**Федеральное государственное бюджетное образовательное учреждение   
высшего образования**

**«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»**

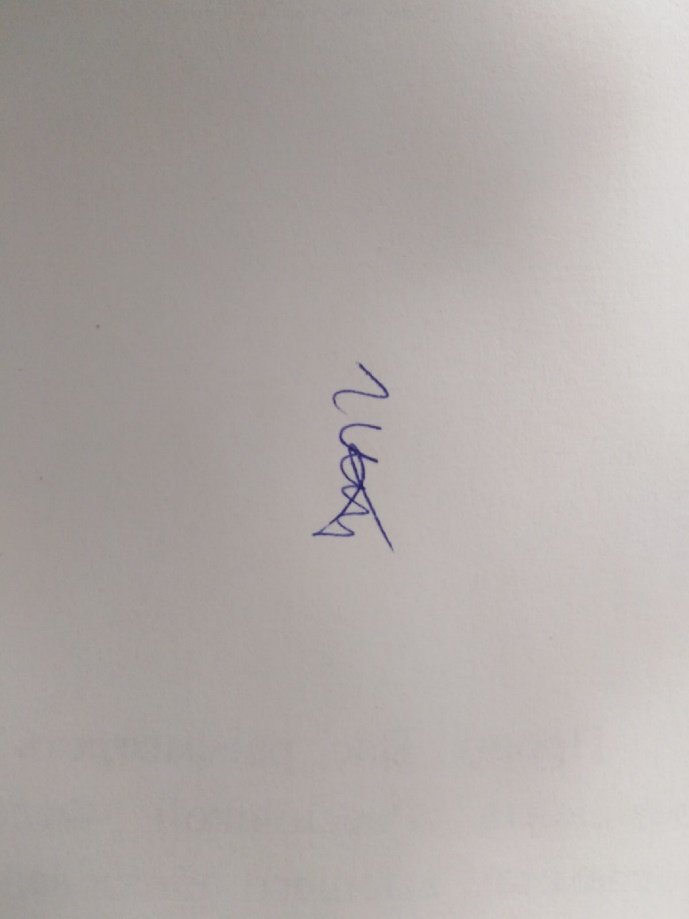
**Институт приборостроения, автоматизации и информационных технологий**

**Кафедра программной инженерии**

**ОТЧЕТ**

по производственной практике

на материалах АНО «Центр Интернет-образования»



Студент Шорин В.Д.

Группа 71ПГ

Институт приборостроения, автоматизации

и информационных технологий

Направление 09.03.04 «Программная инженерия»

Руководитель практики

от университета \_\_\_\_\_\_\_\_\_\_\_\_Фролов А.И.

Руководитель практики

от профильной организации \_\_\_\_\_\_\_\_\_\_\_\_Забелин С.А.

М.П.

Оценка защиты \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Орел, 2020

**Федеральное государственное бюджетное образовательное учреждение   
высшего образования**

**«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»**

**Институт приборостроения, автоматизации и информационных технологий**

**Кафедра программной инженерии**

Направление подготовки 09.03.04 «Программная инженерия»

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**на производственную практику**

для обучающегося Шорина Владислава Дмитриевича

3 курса очной формы обучения, группы 71ПГ

Место прохождения практики: АНО «Центр Интернет-образования»

Срок прохождения практики с «7» июля 2020 г. по «20» июля 2020 г.

**Содержание практики (вопросы, подлежащие изучению):**

* Определение требований к функциональной возможности управления проектом;
* Проектирование и реализация базы данных;
* Проектирование функциональной возможности управления проектом;
* Реализация функциональной возможности управления проектом;

**Планируемые результаты практики:**

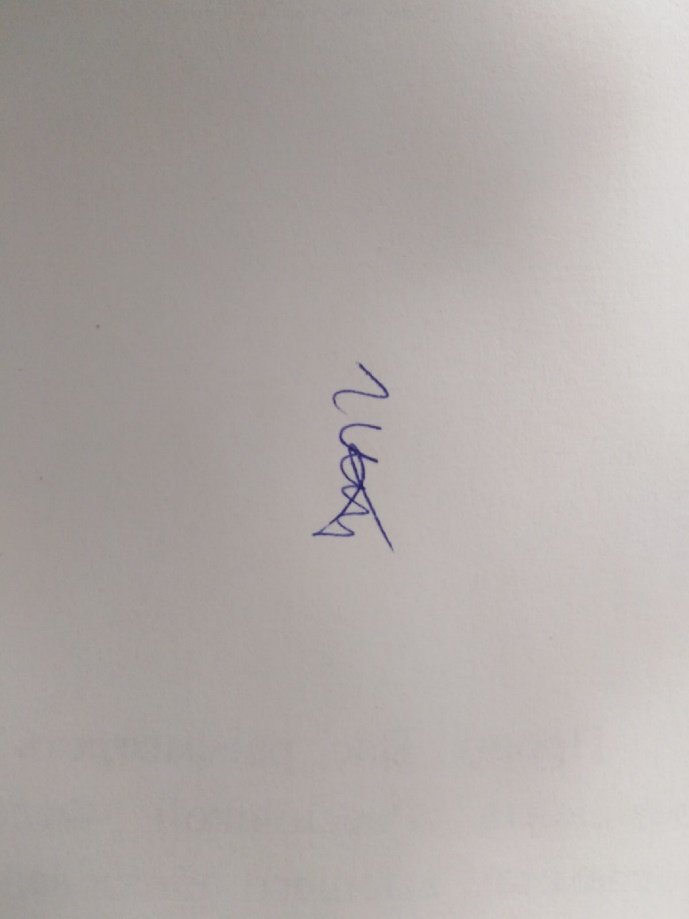
* Требования к функциональной возможности управления проектом;
* База данных;
* Реализованные функции управления проектом;
* Исходный текст функции управления проектом.

Руководитель практики

от университета \_\_\_\_\_\_\_\_\_\_\_\_Фролов А.И.

Согласовано:

Руководитель практики

от профильной организации \_\_\_\_\_\_\_\_\_\_\_\_Забелин С.А.

Задание принял: «7» июля 2020 г.

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 4](#_Toc45783852)

[1 ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ 5](#_Toc45783853)

[1.1 Проектирование базы данных 5](#_Toc45783854)

[1.2 Реализация базы данных 6](#_Toc45783855)

[2 ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ВОЗМОЖНОСТИ УПРАВЛЕНИЯ ПРОЕКТОМ 7](#_Toc45783856)

[2.1 Описание функциональной возможности управления проектом 7](#_Toc45783857)

[2.2 Диаграммы классов 8](#_Toc45783858)

[2.3 Архитектура приложения 8](#_Toc45783859)

[3 РЕАЛИЗАЦИЯ ФУНКЦИОНАЛЬНОЙ ВОЗМОЖНОСТИ УПРАВЛЕНИЯ ПРОЕКТОМ 10](#_Toc45783860)

[3.1 Особенности реализации класса… 10](#_Toc45783861)

[3.2 Примеры функционирования программного обеспечения 10](#_Toc45783862)

[ЗАКЛЮЧЕНИЕ 11](#_Toc45783863)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc45783864)

[ПРИЛОЖЕНИЕ А 13](#_Toc45783865)

# ВВЕДЕНИЕ

?

**Целью** практики является проектирование и реализация функциональных возможностей создания проекта и ведения его общей информации.

**Задачами**, которые необходимо выполнить для достижения поставленной цели, являются:

1. проектирование и реализация базы данных;
2. проектирование систем создания проекта и ведения общей информации проекта;
3. построение диаграмм классов;
4. создание исходного кода спроектированных систем.

# ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ БАЗЫ ДАННЫХ

# Проектирование базы данных

Предметной областью для данной базы данных является проект и вся информация, касающаяся него. У проекта всегда есть название, какая-то информация о нем и т.д. Поэтому выделим следующий список атрибутов, относящихся к проекту:

* Название проекта.
* Краткое название проекта.
* Краткое описание проекта.
* Сроки проекта.
* Ресурс.
* Статус проекта.

Понятие «статус проекта» подразумевает под собой состояние проекта в текущий момент – активирован он или нет.

«Ресурс» подразумевает под собой, например, ссылку на репозиторий с проектом.

У каждого члена команды проекта помимо основной информации о нем также есть и информация о том, какую должность он занимает в текущем проекте и состояние относительно этого проекта.

У любого проекта также всегда есть список заинтересованных лиц, которые принимают непосредственное или косвенное участие в разработке проекта. Потому важно хранить наиболее полную информацию о них для более качественной разработки проекта. Необходимо иметь данные о том, какую роль занимает заинтересованное лицо в проекте, как с ним можно взаимодействовать, для чего он хочет взаимодействовать с командой проекта и многое другое.

Исходя из данной информации, получим следующую логическую схему базы данных (рисунок 1.1).

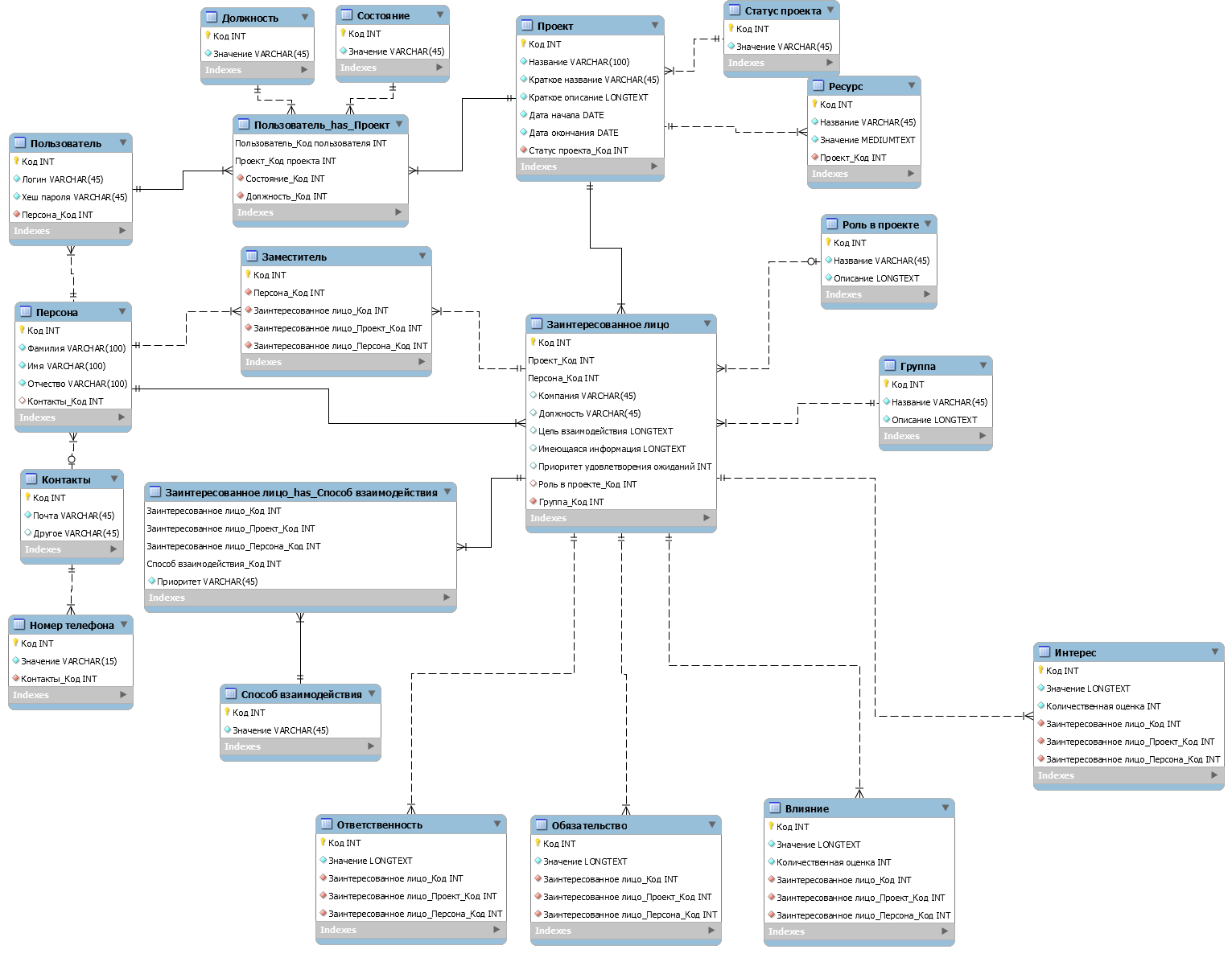


Рисунок 1.1 – Логическая схема базы данных

Также, необходимо будет разработать триггеры к базе данных. Например, очень важным будет наличие триггера, который будет удалять из всех таблиц, связанных с проектом, данные, относящиеся к удаляемому проекту. Аналогичные триггеры необходимо разработать для операций удаления заинтересованного лица, пользователя и т.д.

# Реализация базы данных

Для реализации спроектированной базы данных будет использоваться MySQL в качестве СУБД.

# ПРОЕКТИРОВАНИЕ ФУНКЦИОНАЛЬНОЙ ВОЗМОЖНОСТИ УПРАВЛЕНИЯ ПРОЕКТОМ

# Описание функциональной возможности управления проектом

Функциональная возможность управления проектом состоит в том, чтобы руководитель проекта, вошедший в систему, мог создать новый проект и удалить или отредактировать информацию по любому из проектов из списка.

Выделим следующие структурные части для функциональной возможности управления проектом: модель и контроллер. Структурная часть модель обеспечивает хранение информации и получение информации из базы данных. Структурная часть контроллер обеспечивает обработку запросов пользователя, реализацию бизнес-логики приложения, обращение к модели за необходимыми данными и ответ интерфейсной части приложения.

Все модели будут соответствовать сущностям разработанной ранее базы данных для более удобной работы с ними.

Все модели должны выполнять одинаковые по своей сути функции по обращению к базе данных, отличающиеся реализацией. Потому, целесообразно будет эти функции вынести в один общий интерфейс, который будет наследоваться моделями.

Выделим контроллеры, необходимые для задачи создания проекта:

* Контроллер управления проектом.
* Контроллер управления пользователем.
* Контролер управления заинтересованным лицом.

Контроллер управления проектом отвечает за все действия, связанные с проектом (создание, удаление, просмотр списка проектов или одного конкретного и т.д.) Для каждого действия должен быть разработан свой метод, который будет обращаться к необходимой модели (моделям), выполнять действия с полученными данными и возвращать результат.

По аналогичной схеме работают контроллеры управления пользователем и заинтересованным лицом.

# Диаграммы классов

Ниже представлены диаграммы классов для вышеперечисленных структурных частей (рисунки 2.1 – …).

# Архитектура приложения

Для проектирования архитектуры веб-приложения будет использоваться архитектурный стиль REST (от англ. **R**epresentational **S**tate **T**ransfer — «передача состояния представления»). REST является согласованным набором ограничений, которые должны учитываться при проектировании распределенных веб-приложений. Использование стиля REST в большинстве случаев ведет к упрощению архитектуры разрабатываемого приложения (что способствует упрощенному расширению системы, изменению каких-либо ее компонентов и т.д.), а также к повышению его производительности. Веб-приложения, разработанные по принципам REST, называют также «RESTful» приложениями.

Также, с помощью REST архитектуры можно наилучшим образом использовать протокол HTTP. Помимо понимая данного протокола, необходимо также иметь представления о следующих абстракциях:

* **Ресурс** – ключевая абстракция. Представляет собой любую информацию, которую мы хотим представить через наше приложение (например, какой-то конкретный пользователь, список постов на сайте и т.д.).
* **URI** (от англ. **U**niform **R**esource **I**dentifier – «унифицированный (единообразный) идентификатор ресурса») ресурса – способ идентификации ресурса для предоставления клиенту. Например, для того чтобы создать пользователя, можно использовать следующий URI: POST/users (где POST – метод, отправляемый клиентом серверу, по которому сервер понимает, что ему необходимо сделать с данными в пришедшем запросе. В данном случае означает создание ресурса). Для получения списка пользователей: GET/users (где GET означает получить) и т.д.

Для обеспечения общения клиента и сервера чаще всего в REST применяется формат JSON (как наиболее простой и удобный). Но не исключается использование и других форматов (как XML и др.).

В HTTP определяется следующая структура запроса (**request**):

* строка запроса (**request line**) – с ее помощью определяется тип запроса;
* заголовки запроса (**header fields**) – с их помощью характеризуются различные сведения (тело сообщения, параметры передачи и т.д.);
* тело сообщения (**body**) – является не обязательным пунктом.

В HTTP определяется следующая структура ответного сообщения (**response**):

* строка состояния (**status line**) – включает в себя код состояния и сообщение о причине этого состояния;
* заголовки ответа (**header fields**) – с их помощью характеризуются различные сведения (тело сообщения, параметры передачи и т.д.);
* дополнительное тело сообщения (**body**) – является не обязательным пунктом.

Для того, чтобы сервер понял, какое действие ему необходимо выполнить с пришедшим запросом, в заголовке указывается соответствующий метод:

* **GET** – получение информации о ресурсе.
* **POST** – создание нового ресурса.
* **PUT** – обновление существующего ресурса.
* **DELETE** – удаление ресурса и другие.

Также, при разработке REST архитектуры выделяют два подхода определения способов взаимодействия клиента и сервера:

* Contract First.
* Code First.

Первый подход подразумевает первым делом разработку контракта по которому в дальнейшем будет разрабатываться сервер. Это вносит следующие преимущества:

1. **Параллельная разработка.** Поскольку контракт уже сформирован, каждая команда, которая занимается разработкой сервера или интерфейса, может спокойно выполнять свои обязанности и не будет необходимости кого-то ждать.
2. **Известный ответ.** Исходя из того, что кодирование происходит на основе уже составленного контракта, команды знают, чего они могут ждать от другой стороны в качестве ответа/запроса. Это позволяет использовать заглушки при тестировании приложений в случаях, когда межгрупповое тестирование является невозможным.
3. **Кроссплатформенная совместимость.** Выбор технологии реализации сервиса не имеет большого значения, поскольку его параметры зависят только от составленного контракта. Как клиент, так и сервер могут использовать разные технологии.

У данного способа имеются также и недостатки. Первым является повышение начальных затрат (особенно временных) поскольку необходимо быть точно уверенным, что составленный контракт является четко определенным и не потребуется его частое редактирование. Вторым недостатком является сложный механизм обновления контракта и обмена. Поскольку контракт влияет на все заинтересованные стороны, то должен существовать надежный механизм, который сможет передать изменения всем потребителям.

Второй подход подразумевает собой генерацию контракта из кода. Он имеет следующие преимущества и недостатки:

Преимущества:

1. **Минимальные усилия при создании контракта.** Благодаря различным сервисам, контракт можно сгенерировать автоматически на основе кода. Отсюда следует и второе преимущество:
2. **Синхронизация кода и контракта.** Контракт и код всегда синхронизированы, поскольку первый генерируется из второго.

Недостатки:

1. **Отсутствие параллельной разработки.** Клиент не может разрабатываться до тех пор, пока не будет разработан сервер и составлен на его основе контракт. Это сильно замедляет совместную разработку приложения.
2. **Неизвестный ответ.** Поскольку не будет известно, что возвращает сервер, до тех пор, пока не разработают контракт, направление разработки клиента может откланяться от общей цели. Это может привести к лишним изменениям и последующим исправлениям либо сервера, либо клиента. Следовательно, произойдет напрасная трата усилий.

Проанализировав вышеизложенную информацию, приходим к выводу, что подход Code First является наиболее предпочтительным и дальнейшая разработка будет с его применением.

В качестве формата обмена сообщениями между клиентом и сервером будем использовать формат JSON.

Далее будет разработка контракта, в котором будут описаны будущие роуты (ссылки на сайте), методы, входные и выходные данные, ответы и т.д.

# РЕАЛИЗАЦИЯ ФУНКЦИОНАЛЬНОЙ ВОЗМОЖНОСТИ УПРАВЛЕНИЯ ПРОЕКТОМ

# Особенности реализации класса…

# Примеры функционирования программного обеспечения

# ЗАКЛЮЧЕНИЕ

…

Задачи реализованы, цель выполнена, получили новые знания о…

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

mysql

rest api

php laravel

# ПРИЛОЖЕНИЕ А

**СОВМЕСТНЫЙ РАБОЧИЙ ГРАФИК (ПЛАН)**

**проведения производственной практики**

Обучающегося 3 курса очной формы обучения, группы 71ПГ

Направление подготовки 09.03.04 «Программная инженерия»

Место прохождения практики АНО «Центр Интернет-образования»

Срок прохождения практики с «07» июля 2020 г. по «20» июля 2020 г.

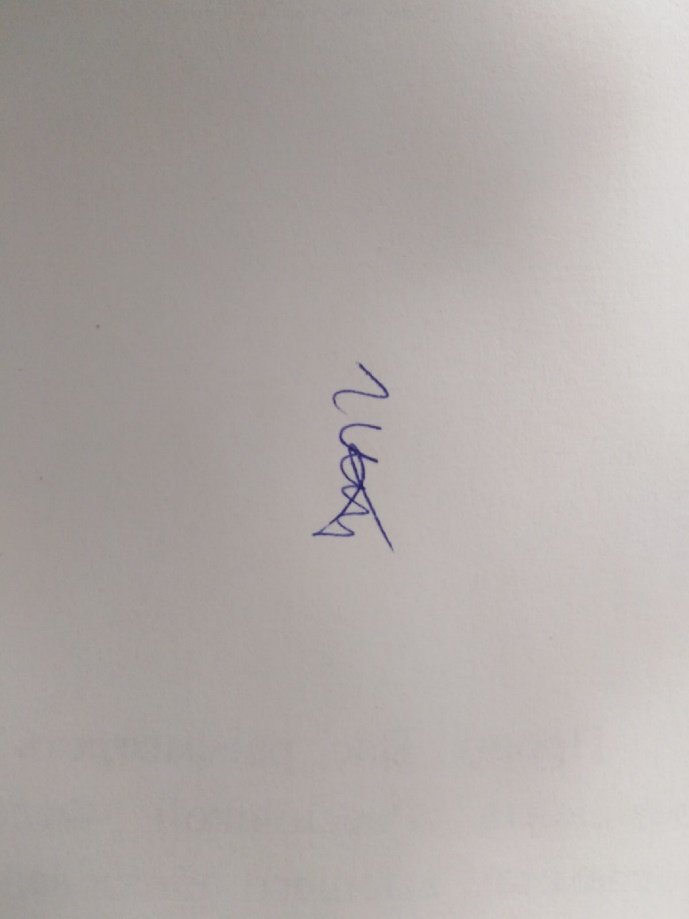
| № п/п | Наименование этапа проведения практики\* | Вид работ | Срок прохождения этапа практики | Форма отчетности | Отметка о выполнении |
| --- | --- | --- | --- | --- | --- |
| 1 | Подготовительный (организационный) этап | 1.Организационное собрание для разъяснения целей, задач, содержания и порядка прохождения практики.  2. Инструктаж по технике безопасности.  3. Выдача индивидуального задания. | 07.07.2020 | Индивидуальное задание |  |
| 2 | Основной этап | 1. Ознакомление с конкретными видами деятельности в соответствии с положениями структурных подразделений и должностными инструкциями.  2. Сбор информации и материалов практики.  3. Выполнение программы практики, индивидуального задания на практику.  4.Обработка, систематизация и анализ фактического и теоретического материала. | 28.07.2020  -  18.07.2020 | Фактический и теоретический материал, результаты анализа |  |
| 3 | Заключительный этап | Составление отчета по практике.  Защита отчета по практике с представлением материалов конкретной профильной организации. | 20.07.2020 | Отчет по практике |  |

Руководитель практики

от университета \_\_\_\_\_\_\_\_\_\_\_\_Фролов А.И.

Руководитель практики

от профильной организации \_\_\_\_\_\_\_\_\_\_\_\_Забелин С.А.



С рабочим графиком (планом) ознакомлен:

Обучающийся Шорин В.Д.

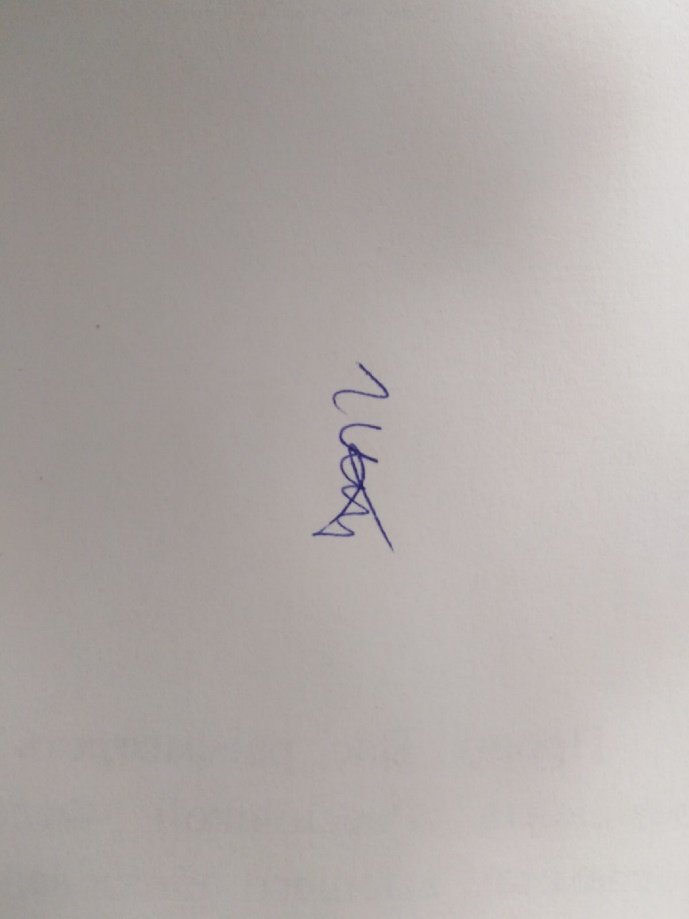
**Федеральное государственное бюджетное образовательное учреждение   
высшего образования**

**«ОРЛОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
ИМЕНИ И.С. ТУРГЕНЕВА»**

**Институт приборостроения, автоматизации и информационных технологий**

**ДНЕВНИК**

производственной практики студента



Шорин Владислав Дмитриевич

Курс 3

Группа 71ПГ

Место прохождения практики АНО «Центр Интернет-образования»

Руководитель практики

от университета \_\_\_\_\_\_\_\_\_\_\_\_Фролов А.И.

Руководитель практики

от профильной организации \_\_\_\_\_\_\_\_\_\_\_\_Забелин С.А.

М.П.

Начало практики

«7» июля 2020 года

Окончание практики

«20» июля 2020 годаГрафик

прохождения практики

|  |  |  |  |
| --- | --- | --- | --- |
| № п/п | Дата | Работа, выполненная студентом | Примечание |
| 1 | 07.07.2020 |  |  |
| 2 | 08.07.2020 |  |  |
| 3 | 09.07.2020 |  |  |
| 4 | 10.07.2020 |  |  |
| 5 | 11.07.2020 |  |  |
| 6 | 13.07.2020 |  |  |
| 7 | 14.07.2020 |  |  |
| 8 | 15.07.2020 |  |  |
| 9 | 16.07.2020 |  |  |
| 10 | 17.07.2020 |  |  |
| 11 | 18.07.2020 |  |  |
| 12 | 20.07.2020 |  |  |

Руководитель практики

от университета \_\_\_\_\_\_\_\_\_\_\_\_Фролов А.И.

Руководитель практики

от профильной организации \_\_\_\_\_\_\_\_\_\_\_\_Забелин С.А.

**Отзыв руководителя практики от профильной организации**

Студент Шорин Владислав Дмитриевич

Направление 09.03.04 «Программная инженерия»

Место прохождения практики: АНО «Центр Интернет-образования»

Сроки проведения практики: с «7» июля 2020 г. по «20» июля 2020 г.

Отчет по практике отвечает требованиям выданного индивидуального задания. Все поставленные вопросы разрешены и освещены в достаточной степени.

В ходе прохождения студентом практики наблюдалось ответственное отношение к выполняемой работе, оперативное выполнение поручений, качественный уровень и достаточная степень подготовленности студента к самостоятельному выполнению отдельных заданий.

Отчет по практике изложен грамотно, в достаточном объеме, с необходимой для выполнения индивидуального задания степенью использования отечественной и иностранной литературы, а также материалов и справочников.

Дисциплина студента во время практики оценивается удовлетворительно. Случаев нарушения трудовой дисциплины не выявлено. Отрицательные черты, действия, проявления, характеризующие студентку с негативной стороны в период прохождения практики не обнаружены.

Оценка выполнения программы практики и индивидуального задания в целом: зачтено.

**Руководитель** практики от АНО «Центр Интернет-образования»

инженер-программист, Забелин С.А. «20» июля 2020 г.

М.П.