|  |
| --- |
| Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования «Новгородский государственный университет имени  Ярослава Мудрого»  ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ  ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ |

|  |
| --- |
| Утверждаю: |
| Зам. директора по УМ и ВР |

ПРОИЗВОДСТВЕННАЯ ПРАКТИКА

ПМ.02 ОСУЩЕСТВЛЕНИЕ ИНТЕГРАЦИИ   
ПРОФЕССИОНАЛЬНЫХ МОДУЛЕЙ

09.02.07 Информационные системы и программирование

код и наименование специальности

ОТЧЁТ

Руководитель от организации:

/ А.С. Латышев

«30» июня 2025 г.

Руководитель от колледжа:

/ Л.Н. Цымбалюк

«30» июня 2025 г.

Выполнил:

Студент гр. 3992

/ К.А. Рыжов

«30» июня 2025 г.

# Содержание

[Содержание 2](#_Toc202379531)

[Постановка задачи на практику 3](#_Toc202379532)

[1. Инструкция по технике безопасности на рабочем месте 4](#_Toc202379533)

[1.1. Меры безопасности перед началом работы за компьютером 4](#_Toc202379534)

[1.2. Меры безопасности во время работы за компьютером 4](#_Toc202379535)

[1.3. Меры безопасности после окончания работы за компьютером 4](#_Toc202379536)

[2. Должностные обязанности специалиста 5](#_Toc202379537)

[3. Перечень и назначение программных средств, установленных на ПК предприятия 6](#_Toc202379538)

[4. Выполнение индивидуального задания 7](#_Toc202379539)

[4.1. Задание на разработку 7](#_Toc202379540)

[4.2. Спецификация программного продукта 8](#_Toc202379541)

[4.3. Руководство пользователя 10](#_Toc202379542)

[4.4. Тестирование 11](#_Toc202379543)

[5. Выполнение задания от руководителя практики от предприятия 14](#_Toc202379544)

[Заключение 15](#_Toc202379545)

[Приложение А 16](#_Toc202379546)

# Постановка задачи на практику

В рамках прохождения практики в образовательной организации «Школа 21» практикант был временно включён в учебный процесс, ориентированный на выполнение типовых профессиональных задач, соответствующих квалификации техника-программиста. Основной целью практики было развитие прикладных навыков программирования, отладки и тестирования программного обеспечения, а также участие в проектной деятельности.

В ходе практики планировалось:

* + - * закрепить знания, полученные в процессе обучения;
      * приобрести практический опыт разработки и сопровождения программ;
      * овладеть навыками работы с системами контроля версий;
* научиться взаимодействовать с командами и выполнять проектные задачи в условиях ограниченного времени.

Практика предусматривала выполнение индивидуальных и командных заданий, прохождение внутренней системы проверок (exam/peer review), а также освоение методологии работы с исходным кодом, тестами и документацией.

# 1. Инструкция по технике безопасности на рабочем месте

Во время прохождения практики в «Школе 21» большое внимание уделялось соблюдению техники безопасности при работе за компьютером. Основной целью этих мероприятий было обеспечение безопасности и сохранение здоровья всех участников образовательного процесса.

Вводный инструктаж по технике безопасности включал следующие основные положения:

## 1.1. Меры безопасности перед началом работы за компьютером

* проверить исправность розеток, проводов и кабелей компьютера;
* убедиться в наличии заземления оборудования;
* убедиться, что компьютер включается и функционирует корректно.

## 1.2. Меры безопасности во время работы за компьютером

* прикасаться к компьютеру и системному блоку мокрыми руками;
* класть на корпус компьютера посторонние предметы;
* самостоятельно очищать системный блок от пыли и грязи, если он подключён к сети;
* продолжать работу при обнаружении неисправностей — в этом случае необходимо немедленно прекратить работу и сообщить об инциденте руководителю;
* включать и выключать компьютер из розетки без крайней необходимости, особенно во время рабочего процесса.

## 1.3. Меры безопасности после окончания работы за компьютером

* аккуратно отключить компьютер от электросети;
* убедиться, что оборудование полностью выключено.

Практикант был ознакомлен с данными мерами и подтвердил это через электронную систему организации.

# 2. Должностные обязанности специалиста

В соответствии с квалификационными требованиями к должности техника-программиста, специалист в данной сфере выполняет следующие обязанности:

* разрабатывает и отлаживает прикладные программы для обработки информации;
* участвует в проектировании и внедрении автоматизированных систем;
* составляет техническую и пользовательскую документацию;
* обеспечивает корректную работу программного обеспечения;
* взаимодействует с базами данных, формирует выходные документы;
* участвует в тестировании и сопровождении программных модулей;
* фиксирует объёмы выполненных задач и ведёт отчётность.

Также специалист должен соблюдать нормативы охраны труда, регламенты работы с техникой и действующие методические инструкции по эксплуатации вычислительных средств.

# 3. Перечень и назначение программных средств, установленных на ПК предприятия

В процессе прохождения практики в «Школе 21» практикант работал за персональным компьютером, оснащённым всем необходимым программным обеспечением для выполнения учебных и проектных задач. Операционная система, установленная на рабочих станциях — Linux, что отражает тенденции в индустрии к использованию свободного и надёжного программного обеспечения в разработке.

На компьютерах были установлены следующие ключевые программные средства:

* терминал (Shell) — основной инструмент взаимодействия с операционной системой, используемый для навигации по файловой системе, компиляции, запуска программ и работы с Git;
* Git и GitLab — система контроля версий и веб-сервис для совместной работы над кодом и хранения репозиториев;
* Vim, VSCode, nano — текстовые редакторы и IDE для написания, редактирования и анализа исходного кода;
* GCC — компилятор языка C, используемые для сборки программных проектов;
* браузеры — использовались как для доступа к внутренним образовательным платформам, так и для поиска информации;
* сайт контроля посещаемости, обучения и прогресса — проприетарные инструменты Школы 21, через которые фиксируются выполненные задания и отслеживается результативность работы.

Программное обеспечение подбиралось таким образом, чтобы максимально приблизить среду обучения к условиям, с которыми студент может столкнуться в реальной рабочей обстановке программиста.

# 4. Выполнение индивидуального задания

## 4.1. Задание на разработку

В рамках прохождения практики студенту было выдано индивидуальное задание, связанное с реализацией базового функционала для информационной системы по бронированию билетов в театр. Несмотря на то, что исходное описание предметной области предусматривало более широкие требования к системе, в рамках учебной практики студент сконцентрировался на выполнении упрощённого, но функционального прототипа одной из подсистем — работы с репертуаром спектаклей.

Задание включало разработку настольного приложения с использованием языка программирования C# и технологии Windows Forms. Основная задача состояла в реализации интерфейса для добавления, редактирования, удаления и отображения данных о спектаклях, хранящихся в базе данных MySQL. Программа подключалась к удалённой базе данных и обеспечивала CRUD-функциональность (создание, чтение, обновление, удаление) в отношении таблицы spectacles.

В интерфейсе были реализованы следующие компоненты:

* Поля для ввода названия спектакля, его описания и длительности;
* Таблица с отображением всех записей;
* Кнопки управления: «Добавить», «Изменить», «Удалить»;
* Механизм предварительной подстановки данных в поля ввода при выборе записи из таблицы.

Дополнительно была реализована валидация данных: программа проверяла корректность введённой информации и выводила соответствующие сообщения при ошибках (например, если длительность введена некорректно или не заполнено обязательное поле).

Таким образом, в рамках задания была успешно разработана и протестирована упрощённая подсистема, демонстрирующая основные приёмы взаимодействия с базой данных через C# и Windows Forms. Полученные навыки позволили студенту закрепить знания в области проектирования пользовательского интерфейса, работы с SQL и обработки пользовательского ввода.

## 4.2. Спецификация программного продукта

### 4.2.3 Спецификация модуля управления спектаклями

Модуль управления спектаклями реализован на языке C# в среде Microsoft Visual Studio 2022. Он предназначен для сотрудников театра и предоставляет функциональность для ведения базы данных спектаклей. Модуль позволяет выполнять операции добавления, редактирования и удаления спектаклей, а также отображать текущий список с возможностью выбора и изменения записей.

#### 4.2.3.1 Функциональные возможности модуля

Функции модуля включают:

* отображение таблицы со списком спектаклей;
* добавление нового спектакля с указанием названия, описания и длительности;
* редактирование выделенной записи;
* удаление выбранного спектакля после подтверждения;
* обновление таблицы после выполнения операций;
* автоматическая очистка полей ввода после действий.

#### 4.2.3.2 Взаимодействие с другими модулями

Модуль входит в состав главной формы и может использоваться в связке с модулями бронирования и формирования расписания. Информация о спектаклях может отображаться в интерфейсе бронирования и используется при выборе спектакля для сеанса.

#### 4.2.3.3 Взаимодействие с пользователем

Пользователь работает с вкладкой управления спектаклями. Для добавления необходимо заполнить поля (название, описание, длительность) и нажать кнопку «Добавить». Для редактирования нужно выделить запись, изменить данные и нажать «Изменить». Для удаления — выбрать строку и нажать «Удалить». Все действия сопровождаются диалоговыми окнами с подтверждением или уведомлением об ошибке.

#### 4.2.3.4 Входные данные

* текстовые поля: название спектакля, описание, длительность;
* выделенная строка в таблице (для редактирования или удаления);
* кнопки «Добавить», «Изменить», «Удалить».

#### 4.2.3.5 Выходные данные

* обновлённая таблица со списком спектаклей;
* сообщения об успешном добавлении, редактировании или удалении;
* сообщения об ошибке при некорректном вводе или сбое выполнения запроса.

#### 4.2.3.6 Навигация внутри модуля

* модуль размещён во вкладке основной формы;
* при выделении строки таблицы поля ввода заполняются автоматически;
* действия выполняются без переходов между формами;
* после действий происходит обновление таблицы и очистка полей.

#### 4.2.3.7 Требования к макету формы

* таблица DataGridView с колонками: id, название, описание, длительность;
* три текстовых поля для ввода информации;
* кнопки «Добавить», «Изменить», «Удалить» под полями;
* элементы расположены по центру окна;

Макет формы указан в приложении А под рисунком 5.

#### 4.2.3.8 Обработка ошибок

Модуль обрабатывает следующие ситуации:

* попытка добавить спектакль без указания названия или длительности;
* некорректный формат длительности (нечисловое значение);
* попытка редактировать или удалить запись без выбора строки;
* ошибки при выполнении SQL-запросов (например, потеря соединения).

## 4.3. Руководство пользователя

После запуска приложения открывается окно управления спектаклями. В верхней части формы отображается таблица со списком всех добавленных спектаклей. Ниже расположены поля ввода: название, описание и длительность спектакля, а также кнопки управления: «Добавить», «Изменить» и «Удалить».

Добавление нового спектакля:

* Введите название спектакля в соответствующее поле.
* При необходимости добавьте описание.
* Укажите длительность спектакля в минутах (только числовое значение).
* Нажмите кнопку «Добавить».
* После успешного добавления появится сообщение, поля очистятся, а таблица обновится.

Редактирование существующего спектакля:

* Выделите нужную строку в таблице — поля ввода заполнятся автоматически.
* Внесите изменения в поля: название, описание или длительность.
* Нажмите кнопку «Изменить».
* При успешном обновлении данных появится уведомление и обновится таблица.

Удаление спектакля:

* Выберите строку в таблице.
* Нажмите кнопку «Удалить».
* Подтвердите действие во всплывающем окне
* После удаления таблица обновится автоматически.

Если поля не заполнены или указано некорректное значение (например, нечисловая длительность), система отобразит соответствующее сообщение об ошибке.

## 4.4. Тестирование

Проведено тестирование программного обеспечения с использованием тест-кейсов. Тест-кейсы описывают отдельные действия пользователя и ожидаемые результаты. Каждый тест-кейс проверяет определенный аспект функциональности программы.

Тест-кейс TC01

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC01 |
| Заголовок | | | Проверка с полностью корректными данными |
| Предусловие | | | Объект Spectacle создан с валидными значениями |
| Тестовые данные | | | Название: "Гамлет",  описание: "Трагедия Шекспира",  длительность: 180 |
| Ожидаемый результат | | | true — данные корректны |
| Шаг | Действие | Предполагаемый результат | Pass / Fall |
| 1. | Вызвать IsValid(Spectacle) | Метод возвращает true | Pass |
| Постусловие | | | - |
| Результат | | | Pass |

Тест-кейс TC02

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC02 |
| Заголовок | | | Проверка с пустым названием |
| Предусловие | | | Объект Spectacle создан с пустым названием |
| Тестовые данные | | | Название: "",  описание: "Пьеса",  длительность: 100 |
| Ожидаемый результат | | | false — название не указано |
| Шаг | Действие | Предполагаемый результат | Pass / Fall |
| 1. | Вызвать IsValid(Spectacle) | Метод возвращает false | Pass |
| Постусловие | | | - |
| Результат | | | Pass |

Тест-кейс TC03

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC03 |
| Заголовок | | | Проверка с нулевой длительностью |
| Предусловие | | | Объект Spectacle создан с длительностью 0 |
| Тестовые данные | | | Название: "Ревизор",  описание: "Комедия",  длительность: 0 |
| Ожидаемый результат | | | false — длительность некорректна |
| Шаг | Действие | Предполагаемый результат | Pass / Fall |
| 1. | Вызвать IsValid(Spectacle) | Метод возвращает false | Pass |
| Постусловие | | | - |
| Результат | | | Pass |

Тест-кейс TC04

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC04 |
| Заголовок | | | Пограничный тест — корректная длительность, но пустое название |
| Предусловие | | | Объект Spectacle с одним ошибочным полем |
| Тестовые данные | | | Название: " ",  описание: "Мистика",  длительность: 150 |
| Ожидаемый результат | | | false — невалидное название |
| Шаг | Действие | Предполагаемый результат | Pass / Fall |
| 1. | Вызвать IsValid(Spectacle) | Метод возвращает false | Pass |
| Постусловие | | | - |
| Результат | | | Pass |

Тест-кейс TC05

|  |  |  |  |
| --- | --- | --- | --- |
| Идентификатор | | | TC05 |
| Заголовок | | | Проверка поведения при передаче null |
| Предусловие | | | Метод IsValid вызывается с null |
| Тестовые данные | | | Объект: null |
| Ожидаемый результат | | | false — метод корректно обрабатывает null |
| Шаг | Действие | Предполагаемый результат | Pass / Fall |
| 1. | Вызвать IsValid(null) | Метод возвращает false | Pass |
| Постусловие | | | - |
| Результат | | | Pass |

# 5. Выполнение задания от руководителя практики от предприятия

В рамках прохождения практики на базе образовательной программы «Школа 21» студент выполнял индивидуальные задания, направленные на углублённое изучение языка программирования C. Практика включала в себя выполнение учебных проектов и прохождение внутренних экзаменов, каждый из которых проверял навыки самостоятельного проектирования и написания кода, отладку, тестирование и соблюдение норм кодирования.

Во время практики были последовательно освоены основные темы, включая:

* + работу с типами данных и операциями ввода/вывода;
  + использование управляющих конструкций и функций;
  + указатели, массивы и строки;
  + динамическое распределение памяти;
  + реализацию пользовательских структур данных (связные списки, деревья и др.).

Выполнение заданий сопровождалось применением системы контроля версий, инструментов и автоматизированной проверки кода. Такой подход обеспечивал развитие как технических, так и soft-навыков, включая работу в команде, соблюдение дедлайнов и автономную проработку проблем.

Полученные в ходе практики знания и навыки соответствуют требованиям, предъявляемым к начинающим специалистам в области программирования и разработки программного обеспечения.

# Заключение

Прохождение производственной практики в формате, предусмотренном образовательной программой Школы 21 и колледжем, стало важным этапом в формировании профессиональных навыков и личностных компетенций студента по направлению «Программист-Инженер».

В процессе практики удалось познакомиться с организацией проектной и командной деятельности, а также углубить знания по программированию на языке C. Благодаря индивидуальному заданию была получена практическая реализация небольшого, но законченного модуля информационной системы, включающего основные функции по работе с базой данных. Это способствовало закреплению навыков работы с Windows Forms, SQL-запросами и пользовательским интерфейсом.

Помимо технических умений, практика способствовала развитию самоорганизации, ответственности, а также способности принимать решения в условиях отсутствия жёсткой структуры — что является особенностью образовательной среды Школы 21.

В целом, цели практики были достигнуты, а полученные знания и опыт будут полезны как при дальнейшем обучении, так и при трудоустройстве в сфере информационных технологий.

# Приложение А

(обязательное)

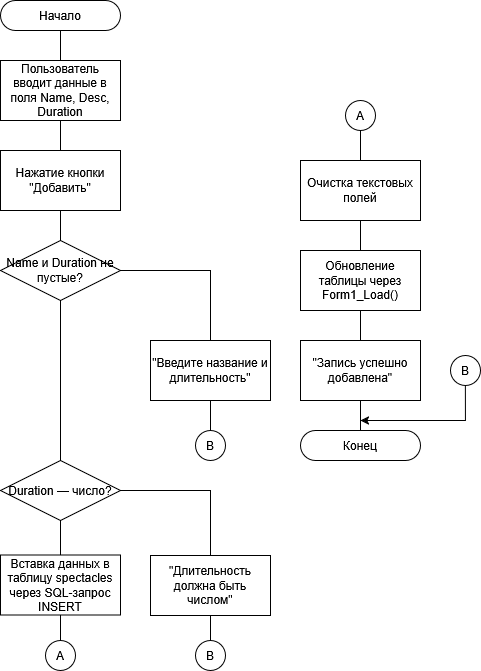


Рисунок А1. Блок-схема Добавление спектакля

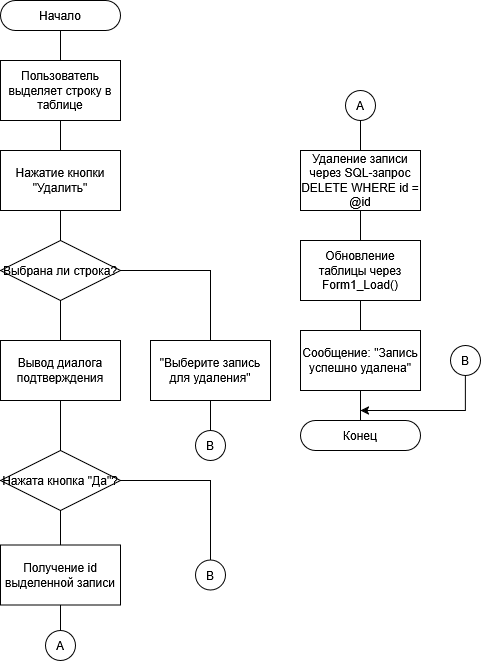


Рисунок А2. Блок-схема Удаление спектакля

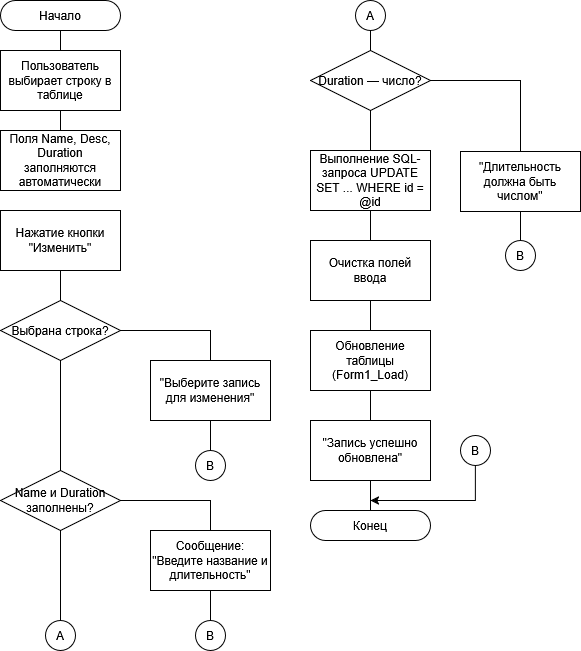


Рисунок А3. Блок-схема Редактирование спектакля



Рисунок А4. Блок-схема Заполнение полей при клике по строке таблицы

Изображение выглядит как снимок экрана, текст, дисплей, программное обеспечение

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок А5. Макет приложения