**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

**Факультет Искусственного интеллекта**

(наименование факультета/ института)

**Направление подготовки /специальность:** 09.03.03 Прикладная информатика .

(код и наименование направления подготовки /специальности)

**Профиль/специализация:** Искусственный интеллект и большие данные .

(наименование профиля/специализации)

**Форма обучения:** Заочная .

(очная, очно-заочная, заочная)

|  |
| --- |
| **УТВЕРЖДАЮ** |
| Декан факультета |
| Э.Р.Жданов Подпись (Подпись) (ФИО) |
| «\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_ г. |

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**НА УЧЕБНУЮ ПРАКТИКУ**

(вид практики)

**ТЕХНОЛОГИЧЕСКАЯ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ) ПРАКТИКА**

(тип практики)

**\_2\_\_ семестр**

обучающегося группы ОБПз-42404МОиибд \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Шифр и № группы) (ФИО обучающегося)

Место прохождения практики:

|  |
| --- |
| Университет «Синергия» |

(наименование Профильной организации)

Срок прохождения практики: с «01» октября 2024 г. по «16» марта 2025 г.

**Содержание индивидуального задания на практику:**

| **№ п/п** | **Виды работ** |
| --- | --- |
| 1. | Инструктаж по соблюдению правил противопожарной безопасности, правил охраны труда, техники безопасности, санитарно-эпидемиологических правил и гигиенических нормативов. |
| 2. | Выполнение определенных практических кейсов-задач, необходимых для оценки знаний, умений, навыков и (или) опыта деятельности по итогам учебной (технологическая (проектно-технологическая) практика) практики.  (вид практики, тип практики) |
| 2.1. | Кейс-задача № 1  Дан одномерный массив А размерности N. Найти сумму отрицательных элементов, расположенных между максимальным и минимальным.  Ответом на задачу будет ссылка на репозиторий GitHub, где хранится Ваша программа. Или иным удобным для Вас способом. |
| 2.2. | Кейс-задача № 2  Написать тестовую программу, которая демонстрирует работу методов базового и производного классов.  Ответом на задачу будет ссылка на репозиторий GitHub, где хранится Ваша программа. Или иным удобным для Вас способом. |
| 2.3. | * + Кейс-задача № 3   + Спроектируйте базу данных «Туризм» (перечень предоставляемых услуг, заказ туров и др.). При проектировании базы данных необходимо создать 4-5 таблиц предметной области: 3-4 таблицы-справочника и 1 таблицу переменной информации. Для всех таблиц создать первичные ключи. Построить связи между таблицами при помощи внешних ключей: атрибуты таблицы переменной информации должны ссылаться на ключевые атрибуты таблиц справочников.   + Когда вы создаете базу данных в MySQL с помощью MySQL Workbench (или любого другого инструмента), вы можете экспортировать схему базы данных в виде скрипта SQL. Этот скрипт SQL содержит определения таблиц, связей, индексов и других структур базы данных, которые вы создали. Или иным удобным для Вас способом. |
| 2.4. | Кейс-задача № 4  Проведите анализ и опишите имеющихся на рынке программного обеспечения информационных систем, построенных по архитектуре WEB-приложений. Оцените и опишите возможности предлагаемых систем по архитектуре WEB-приложений и варианты их использования в компании. Создайте с помощью Delphi 10.2 и MS Internet Information Server (IIS) приложение WEB-архитектуры на любую тему. Базу данных для WEB-приложения создать в MS SQL Server.   * Ответом на задачу будет ссылка на репозиторий GitHub, где хранится Ваша программа. Или иным удобным для Вас способом. * Когда вы создаете базу данных в MySQL с помощью MySQL Workbench (или любого другого инструмента), вы можете экспортировать схему базы данных в виде скрипта SQL. Этот скрипт SQL содержит определения таблиц, связей, индексов и других структур базы данных, которые вы создали. Или иным удобным для Вас способом. |
| 2.5. | Кейс-задача № 5  Сформировать и описать аналитический обзор проделанной работы в одной из кейс-задач № 1-4 на Ваш выбор, по следующим критериям:   * Функциональность: Это основной критерий, который оценивает, насколько хорошо программа выполняет свои задачи. Программа должна корректно обрабатывать все входные данные и выдавать правильные результаты. * Производительность: Этот критерий оценивает скорость работы программы. Быстрая работа программы важна для приложений, где требуется обработка большого объема данных или высокая интерактивность. * Удобство использования (юзабилити): Критерий, который оценивает, насколько легко пользователь может взаимодействовать с программой. Удобный интерфейс и понятные сообщения об ошибках делают программу более привлекательной для пользователей. * Безопасность: Важный критерий, особенно для программ, которые работают с конфиденциальными данными. Безопасная программа должна защищать данные от несанкционированного доступа и предотвращать возможные уязвимости. * Масштабируемость: Критерий, который оценивает, насколько легко программу можно расширить или модифицировать для обработки большего объема данных или добавления новых функций. * Сопровождаемость: Оценивает, насколько легко поддерживать и обновлять программу. Хорошо структурированный код, четкие комментарии и документация упрощают процесс сопровождения. * Переносимость: Критерий, который оценивает, насколько легко программу можно перенести на другую платформу или систему без значительных изменений. * Качество кода: Оценка качества кода включает такие аспекты, как читаемость, структурированность, использование стандартных практик и паттернов проектирования, отсутствие ошибок и т.д. * Тестирование: Критерий, который оценивает, насколько хорошо программа была протестирована. Полное тестирование помогает выявить ошибки и улучшить качество программы.   Это лишь примерный список параметров, которые могут быть использованы для описания задания. Конкретные параметры могут варьироваться в зависимости от программы.  На основе проведенного анализа выше предложите способы решения выявленных проблем в ходе выполнения выбранной Вами кейс-задачи. |
| 3. | Систематизация собранного нормативного и фактического материала. |
| 4. | Оформление отчета о прохождении практики. |
| 5. | Защита отчета по практике. |

Обучающийся индивидуальное задание получил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_

(ФИО) (Подпись)

«26» января 2025г.



**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ «СИНЕРГИЯ»**

**Факультет Искусственного интеллекта**

(наименование факультета/ института)

**Направление подготовки /специальность:** 09.03.03 Прикладная информатика .

(код и наименование направления подготовки /специальности)

**Профиль/специализация:** Искусственный интеллект и большие данные .

(наименование профиля/специализации)

**Форма обучения:** Заочная .

(очная, очно-заочная, заочная)

**ОТЧЕТ**

**ПО УЧЕБНОЙ ПРАКТИКЕ**

(вид практики)

**ТЕХНОЛОГИЧЕСКАЯ (ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ) ПРАКТИКА**

(тип практики)

**2 семестр**

|  |  |  |  |
| --- | --- | --- | --- |
| **Обучающийся** |  |  |  |
|  | (ФИО) |  | (подпись) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Ответственное лицо**  **от Профильной организации** |  |  |  |
| М.П. (при наличии) | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Практические кейсы-задачи, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности по итогам практики**

| **№ п/п** | **Подробные ответы обучающегося на практические кейсы-задачи** |
| --- | --- |
| Кейс-задача № 1 | Репозиторий с задачей и решением: https://github.com/rinerte/1YearPractice\_Synergy/tree/main/1Task\_Go  1) В файле README.md содержится описание задачи, инструкция по запуску программы и тестов к ней, а также скриншот с успешным запуском тестов.  2) В файле main.go содержится программа  3) В файле main\_test.go содержатся тесты  Описание программы:  В файле main.go написана функция computeSumBetweenMaxAndMin(args []string) (int, error), которая принимает массив элементов.  Эта функция считает сумму отрицательных элементов между максимальным и минимальным элементом.  Алгоритм нахождения следующий:  1) Создаем переменную для хранения максимального значения из массива и присваиваем ей минимальное возможное значение*.*  2) Создаем переменную для хранения минимального значения из массива и присваиваем ей максимальное возможное значение*.*  *3)* Создаем переменные для хранения индексов этих значений в массиве.  4) Проходим по каждому элементу массива и сверяем его с максимумом и минимумом. В случае если элемент меньше минимума — обновляем минимум и его индекс. В случае если элемент больше максимума — обновляем максимум и его индекс.  5) Находим меньший из индексов  6) Проходим по массиву от меньшего индекса до большего и суммируем отрицательные элементы.  В функции main используем функцию computeSumBetweenMaxAndMin  передавая ей аргументы командной строки. |
| Кейс-задача № 2 | Репозиторий с задачей и решением: https://github.com/rinerte/1YearPractice\_Synergy/tree/main/2Task\_CSharp\_Classes  1) В файле README.md содержится описание задачи, описание решения, а также скриншот с успешным запуском программы.  Описание решения: Программа содержит базовый класс Geometry, его метод Show наследует дочерний абстрактный класс Figure. Классы Circle, Square, Triangle - реализуют интерфейс, определенный родительским классом и наследуют метод Show из класса Geometry.  Класс Geometry содержит один виртуальный метод Show(), который выводит сообщение "This is method from base class".  Класс Figure наследуется от Geometry. Этот класс определяет абстрактное поле Name, а также два абстрактных метода GetArea() и GetPerimeter(). Так как класс абстрактный — нельзя создать его экземпляр и по сути он служит как интерфейс для классов, его наследующих. А именно: Square, Circle, Triangle.  Классы Square, Circle, Triangle наследуют Figure. Переопределяют метод Show() из родительского класса Geometry, теперь этот метод показывает сообщение "This method from child class".  В каждом из этих классов методы GetArea и GetPerimeter вычисляют площадь и периметр своим способом, соответсвующим фигуре. Алгоритм вычисления описывать здесь излишне, так как это стандартные геометрические формулы.  В каждом классе описан конструктор, который создает экземпляр фигуры. Представляет интерес здесь лишь конструктор треугольника. В нем проводится проверка на возможность существования заданного треугольника. Если стороны были указаны некорректно программа выдаст исключение.  В файле Program.cs проводится демонстрация возможного использования полученной системы классов: создается два треугольника, окружность и квадрат. Все фигуры помещаются в список, который типизирован классом Figure. Т.е. родительским для фигур.  Далее выполняется проход в цикле по этому списку и вызываются методы GetArea() и GetPerimeter().  В конце создается экземпляр базового для всех фигур класса Geometry и экземпляр класса Square. Вызываются методы Show() базового и производного класса. |
| Кейс-задача № 3 | Репозиторий с задачей и решением: <https://github.com/rinerte/1YearPractice_Synergy/tree/main/3Task_DB_Schema>  1) В файле README.md содержится описание задачи и описание решения.  2) В файле main/main.go содержатся структуры, которые используются для работы с базой данных в программе на языке GO.  Описание структуры данных:  База данных включает 5 таблиц:  4 справочника: services, clients, hotels, countries  1 таблица переменных данных: Tours  Таблицы справочники: 1) Таблица services: - id (INT, PK) - уникальный идентификатор услуги.  -name (text) - название услуги  - price (bigint) – цена услуги  2) Таблица clients: - id (INT, PK) - уникальный идентификатор клиента.  -name (text) - ФИО клиента  - phone (text) – телефон клиента  3) Таблица hotels: - id (INT, PK) - уникальный идентификатор  -name (text) - название отеля  - country\_id (bigint, FK) – ссылка на страну  4) Таблица countries: - id (INT, PK) - уникальный идентификатор  -name (text) - название страны  - visa (boolean) – флаг необходимости визы для посещения данной страны  Таблица переменных данных Tours:  Содержит информацию о сформированных турах:  - id (bigint) - идентификатор - client\_id (bigint, FK) — ссылка на клиента  - country\_id (bigint,FK) – ссылка на страну  - departure\_time (timestamp with time zone) — дата отбытия  - arrival\_time (timestamp with time zone) — дата прибытия  Описание программы на Go:  Программа использует ORM-библиотеку GORM для работы с PostgreSQL.  Структуры данных в программе соответствуют таблицам БД. Также содержат встроенную структуру gorm.Model (добавляет поля ID, CreatedAt, UpdatedAt, DeletedAt).  Алгоритм работы: 1) Проверяется существование базы данных tours. Если она отсутствует, создается.  2) Обрабатывается ошибка дублирования БД  3) Используется PostgreSQL-драйвер для подключения к БД  4) GORM автоматически создает таблицы и связи на основе структур.  5) Для связей многие-ко-многим (Hotels и Services в Tour) создаются промежуточные таблицы tour\_hotels и tour\_services.  Заключение  Структура БД соответствует требованиям задачи: 4 справочника countries, hotels, services, clients и 1 таблица переменных данных tours.  Реализованы связи:  Один-ко-многим (отель → страна, тур → клиент, тур → страна).  Многие-ко-многим (тур ↔ отели, тур ↔ услуги).  Данные нормализованы, исключены дублирования.  Использование GORM:  Упрощает работу с БД: автомиграция, управление связями, CRUD-операции.  Обработка ошибок:  Корректно обрабатывается создание дубликата БД.  Использование gorm.Model добавляет стандартные поля для аудита.  Программа готова к интеграции с API для управления турами, клиентами и услугами. Для расширения функционала можно добавить методы для работы с сущностями (создание, удаление, фильтрация). |
| Кейс-задача № 4 | 1) Ответ на первую часть задания:  Анализ веб-приложений и их использование в компаниях:  **1. Архитектура веб-приложений: основные типы**  Веб-приложения классифицируются по архитектурным подходам и целям использования:  a) SPA (Single-Page Application)  - Особенности: Загружается однократно, обновляет контент динамически через API (например, React, Angular).  - Использование:  - Корпоративные порталы (внутренние системы компаний).  - Сервисы с высоким уровнем интерактивности (онлайн-банкинг, аналитические платформы).  b) MPA (Multi-Page Application)  - Особенности Каждое действие перезагружает страницу (традиционные CMS, интернет-магазины).  - Примеры  - WordPress (сайты-визитки, блоги).  - Magento (электронная коммерция).  c) SSR (Server-Side Rendering  - Особенности: Рендеринг на сервере для SEO и скорости (Next.js, ASP.NET MVC).  - Применение:  - Контентные платформы (новостные сайты).  - Маркетплейсы.  d) PWA (Progressive Web App)  -Особенности: Могут работать оффлайн, устанавливаются на устройство.  - Сценарии:  - Мобильные приложения для ритейла.  - Сервисы с ограниченным интернет-соединением.  **2. Популярные фреймворки и их возможности**  a) ASP.NET Core (Microsoft)  - Преимущества:  - Высокая производительность.  - Интеграция с Azure, MS SQL.  - Использование:  - Корпоративные системы (ERP, CRM).  - Сложные веб-сервисы (госучреждения, банки).  b) Spring Boot (Java)  - Особенности:  - Масштабируемость, безопасность.  - Поддержка микросервисов.  - Примеры:  - Финансовые платформы (транзакционные системы).  - Логистические сервисы (управление цепочками поставок).  c) Django (Python)  - Сильные стороны:  - Быстрая разработка.  - Встроенная админ-панель.  - Применение:  - Стартапы (MVP).  - Data-driven приложения (аналитика).  d) React/Angular (JS. Frontend)  - React: Гибкость, экосистема (SPA для соцсетей, дашборды).  - Angular: Полнофункциональность (корпоративные порталы).  **3. Готовые бизнес-решения**  a) CRM-системы  - Примеры: Salesforce  - Возможности:  - Управление клиентской базой.  - Автоматизация продаж.  - Для кого: Средний и крупный бизнес.  b) ERP-системы  - Платформы: Odoo, SAP.  - Функционал:  - Учет ресурсов, логистика, финансы.  - Целевая аудитория: Производственные предприятия.  c) CMS  - Пример: WordPress, Drupal.  - Использование:  - Контент-маркетинг (блоги, новостные сайты).  - Интернет-магазины (WooCommerce).  **4. Варианты использования в компаниях**   |  |  |  | | --- | --- | --- | | **Тип компании** | **Пример решения** | **Технологии** | | Ритейл | Интернет-магазин | React + Node.js + MongoDB | | Стартап | MVP для сервиса бронирования | Django + PostgreSQL | | Банк | Онлайн-банкинг | ASP.NET Core + Angular + MS SQL | | Медиа | Новостной портал | Next.js (SSR) + Headless CMS | | Логистика | Трекинг грузов | Spring Boot + React + Kafka |   **Заключение**  Современные веб-приложения предлагают гибкие решения для любых бизнес-задач. Выбор архитектуры и технологий зависит от:  - Масштаба проекта (стартап vs корпорация).  - Требований к производительности и безопасности.  - Необходимости интеграции с legacy-системами.  Например:  - ASP.NET Core подходит для компаний, ориентированных на экосистему Microsoft.  - React + Django — для быстрого прототипирования и стартапов.  - Готовые CRM/ERP — для сокращения времени на разработку.  2) Репозиторий с программой: <https://github.com/rinerte/1YearPractice_Synergy/tree/main/4Task_Web-app>  Описание программы:  Веб-приложение для просмотра списка стран и отелей в выбранной стране. Тема приложения продолжает тему задания №3. Туризм.  Структура проекта:  - Бэкенд (ASP.NET Core). При работе приложения становятся доступно API, которое предоставляет доступ к данным из базы данных с помощью HTTP GET запросов.  Модели:  Country (Страна: Id, Name, Visa)  Hotel (Отель: Id, Name, CountryId)  Контроллеры:  CountriesController (GET /countries)  HotelsController (GET /hotels/{countryId})  Фронтенд (React)  SPA приложение, которое использует API для получения данных и отображения их пользователю.  Структура файлов проекта следующая:  1)Папка components: содержит два компонента  - selector – для отображения списка стран  - pageList – для отображения списка отелей в стране  2) Папка services содержит файл api.service.js в котором стандартным методом fetch выполняются запросы к API.  Главная страница отображает список стран.  При выборе страны отправляется запрос к API для получения отелей.  Интерфейс приложения:  Рисунок 1: Список отелей в Китае  Использованные технологии и инструменты:  Бэкенд:  - ASP.NET Core Web API — фреймворк для создания веб-приложений в среде .NET  - Entity Framework Core — ОРМ, используется для управления БД и удобного доступа к ней, удобен для более быстрой разработки приложений  - PostgreSQL — сервер БД  - pgAdmin — админка сервера БД  Фронтенд  - React — JS библиотека, которая удобна для разработки реактивных SPA  - Vite — сборщик  - Tailwind CSS — библиотека css классов  Инструменты  - Visual Studio Code  - npm |
| Кейс-задача № 5 | Решение:  Для анализа была выбрана программа из задания 1  **→** Функциональность  Программа корректно решает задачу поиска суммы отрицательных элементов между максимальным и минимальным элементами массива:  - Все предоставленные тесты пройдены, включая обработку отрицательных чисел и различных позиций min/max.  - Обрабатываются ошибки ввода (нечисловые значения).  **→** Производительность  - Алгоритм работает за линейное время O(n), что оптимально для данной задачи.  - Нет вложенных циклов или операций с высокой вычислительной сложностью.  - Подходит для обработки больших массивов данных.  **→** Удобство использования (юзабилити)  Сильные стороны:  - Программа принимает входные данные через аргументы командной строки, что удобно для интеграции в скрипты.  Слабые стороны:  - Нет сообщений с подсказками (например, при отсутствии аргументов).  - Ошибки выводятся через panic, что не всегда удобно для пользователя.  **→** Безопасность  - Программа не обрабатывает конфиденциальные данные, поэтому угрозы минимальны.  - Проверка ввода на корректность (нечисловые значения отлавливаются).  Риск: Переполнение типа int не обрабатывается, но это зависит от ограничений языка Go.  **→** Масштабируемость  - Код легко модифицировать для обработки других типов данных (например, float64).  Ограничение: Входные данные передаются через аргументы командной строки, что не подходит для очень больших массивов. Решение — чтение из файла или потока ввода.  **→** Сопровождаемость  - Код хорошо структурирован: основная логика вынесена в отдельную функцию computeSumBetweenMaxAndMin.  **→** Переносимость  - Программа написана на Go, который поддерживает кросс-платформенную компиляцию.  - Нет зависимостей от специфичных для ОС функций.  **→** Качество кода  Сильные стороны:  - Читаемый код с разделением на функции.  - Используются стандартные библиотеки (math, strconv).  **→** Тестирование  Написаны юнит-тесты, покрывающие основные сценарии:  - Разные позиции min/max.  - Отрицательные числа внутри и вне диапазона.  - Ошибки ввода.  **→** Рекомендации  - Внедрить чтение данных из файла для больших массивов.  - Улучшить названия переменных и добавить комментарии.  - Расширить тестовое покрытие. |

|  |  |
| --- | --- |
| Дата: 22.02.2025 | \_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) (ФИО обучающегося) |

**Аттестационный лист**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

*(Ф.И.О. обучающегося)*

обучающий(ая)ся \_\_\_\_\_ 1 \_\_\_\_\_\_\_\_ курса \_\_\_\_\_\_\_\_ заочной \_\_\_\_\_\_\_\_\_\_\_\_\_ формы обучения

*(указать курс) (очной, очно-заочной, заочной)*

группы \_ОБПз-42404МОиибд\_\_ по направлению подготовки/ специальности 09.03.03  *(шифр группы)*

Прикладная информатика \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_,

*(код, наименование направления подготовки/ специальности)*

профиль/ специализация Искусственный интеллект и базы данных **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**,

*(наименование профиля/ специализации)*

успешно прошел(ла) учебную (технологическая (проектно-технологическая) практика) практику \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(наименование вида и типа практики)*

с «01» октября 2024 года по «16» марта 2025 года

1. **Заключение-анализ результатов освоения программы практики:**

**Индивидуальное задание обучающимся (нужное отметить √):**

* выполнено;
* выполнено не в полном объеме;
* не выполнено;

**Владение материалом (нужное отметить √):**

**Обучающийся:**

* умело анализирует полученный во время практики материал;
* анализирует полученный во время практики материал;
* недостаточно четко и правильно анализирует полученный во время практики материал;
* неправильно анализирует полученный во время практики материал;

**Задачи, поставленные на период прохождения практики, обучающимся (нужное отметить √):**

* решены в полном объеме;
* решены в полном объеме, но не полностью раскрыты;
* решены частично, нет четкого обоснования и детализации;
* не решены;

**Спектр выполняемых обучающимся функций в период прохождения практики профилю соответствующей образовательной программы (нужное отметить √):**

* соответствует;
* в основном соответствует;
* частично соответствует;
* не соответствует;

**Ответы на практические кейсы-задачи, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, обучающийся (нужное отметить √):**

* дает аргументированные ответы на вопросы;
* дает ответы на вопросы по существу;
* дает ответы на вопросы не по существу;
* не может ответить на вопросы;

**Оформление обучающимся отчета по практике (нужное отметить √):**

* отчет о прохождении практики оформлен правильно;
* отчет о прохождении практики оформлен с незначительными недостатками;
* отчет о прохождении практики оформлен с недостатками;
* отчет о прохождении практики оформлен неверно;

**Аттестуемый продемонстрировал владение следующими профессиональными компетенциями:**

|  |  |  |
| --- | --- | --- |
| **Код** | **Содержание компетенции** | **Уровень освоения обучающимся**  **(нужное отметить √)\*** |
| **Универсальные компетенции** | | |
| УК-1. | Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач; | * высокий * средний * низкий |
| УК-10. | Способен формировать нетерпимое отношение к коррупционному поведению; | * высокий * средний * низкий |
| **Общепрофессиональные компетенции** | | |
| ОПК-1. | Способен применять естественнонаучные и общеинженерные знания, методы математического анализа и моделирования, теоретического и экспериментального исследования в профессиональной деятельности; | * высокий * средний * низкий |
| ОПК-2. | Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их прирешении задач профессиональной деятельности; | * высокий * средний * низкий |
| ОПК-3. | Способен решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности; | * высокий * средний * низкий |
| ОПК-4. | Способен участвовать в разработке стандартов, норм и правил, а также технической  документации, связанной с профессиональной деятельностью. | * высокий * средний * низкий |

Примечание:

* Высокий уровень – обучающийся уверенно демонстрирует готовность и способность к самостоятельной профессиональной деятельности не только в стандартных, но и во внештатных ситуациях.
* Средний уровень – обучающийся выполняет все виды профессиональной деятельности в стандартных ситуациях уверенно, добросовестно, эффективно.
* Низкий уровень – при выполнении профессиональной деятельности обучающийся нуждается во внешнем сопровождении и контроле.

1. **Показатели и критерии оценивания результатов практики**

|  |  |  |
| --- | --- | --- |
| **Оценочный критерий** | **Максимальное количество**  **баллов** | **Оценка качества выполнения каждого вида работ**  **(в баллах)** |
| Выполнение индивидуального задания в соответствии с программой практики | 30 |  |
| Оценка степени самостоятельности проведенного решения практических кейсов-задач, необходимых для оценки знаний, умений, навыков и (или) опыта деятельности по итогам практики | 30 |  |
| Оценка качества проведенного анализа собранных материалов, данных для решения практических кейсов-задач, необходимых для оценки знаний, умений, навыков и (или) опыта деятельности по итогам практики | 40 |  |
| **Итоговая оценка:** | 100 |  |

Замечания руководителя практики от Университета:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.

Руководитель практики от Университета \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись) (ФИО)