегистрироваться в качестве профессионального инженера программного обеспечения. Но по-прежнему, даже в начале нынешнего века, общепризнанным остается тот факт, что разработке программного обеспечения не достает достаточно развитой научной базы.

1.3. Методы проектирования

Говорят, что Генри Форд совершил революцию в производстве автомобилей, когда заметил, что узлы автомобиля можно стандартизировать, так что при сборке автомобилей данной модели можно будет использовать любой экземпляр требуемого узла.

Столь же важным в настоящее время признается возможность при разработке одних приложений заимствовать идеи, архитектуру, проект и исходный код других приложений. Если приложения проектируются таким образом, что различные их части могут быть использованы многократно, то в конечном итоге это приводит к уменьшению стоимости разработки приложений.

Однако, чтобы это было возможным, приложения должны быть модульными. Модульность приложения, собственно, и означает, что оно состоит из легко идентифицируемых и заменяемых частей. По этой причине при правильном проектировании программного продукта особое внимание должно уделяться модульности, особенно на стадии разработки архитектуры.

К формальным методам проектирования относятся те методы, которые основаны на математике. Формальные методы помогают решить задачи обеспечения надежности программ. Они могут быть применены как при анализе требований для обеспечения точности формулировки требований, так и в процессе реализации для обеспечения соответствия кода программы сформулированным требованиям. Как правило формальные методы используют математику в ее логическом аспекте. В вычислительном же аспекте математика задействована в связи с использованием метрик, которые мы будем рассматривать далее.

Сегодня существует огромное количество различных процессов для создания ПО. Тем не менее, именно технологий, рассматривающих полный жизненный цикл проекта разработки ПО, сочетающих в себе научный подход, серьезную базу исследований и имеющих историю реального использования и адаптации, относительно немного. Из методологий и технологий, получивших определенное признание на данный момент, можно назвать следующие: Datarun, CMM, Microsoft Solution Framework (MSF), Oracle Method, Rational Unified Process (RUP), SADT (IDEFx).

Особое место в этом списке занимает технология компании Rational Software. В ее методологии применен наиболее современный процессно ориентированный подход: так как разработка ПО является производством, то, как и на всяком производстве, при выявлении проблем в продукции (симптомов) необходимо корректировать процесс (устранять причины). Особенностью этой технологии является то, что в ее создании участвуют ведущие методисты в области разработки ПО.

1.4. Этапы и элементы процесса разработки.

В 80-е и 90-е в области разработки ПО преобладали две тенденции. Одна – это быстрый рост приложений, в том числе создаваемых для Web. Другая – расцвет инструментальных средств и парадигм (подходов к проектированию).

Несмотря на появление новых тенденций, основные этапы разработки ПО остались неизменными: 

А. Определение процесса разработки ПО; 

Б. Управление проектом разработки; 

В. Описание целевого программного продукта; 

Г. Проектирование продукта; 

Д. Разработка продукта; 

Е. Тестирование частей; 

Ж. Интеграция частей и тестирование продукта в целом; 

З. Сопровождение продукта.

Персонал – те, кем это делается. Команда разработчиков наилучшим образом работает, если каждый участник знает, что он должен делать, и имеет определенные обязанности. Другая сторона аспекта персонала – это заинтересованные в проекте лица: заказчиками, пользователи и инвесторы.

1.5. Инструментарий технологии программирования.

Инструментарий технологии программирования – совокупность программ и программных комплексов, обеспечивающих технологию разработки, отладки и внедрения создаваемых программных продуктов.

1.6. Средства для создания приложений.

Средства для создания приложений – локальные средства, обеспечивающие выполнение отдельных видов работ по созданию программ, делятся на:  языки и системы программирования; 

инструментальная среда пользователя.

Язык программирования – формализованный язык для описания алгоритма решения задачи на компьютере.

Они делятся на классы: 

машинные языки – языки программирования, воспринимаемые аппаратной частью компьютера (машинные коды); 

машинно-ориентированные языки – языки программирования, которые отражают структуру конкретного типа компьютера (ассемблеры);

алгоритмические языки – не зависящие от архитектуры компьютера языки программирования для отражения структуры алгоритма (Паскаль, бейсик, Фортран и др.); 

процедурно–ориентированные языки – языки программирования, где имеется возможность описания программы как совокупности процедур (подпрограмм). 

проблемно–ориентированные языки – предназначены для решения задач определенного класса (Lisp);

ПРАКТИЧЕСКАЯ ЧАСТЬ

Цель работы

Исходя из теории, была выполнена и лабораторная работа. Целью этой работы является:

Здесь будет изучаться процесс технологии разработок программного обеспечения.

Предмет исследования - технологии разработки программного обеспечения, лабораторные работы от 1 до 10

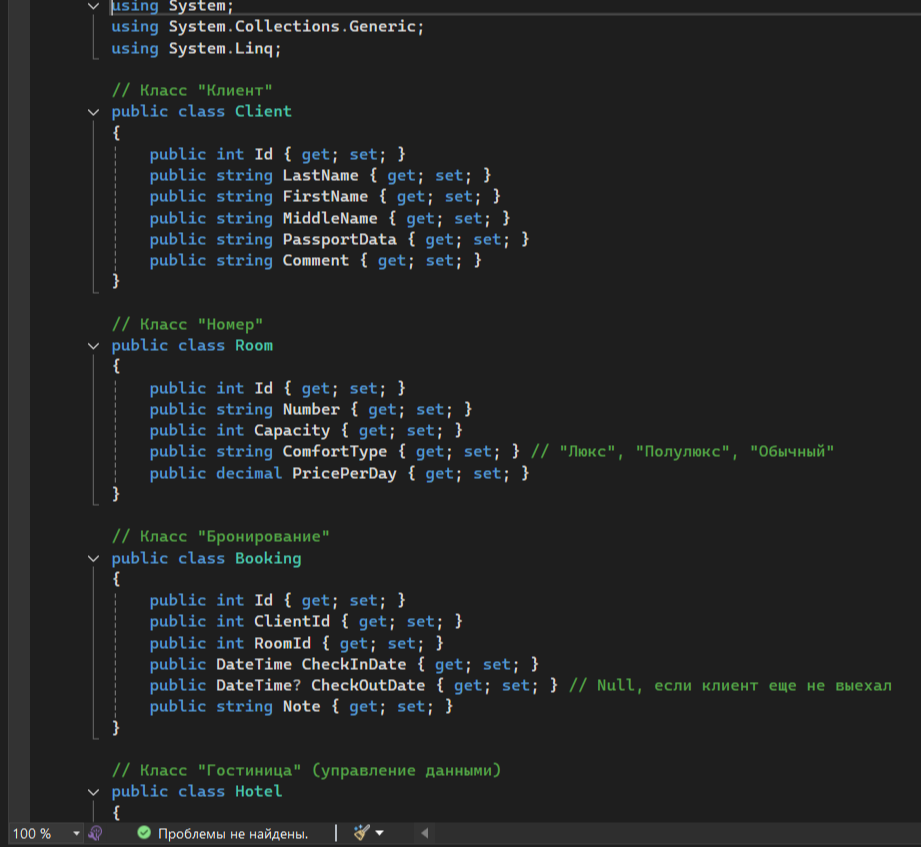
В этом случае будут подробно описаны коды для всех лабораторных работ от 1 до 10.

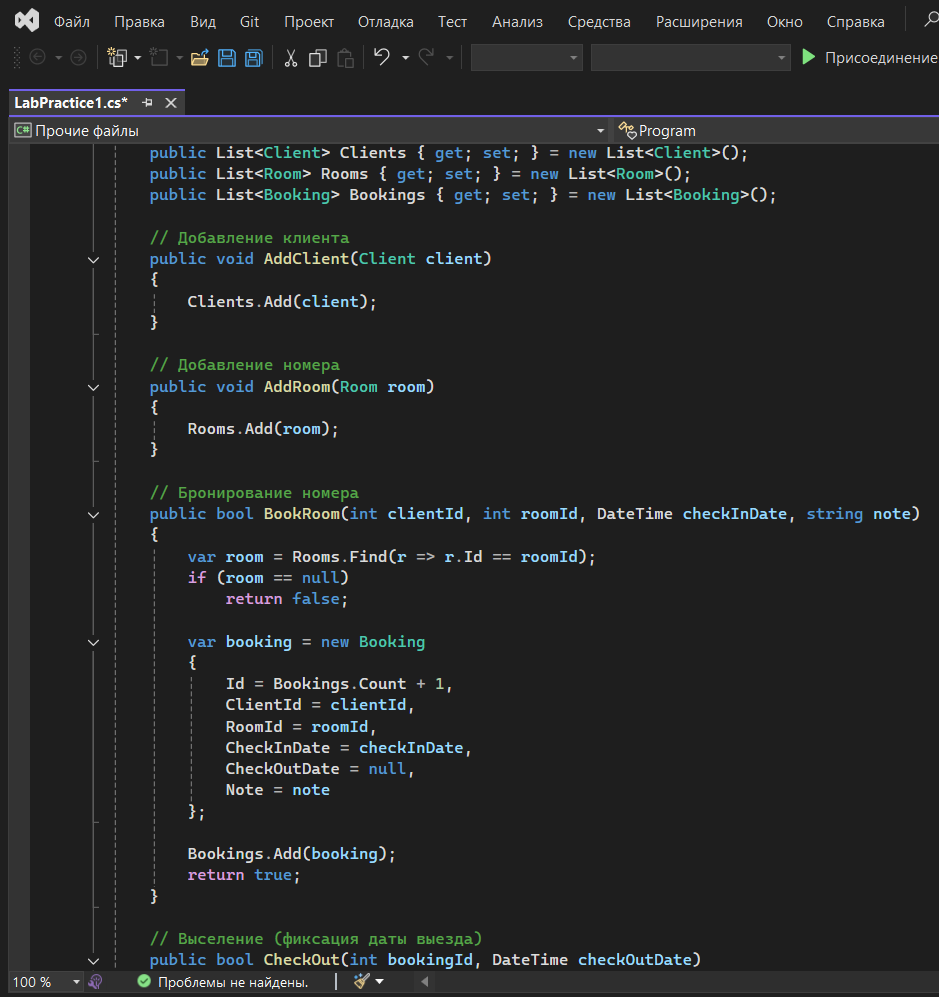
Планируется получить хорошие, добросовестные и положительные результаты от этой работы, также узнать как правильно делать коды для лабораторных работ, чтобы научиться этому и в дальнейшем.

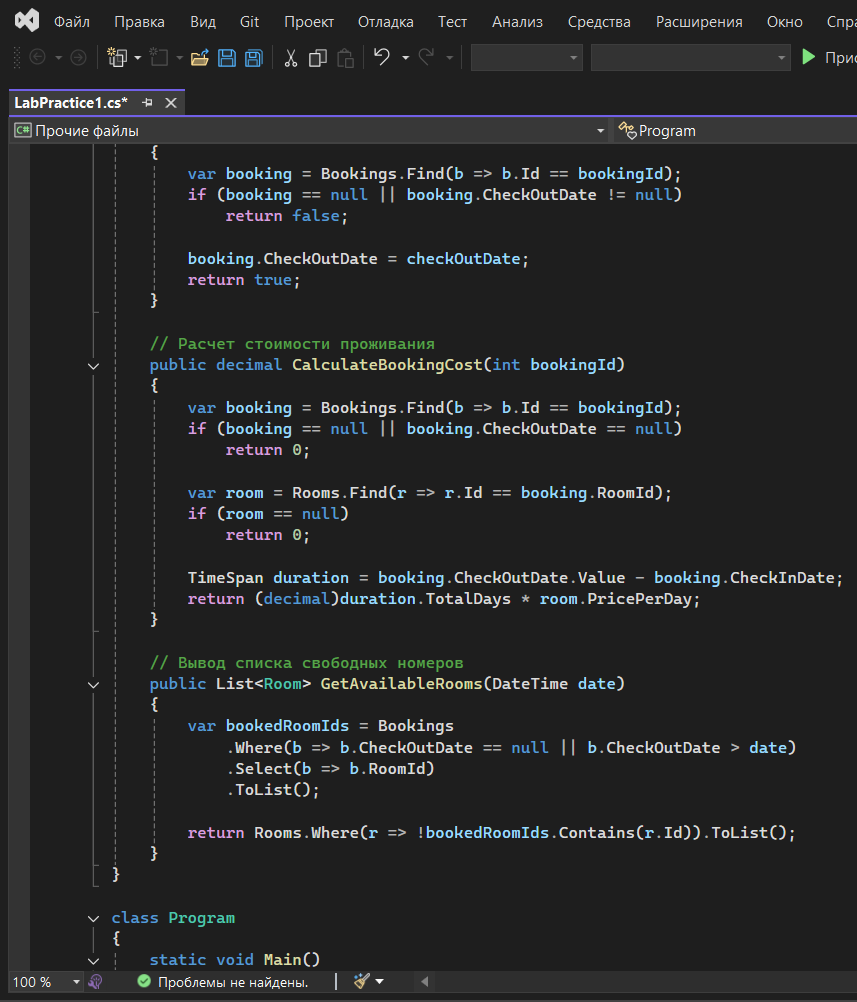
Ход Работы

Вот как были сделаны лабораторные работы от 1 до 10.

Лабораторная Работа 1









Лабораторная работа 2.

Здесь было создано техническое задание на разработку программного обеспечения по ГОСТу. Написано техническое задание.

**ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**на разработку программного обеспечения**  
**«Система автоматизации гостиничного бизнеса»**

**1. Основание для разработки**

Документ составлен в соответствии с:

* **ГОСТ 34.602-89** «Техническое задание на создание автоматизированной системы»
* **ГОСТ 19.201-78** «Техническое задание. Требования к содержанию и оформлению»
* Внутренними регламентами заказчика

Заказчик: **ООО «Гостиничный комплекс "Престиж"»**  
Исполнитель: **ООО «ИТ-Решения»**

**2. Назначение и цели создания системы**

**2.1 Назначение**  
Разработка программного обеспечения (ПО) для автоматизации процессов:

* Бронирования номеров
* Учета клиентов
* Управления занятостью номеров
* Формирования финансовой отчетности

**2.2 Цели**

* Повышение эффективности работы гостиницы на **30%**
* Сокращение времени обработки заявок с **1 часа до 10 минут**
* Уменьшение количества ошибок при ручном вводе данных

**3. Требования к системе**

**3.1 Функциональные требования**

| **№** | **Функция** | **Описание** |
| --- | --- | --- |
| 1 | Управление бронированием | Онлайн-бронирование, изменение/отмена броней, уведомления клиентов |
| 2 | Ведение клиентской базы | Хранение данных гостей (ФИО, паспорт, контакты, история посещений) |
| 3 | Управление номерами | Учет статусов (свободен/занят/на уборке), категорий, цен |
| 4 | Финансовый учет | Формирование счетов, отчетов по выручке, интеграция с платежными системами |
| 5 | Аналитика | Визуализация загруженности номеров, финансовых показателей |

**3.2 Нефункциональные требования**

* **Производительность**:
  + Обработка ≥100 бронирований в минуту
  + Время отклика интерфейса <1 сек
* **Надежность**:
  + Работа 24/7, время простоя ≤0.1% в год
  + Автоматическое резервное копирование данных 2 раза в сутки
* **Безопасность**:
  + Аутентификация по логину/паролю + 2FA для администраторов
  + Шифрование персональных данных (ГОСТ Р 34.10-2012)

**3.3 Требования к совместимости**

* **ОС**: Windows 10+, Linux, macOS
* **СУБД**: PostgreSQL 14+
* **Браузеры**: Chrome, Edge, Firefox (последние версии)

**4. Этапы разработки**

| **Этап** | **Срок** | **Результат** |
| --- | --- | --- |
| Анализ требований | 1 мес. | ТЗ, Use-Case-диаграммы |
| Проектирование | 2 мес. | Архитектурные схемы, прототип интерфейса |
| Реализация | 4 мес. | Рабочая версия ПО |
| Тестирование | 1 мес. | Отчет по тестам, исправление ошибок |
| Внедрение | 1 мес. | Инструкции, обучение персонала |

**5. Требования к документации**

* **Программная документация** (по ГОСТ 19.101-77):
  + Руководство администратора
  + Руководство пользователя
  + Описание API (для интеграций)
* **Эксплуатационные документы**:
  + План развертывания
  + Регламент резервного копирования

**6. Порядок контроля и приемки**

**6.1 Тестирование**

* Проверка всех функций по чек-листу (≥95% успешных тестов)
* Нагрузочное тестирование (JMeter)

**6.2 Приемка**

* Подписание акта сдачи-приемки после успешного пилотного внедрения в 1 филиале гостиницы.

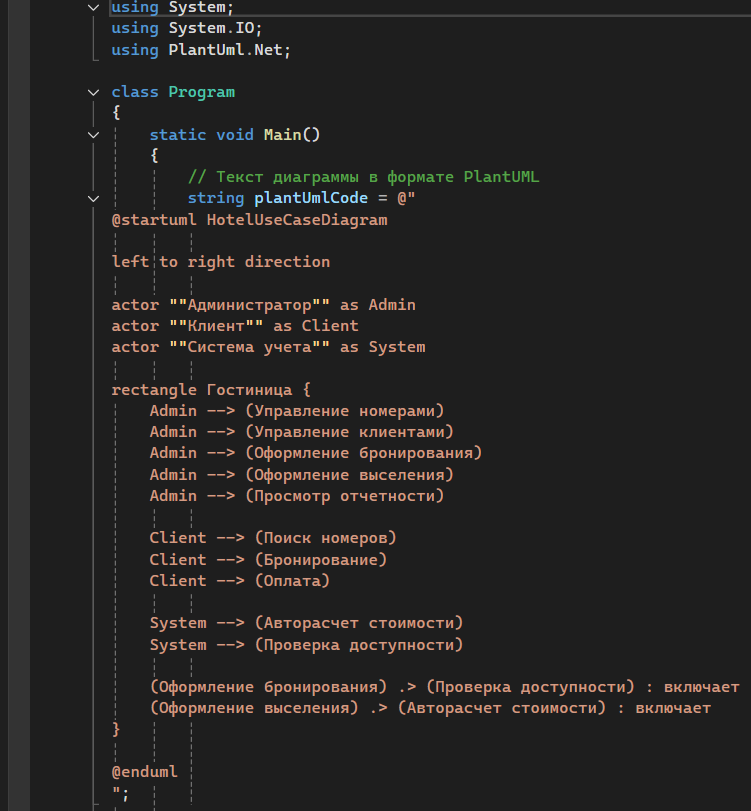
**Приложения**

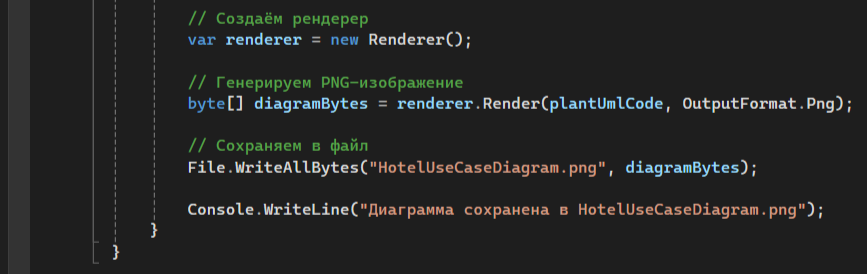
1. **Диаграммы** (IDEF0, DFD, UML)
2. **Макеты интерфейсов** (Figma)
3. **Глоссарий терминов**

**Руководитель проекта от Заказчика**: \_\_\_\_\_\_\_\_\_\_\_/Иванов А.П./  
**Руководитель проекта от Исполнителя**: \_\_\_\_\_\_\_\_\_\_\_/Петров С.И./

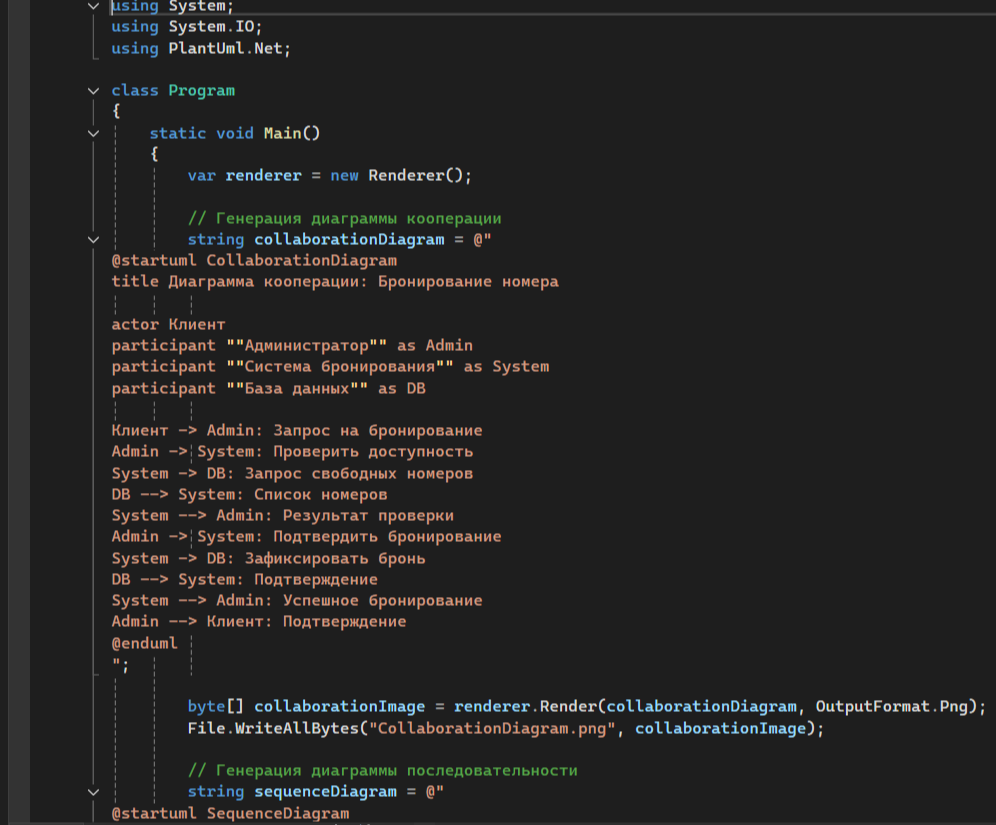
**Дата**: «15» мая 2025 г.

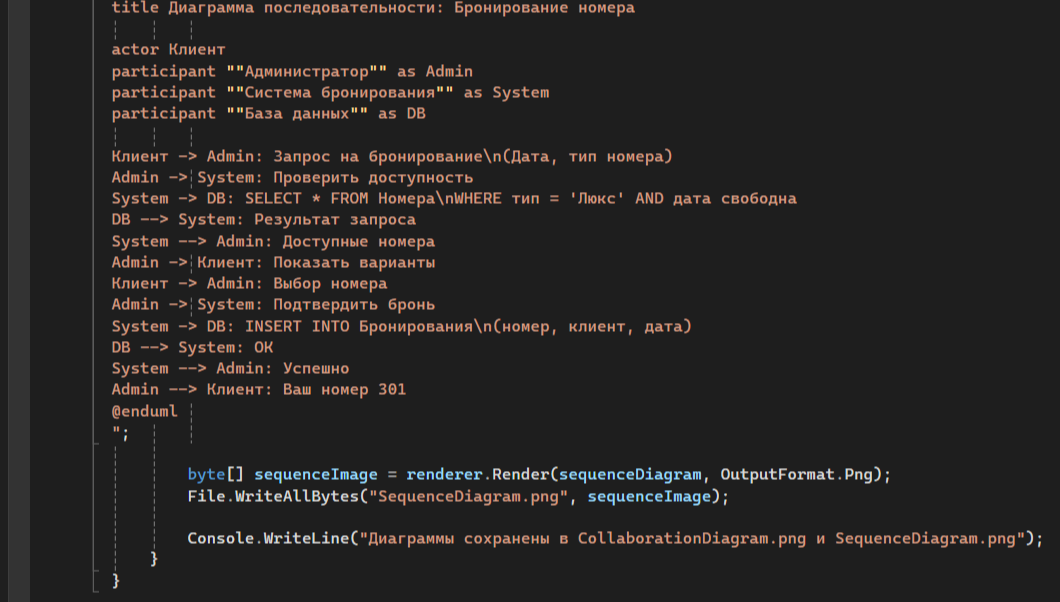
Лабораторная Работа 3.



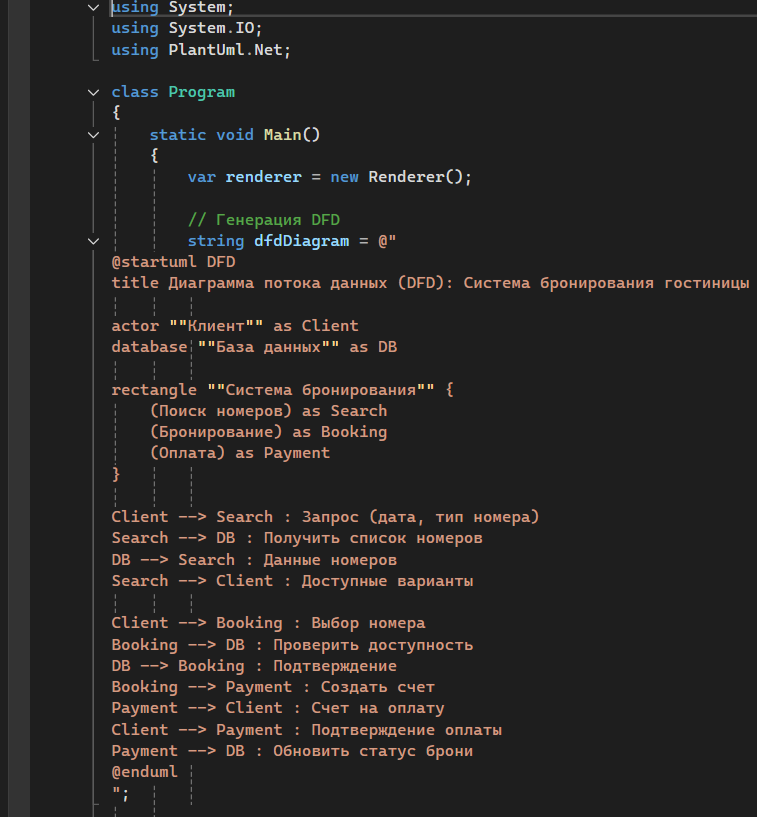


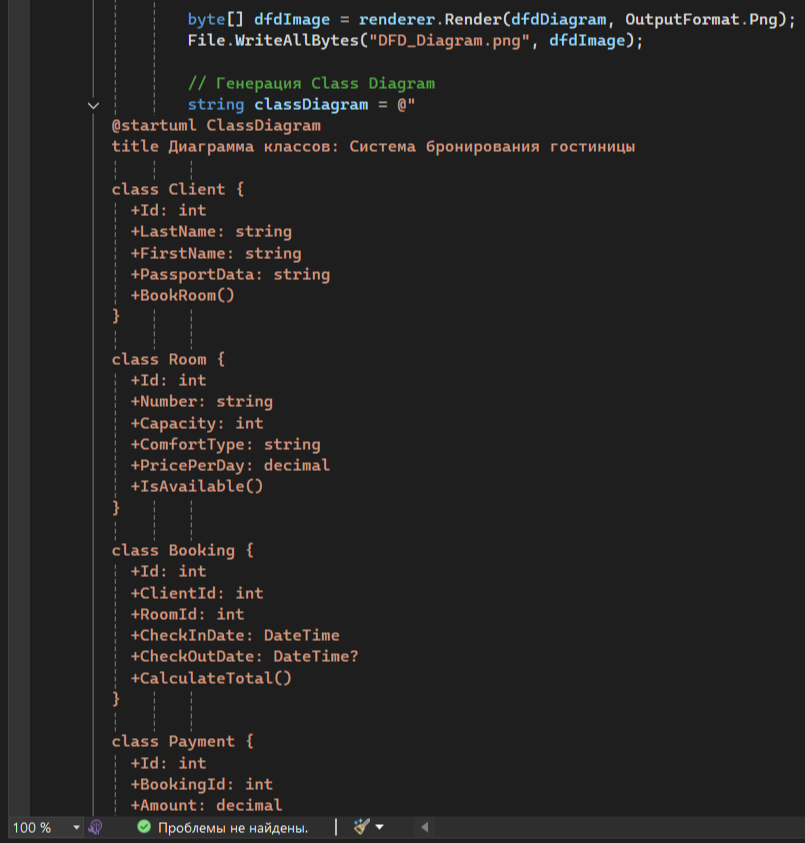
Лабораторная работа 4.

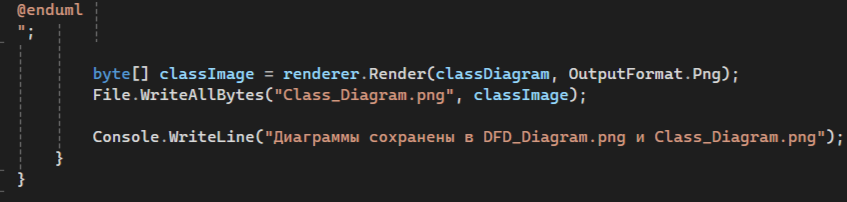




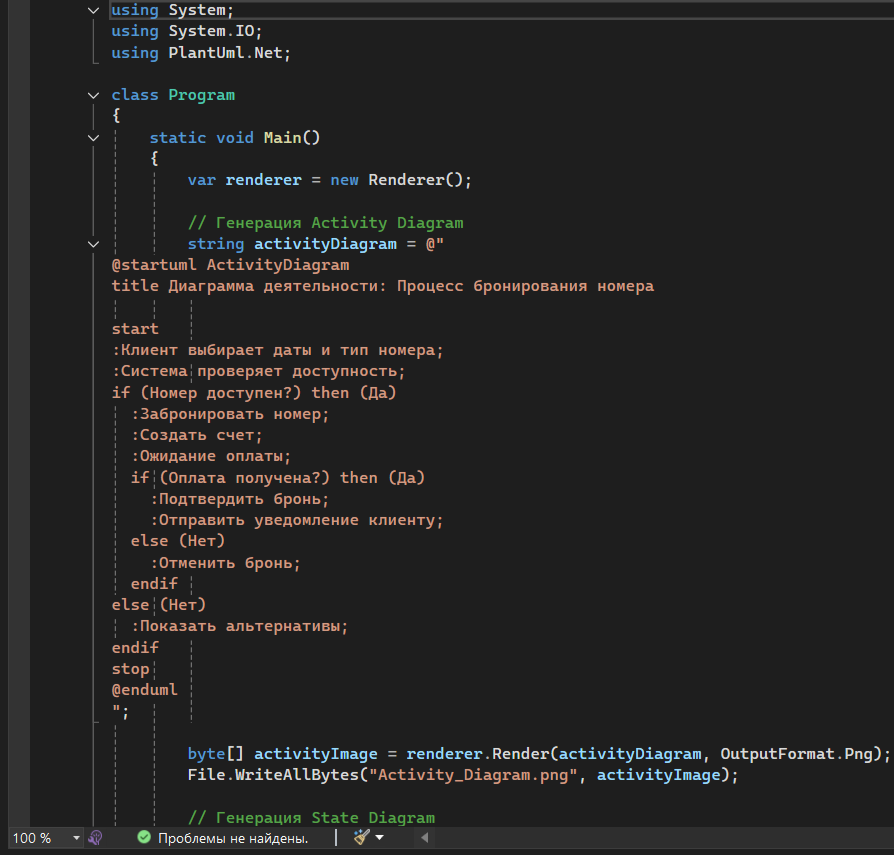
Лабораторная работа 5.

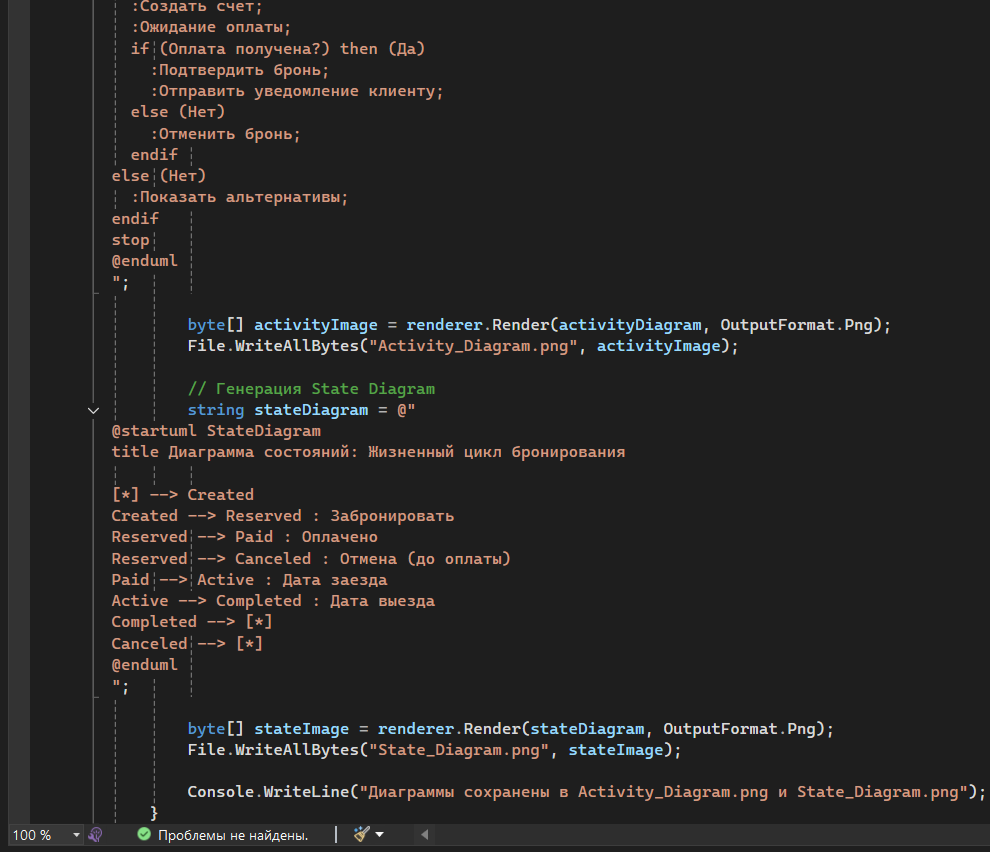




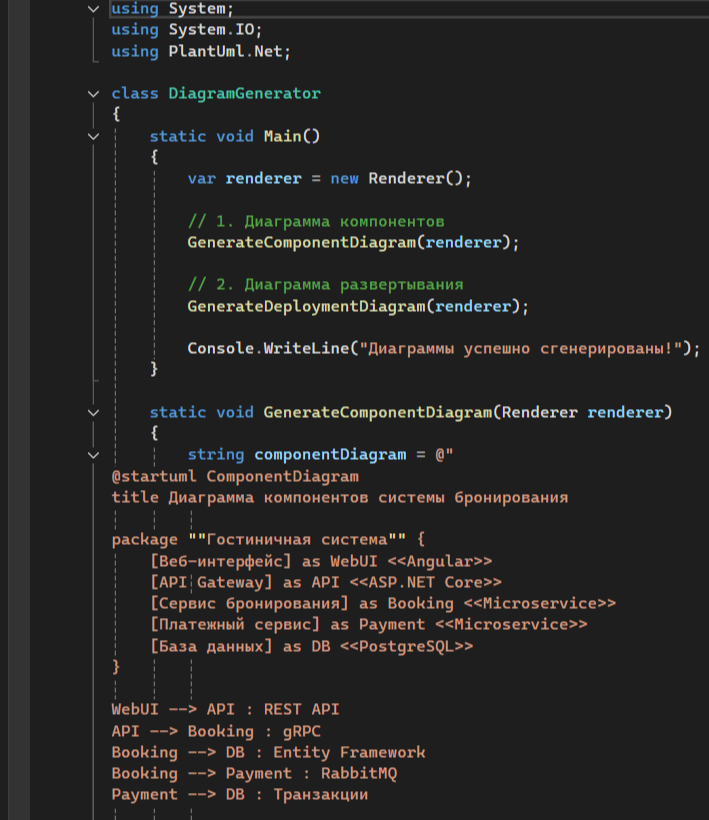


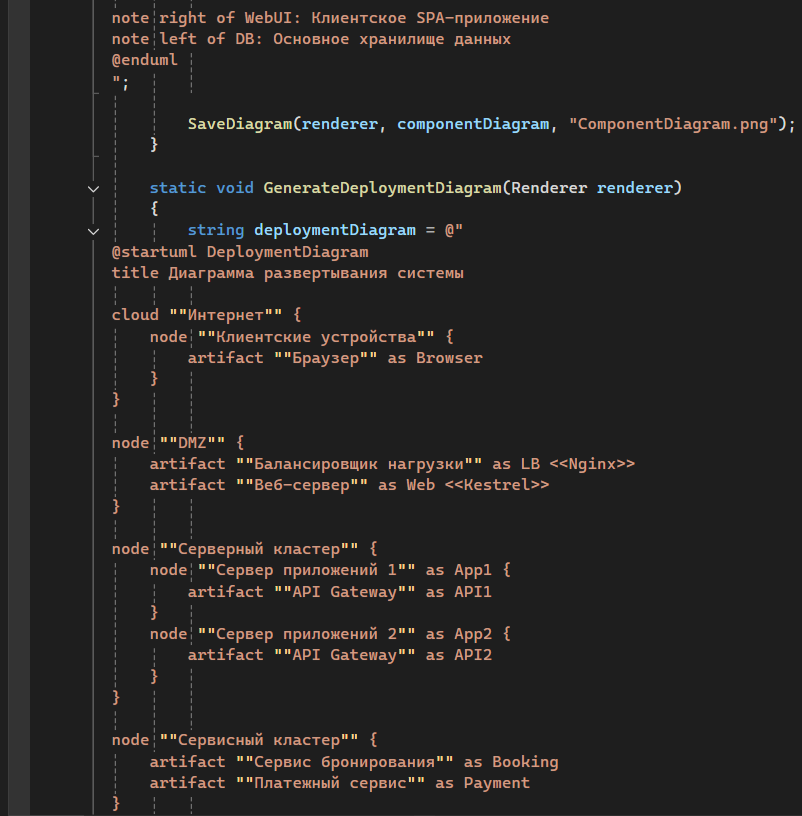
Лабораторная работа 6.

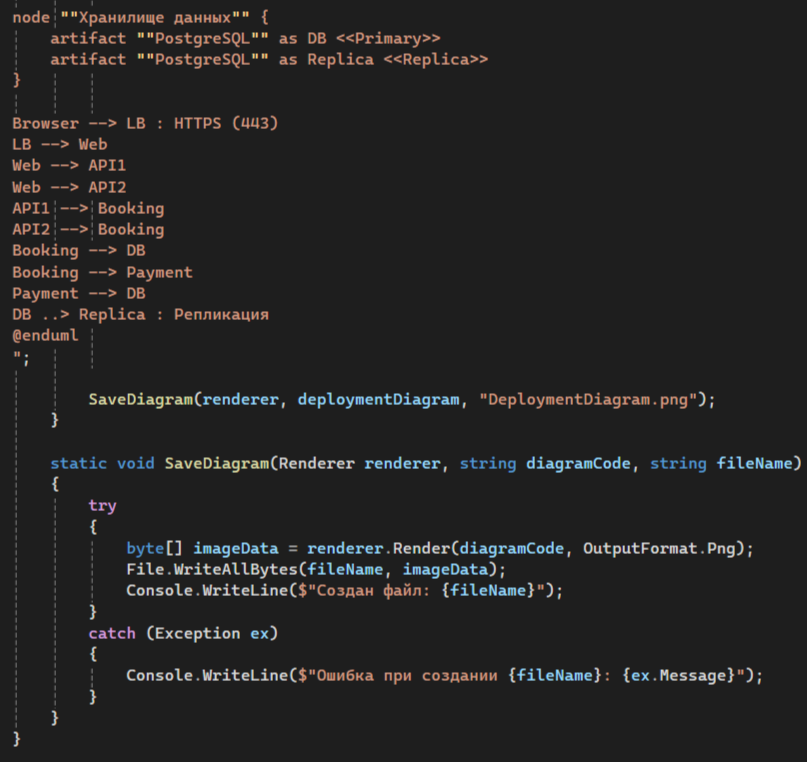




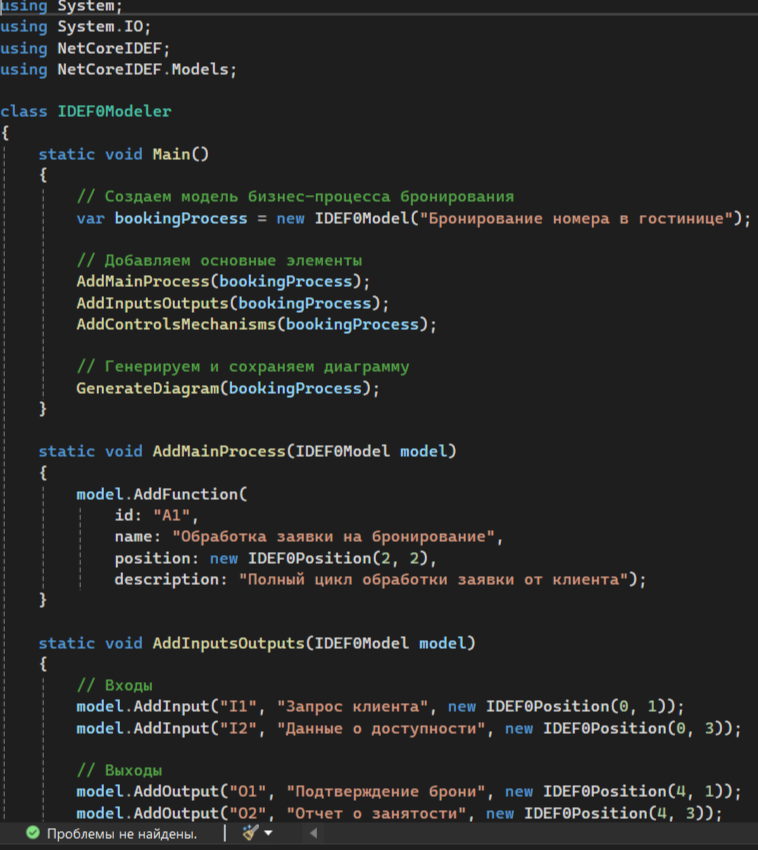
Лабораторная работа 7.





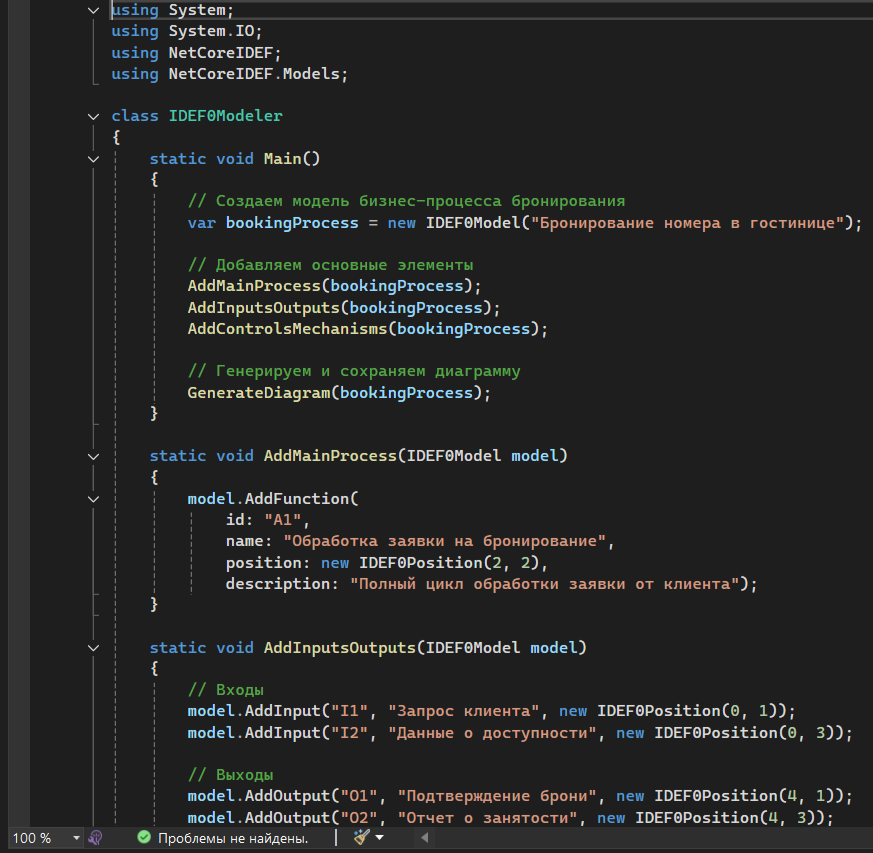


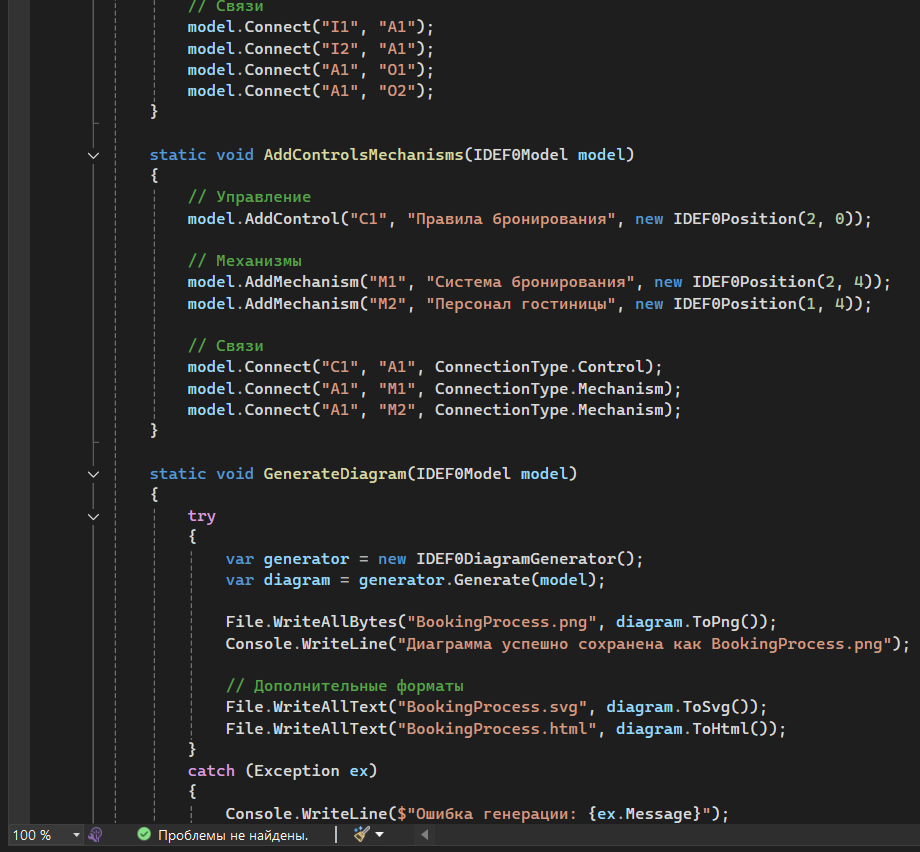
Лабораторная работа 8.



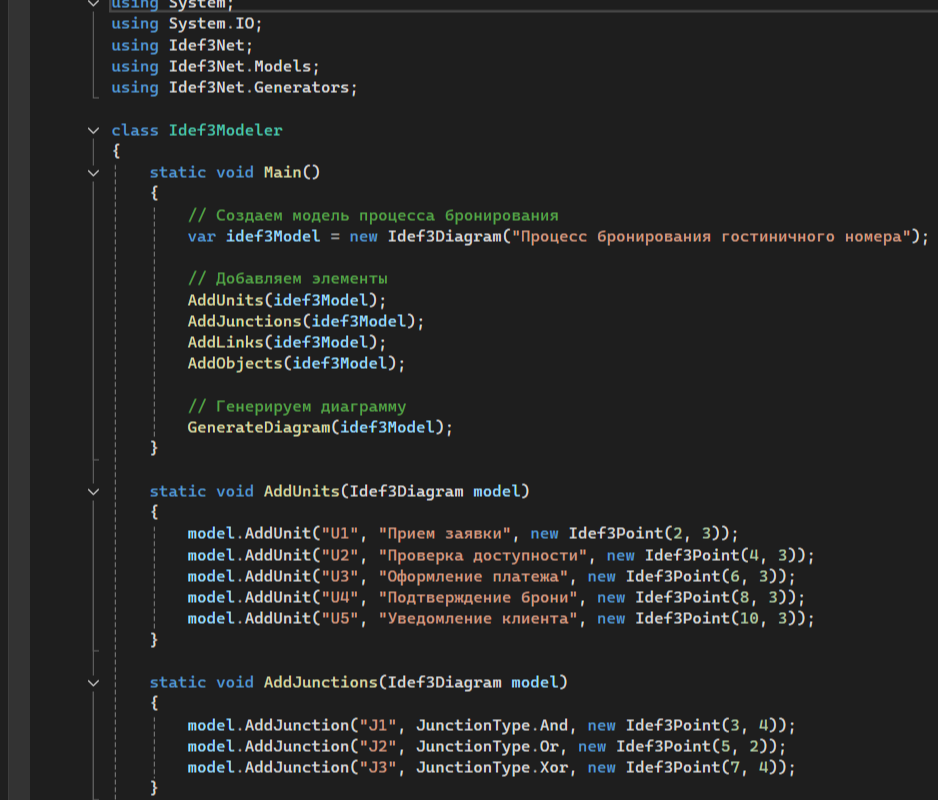


Лабораторная работа 9.

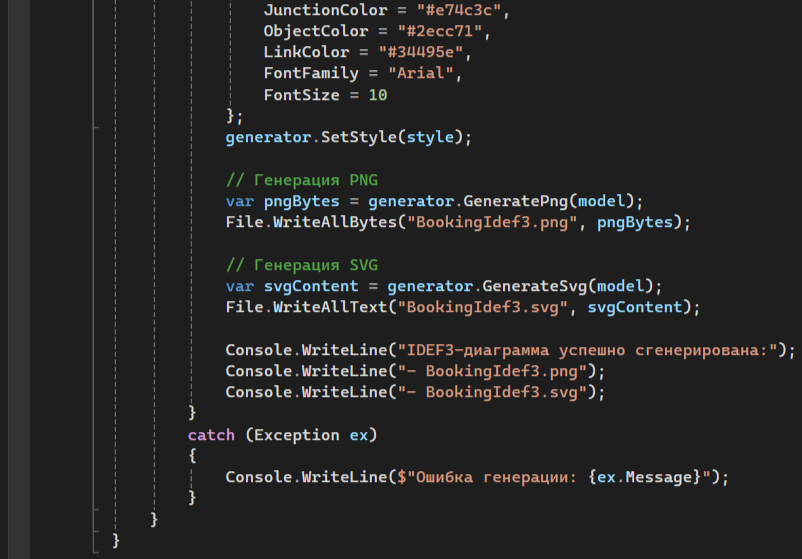




Лабораторная работа 10 (последняя).







Вывод:

Исходя из теорий и практических заданий, можно понять, что технология **разработки программного обеспечения (ПО) включает в себя комплекс мер по созданию программных продуктов**. Эта деятельность включает в себя последовательность этапов, вспомогательные процессы, а также проблемы и перспективы развития.

Ключевым понятием в технологии разработки ПО является **жизненный цикл программного продукта (SDLC, Software Development Life Cycle)**. Он описывает последовательность этапов, которые проходят команды разработчиков от идеи до завершения и поддержки ПО.

Основные этапы жизненного цикла:

1. **Планирование**. Определяются цели проекта, его объём и ресурсы.
2. **Анализ требований**. Собираются и анализируются требования к ПО, включая интервью с заинтересованными сторонами, опросы пользователей и изучение существующих систем.
3. **Проектирование**. Создаётся архитектура системы, интерфейсы, проектируется база данных.
4. **Разработка**. Разработчики создают программный продукт на основе спецификаций, определённых на предыдущих этапах.
5. **Тестирование**. Проводится модульное, интеграционное и системное тестирование, чтобы выявить ошибки и проверить соответствие продукта требованиям.
6. **Развёртывание**. После успешного тестирования ПО внедряется в рабочую среду, включая установку на серверы, настройку окружения и обучение пользователей.
7. **Поддержка и обслуживание**. Включает исправление ошибок, обновления и улучшения, а также ответы на вопросы пользователей.

Подводя итоги, стоит отметить, что технология разработки программного обеспечения - довольно сложный, но очень интересный предмет.