

НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ УНИВЕРСИТЕТ “СИНЕРГИЯ”»
Факультет Информационных технологий
Кафедра Цифровой экономики

ОТЧЕТ
о прохождении производственной практики
по профессиональному модулю
ПМ.02 Осуществление интеграции программных модулей
в период с «08» июня 2025 г. по «21» июня 2025 г.
Специальность 09.02.07 Информационные системы и
программирование

ФИО обучающегося: Разумнов Данила Николаевич
Группа: ДКИП-206прог
ФИО Руководителя: Сибирев Иван Валерьевич



Содержание

1. Инструктаж по соблюдению правил противопожарной безопасности, правил охраны труда, техники безопасности, санитарно-эпидемиологических правил и гигиенических нормативов
2. Ознакомление с инструментальными средствами
3. Сбор информации об объекте практики и анализ содержания источников
4. Экспериментально-практическая работа. Приобретение необходимых умений и практического опыта работы по специальности в рамках освоения вида деятельности ВД 2. Осуществление интеграции программных модулей
5. Обработка и систематизация полученного фактического материала



Организационный этап

Правила внутреннего распорядка, правила и нормы охраны труда, техники безопасности при работе с вычислительной техникой

Я, Разумнов Данила Николаевич, проходил(а) производственную практику на базе ООО СИМУЛТЕХ

При выполнении индивидуального задания по практике решал(а) кейса по интеграции сущности, репозитория, маппера и эндпоинтов для работы с оценками и реализовать фильтрацию данных по критериям.

Перед началом практики:

- Принял(а) участие в организационном собрании по практике.
- Ознакомил(а)сь с комплектом шаблонов отчетной документации по практике.
- Уточнил(а) контакты руководителя практики от Образовательной/ Профильной организации, а также правила в отношении субординации, внешнего вида, графика работы, техники безопасности:

Требования к внешнему виду: официально-деловой стиль.

График работы: 10:00-16:00

Круг обязанностей: практикант

Доступ к данным: Ограниченный



Организационный этап

Ознакомление с ПО

Ознакомиться с инструментальными средствами для выполнения производственной практики и осуществить предустановку программного обеспечения.

Программы (4)

- About Java
- Configure Java
- Java
- Visit Java.com



Программы (6)

- Git Bash
- Git CMD
- Git FAQs (Frequently Asked Questions)
- Git for Windows
- Git GUI
- Git Release Notes

```
MINGW64/c/Users/Acer
Acer@DESKTOP-8D49N84 MINGW64 ~ (Razumnov)
$ java -version
openjdk version "17.0.15" 2025-04-15
OpenJDK Runtime Environment Temurin-17.0.15+6 (build 17.0.15+6)
OpenJDK 64-Bit Server VM Temurin-17.0.15+6 (build 17.0.15+6, mixed mode, sharing)

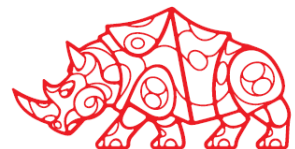
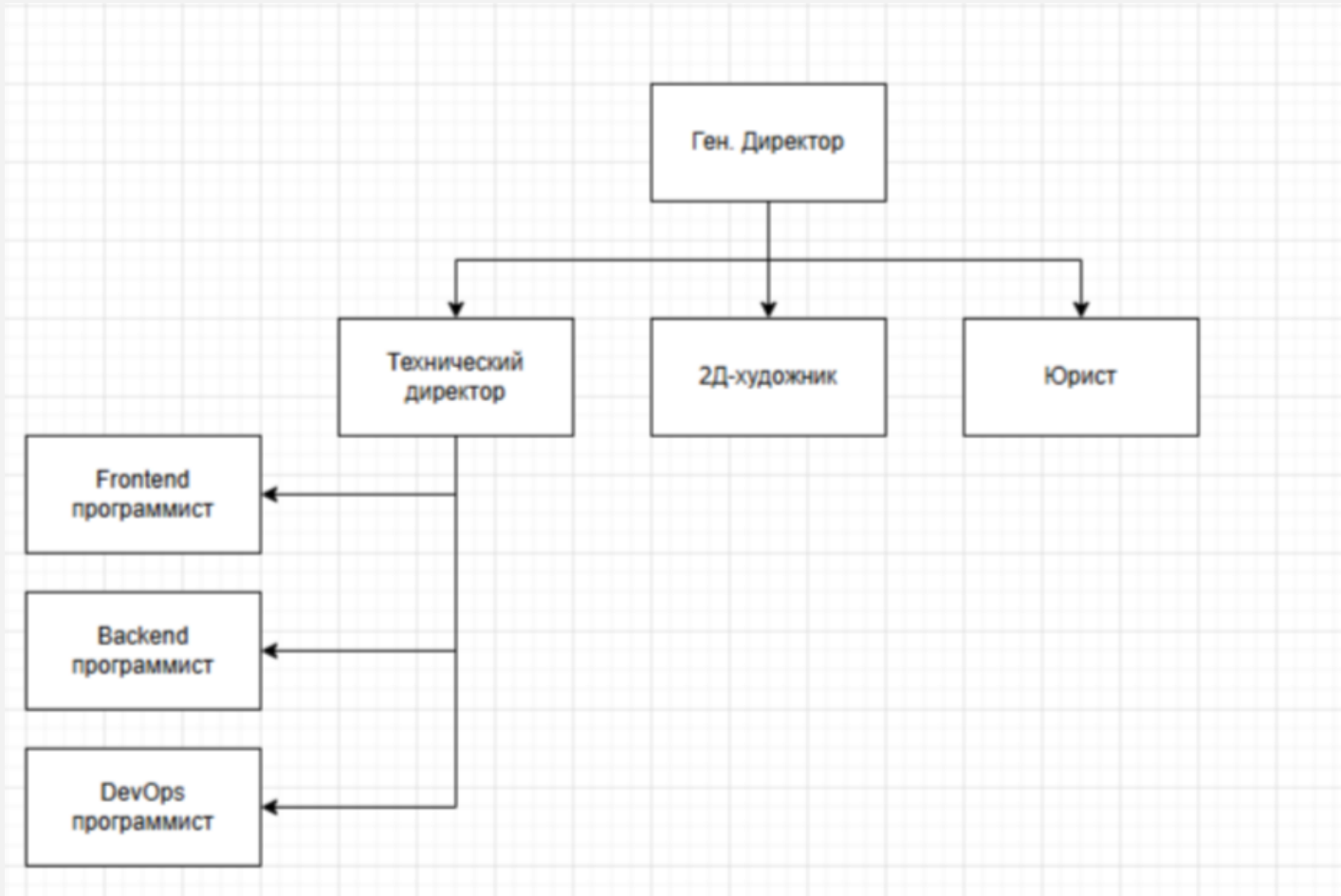
Acer@DESKTOP-8D49N84 MINGW64 ~ (Razumnov)
$ mvn -v
Apache Maven 3.9.10 (5f519b97e944483d878815739f519b2eade0a91d)
Maven home: C:\maven\apache-maven-3.9.10
Java version: 17.0.15, vendor: Eclipse Adoptium, runtime: C:\Program Files\Eclipse Adoptium\jdk-17.0.15.6-hotspot
Default locale: ru_RU, platform encoding: Cp1251
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

Acer@DESKTOP-8D49N84 MINGW64 ~ (Razumnov)
$ git --version
git version 2.49.0.windows.1
```



Организационный этап

Сбор информации об объекте практики и анализ содержания источников



Этап проектирования

Пример заполнения

Интеграция модулей в программное обеспечение

код Java

```
$ cat src/main/kotlin/com/example/gradeapi/controller/GradeController.kt
package com.example.gradeapi.controller

import com.example.gradeapi.dto.GradeDto
import com.example.gradeapi.mapper.GradeMapper
import com.example.gradeapi.repository.GradeRepository
import io.swagger.v3.oas.annotations.Operation
import io.swagger.v3.oas.annotations.media.Content
import io.swagger.v3.oas.annotations.media.Schema
import io.swagger.v3.oas.annotations.responses.ApiResponse
import io.swagger.v3.oas.annotations.tags.Tag
import jakarta.validation.Valid // <-- Добавьте этот импорт
import org.slf4j.LoggerFactory
import org.springframework.http.HttpStatus
import org.springframework.http.ResponseEntity
import org.springframework.web.bind.annotation.*

@Tag(name = "Grades", description = "API for managing student grades")
@RestController
@RequestMapping("/api/grades")
class GradeController {
    private val repository: GradeRepository

    {
        private val logger = LoggerFactory.getLogger(javaClass)

        @Operation(summary = "Get all grades")
        @ApiResponse(
            responseCode = "200",
            description = "List of all grades",
            content = [Content(mediaType = "application/json",
                schema = Schema(implementation = GradeDto::class))]
        )
        @GetMapping
        fun getAllGrades(): ResponseEntity<List<GradeDto>> {
            logger.info("Fetching all grades")
            val grades = repository.findAll().map(GradeMapper::toDto)
            return ResponseEntity.ok(grades)
        }

        @Operation(summary = "Create a new grade")
        @ApiResponse(
            responseCode = "201",
            description = "Grade created successfully",
            content = [Content(mediaType = "application/json",
                schema = Schema(implementation = GradeDto::class))]
        )
        @PostMapping
        fun createGrade(
            @Valid @RequestBody dto: GradeDto // <-- Исправленная строка
        ): ResponseEntity<GradeDto> {
            logger.info("Creating grade: $dto")
            val entity = GradeMapper.toEntity(dto)
            val saved = repository.save(entity)
            return ResponseEntity(GradeMapper.toDto(saved), HttpStatus.CREATED)
        }
    }
}
```

```
Acer@DESKTOP-8D49N84 MINGW64 ~/grade-api/grade-api (Razumnov)
$ cat src/main/kotlin/com/example/gradeapi/entity/Grade.kt
package com.example.gradeapi.entity

import jakarta.persistence.Entity
import jakarta.persistence.GeneratedValue
import jakarta.persistence.GenerationType
import jakarta.persistence.Id
import java.time.LocalDate

@Entity
data class Grade(
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    val id: Long = 0,
    val studentId: Long,
    val courseId: Long,
    val gradeValue: Int,
    val date: LocalDate? = null
) {
    // Конструктор без аргументов для Hibernate
    constructor() : this(0, 0, 0, 0, null)
}

Acer@DESKTOP-8D49N84 MINGW64 ~/grade-api/grade-api (Razumnov)
$ cat src/main/kotlin/com/example/gradeapi/dto/GradeDto.kt
package com.example.gradeapi.dto

import io.swagger.v3.oas.annotations.media.Schema
import jakarta.validation.constraints.Max
import jakarta.validation.constraints.Min
import jakarta.validation.constraints.NotNull

data class GradeDto(
    @Schema(description = "Unique ID of the grade", example = "1")
    val id: Long? = null,

    @field:NotNull(message = "Student ID is required")
    @field:Min(value = 1, message = "Student ID must be at least 1")
    @Schema(description = "Student ID", example = "123", required = true)
    val studentId: Long,

    @field:NotNull(message = "Course ID is required")
    @field:Min(value = 1, message = "Course ID must be at least 1")
    @Schema(description = "Course ID", example = "456", required = true)
    val courseId: Long,

    @field:NotNull(message = "Grade value is required")
    @field:Min(value = 1, message = "Grade must be at least 1")
    @field:Max(value = 5, message = "Grade cannot be more than 5")
    @Schema(description = "Grade value (1-5)", example = "5", required = true)
    val value: Int,

    @Schema(description = "Date of the grade", example = "2025-06-15")
    val date: String? = null
)
```



Этап проектирования

Пример заполнения

Отладка программных модулей

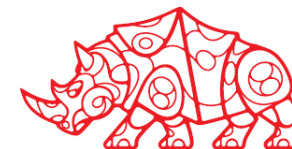
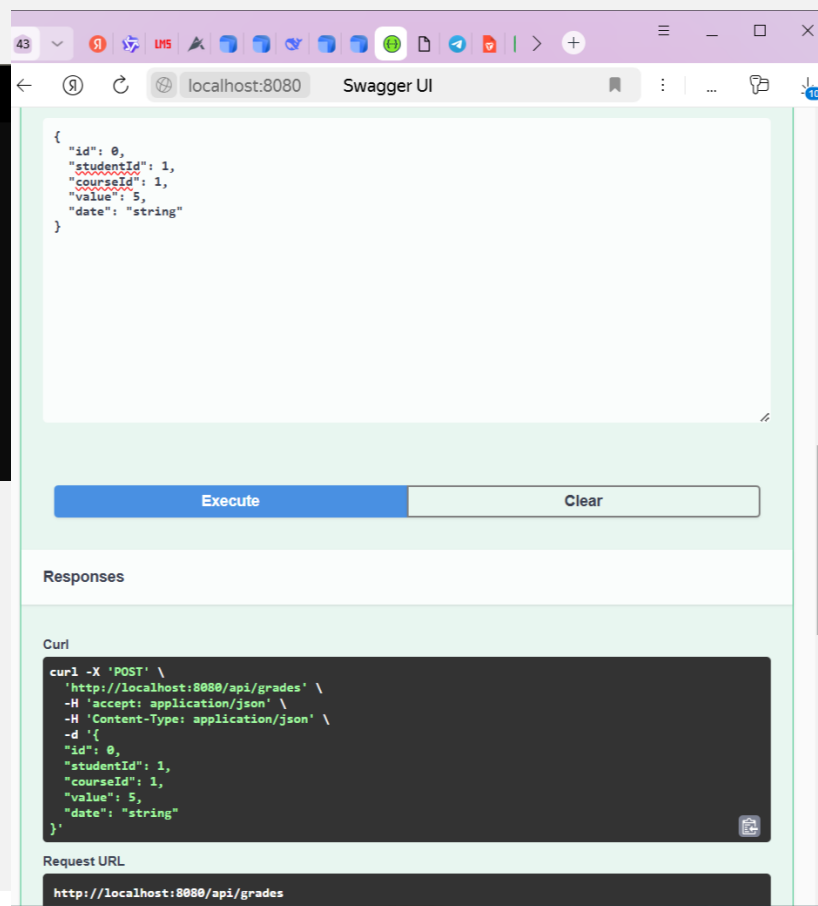
продемонстрировать итоговый скриншот отладки разработанных модулей.

```
C:\> MINGW64:/c/Users/Acer

Acer@DESKTOP-8D49N84 MINGW64 ~ (Razumnov)
$ curl -X POST http://localhost:8080/api/grades \
> -H "Content-Type: application/json" \
> -d '{"studentId": 1, "courseId": 101, "value": 5, "date": "2025-06-16"}'
{"id":1,"studentId":1,"courseId":101,"value":5,"date":"2025-06-16"}
Acer@DESKTOP-8D49N84 MINGW64 ~ (Razumnov)
$ curl http://localhost:8080/api/grades
[{"id":1,"studentId":1,"courseId":101,"value":5,"date":"2025-06-16"}]
Acer@DESKTOP-8D49N84 MINGW64 ~ (Razumnov)
$
```

```
localhost:8080 localhost:8080/api/grades
Автоформатировать ☒

[
  {
    "id": 1,
    "studentId": 1,
    "courseId": 101,
    "value": 5,
    "date": "2025-06-16"
  }
]
```

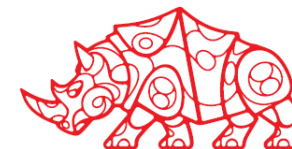


Проектный этап

Формирование отчетной документации по результатам работ

При оформлении отчетных материалов следует придерживаться действующих стандартов.

- В соответствии с ГОСТ 2.105-79 «Общие требования к текстовым документам» иллюстрации (графики, схемы, диаграммы) могут быть приведены как в основном тексте, так и в приложении. Все иллюстрации именуют рисунками. Все рисунки, таблицы и формулы нумеруют арабскими цифрами последовательно (сквозная нумерация) или в пределах раздела (относительная нумерация). В приложении - в пределах приложения. Каждый рисунок должен иметь подрисуночную подпись - название, помещаемую под рисунком.
- Рисунки следует размещать так, чтобы их можно было рассматривать без поворота страницы. Если такое размещение невозможно, рисунки следует располагать так, чтобы для просмотра надо было повернуть страницу по часовой стрелке. В этом случае верхним краем является левый край страницы. Расположение и размеры полей сохраняются.
- Номер таблицы размещают в правом верхнем углу или перед заголовком таблицы, если он есть. Заголовок, кроме первой буквы, выполняют строчными буквами. Ссылки на таблицы в тексте пояснительной записки указывают в виде слова «табл.» и номера таблицы. *Например: Результаты тестов приведены в табл. 4.*



Проектный этап

Формирование отчетной документации по результатам работ

При оформлении отчетных материалов следует придерживаться действующих стандартов.

- Список литературы должен включать все использованные источники. Сведения о книгах (монографиях, учебниках, пособиях, справочниках и т.д.) должны содержать: фамилию и инициалы автора, заглавие книги, место издания, издательство, год издания. При наличии трех и более авторов допускается указывать фамилию и инициалы только первого из них со словами «и др.». Издательство надо приводить полностью в именительном падеже: допускается сокращение названия только двух городов: Москва (М.) и Санкт-Петербург (СПб.).
- Сведения о статье из периодического издания должны включать: фамилию и инициалы автора, наименование статьи, издания (журнала), серии (если она есть), год выпуска, том (если есть), номер издания (журнала) и номера страниц, на которых помещена статья.
- При ссылке на источник из списка литературы (особенно при обзоре аналогов) надо указывать порядковый номер по списку литературы, заключенный в квадратные скобки; например: [5].



Отчетный этап

Выводы о результатах прохождения производственной практики:

выполняемая работа, приобретенные умения и навыки

Подведите итоги прохождения производственной практики:

В ходе прохождения производственной практики мной были освоены следующие навыки:

Разработка Spring Boot приложений на Kotlin

Работа с базами данных

REST API разработка

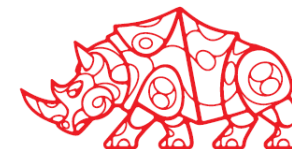
Валидация и обработка ошибок

Документирование API

Система сборки и развёртывания

Инструменты разработки

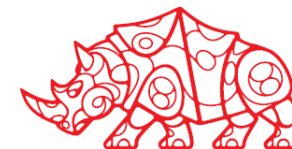
Тестирование



Отчетный этап

Список используемой литературы

1. Советов, Б. Я. Базы данных : учебник для среднего профессионального образования / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 3-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 420 с. — (Профессиональное образование). — ISBN 978-5-534-09324-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/514585>
2. Стружкин, Н. П. Базы данных: проектирование : учебник для среднего профессионального образования / Н. П. Стружкин, В. В. Годин. — Москва : Издательство Юрайт, 2023. — 477 с. — (Профессиональное образование). — ISBN 978-5-534-11635-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/518499>
3. Нагаева, И.А. Основы алгоритмизации и программирования: практикум : [12+] / И.А. Нагаева, И.А. Кузнецов. — Москва ; Берлин : Директ-Медиа, 2021. — 169 с. : схем. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/>
4. ...
5. ...



Отчетный этап

Приложения

Пример заполнения

1.1. Приложения 1.docx

1.2. Приложения 2.docx

