Федеральное государственное автономное образовательное учреждение высшего образования

«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий  
Кафедра «Информатика и информационные технологии»

Направление подготовки/ специальность: Автоматизированные системы обработки информации и управления

ОТЧЕТ

по проектной практике

Студентка: Азизова Марина Эдуардовна

Группа: 241-332

Место прохождения практики: Московский Политех, кафедра «Информатика и информационные технологии»

Отчет принят с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дата \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель практики: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Москва 2025

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

1. Общая информация о проекте:

* Название проекта
* Цели и задачи проекта

1. Общая характеристика деятельности организации *(заказчика проекта)*

* Наименование заказчика
* Организационная структура
* Описание деятельности

1. Описание задания по проектной практике
2. Описание достигнутых результатов по проектной практике

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

ПРИЛОЖЕНИЯ

# Введение

## Общая информация о проекте

Название проекта: «Собственный бизнес в партнёрстве с «Texel». Нейрофоторамки как новое слово в персонализации и умном декоре»

Цель проекта: Создать MVP продукта, которое автоматически анализирует гардероб пользователя и рекомендует оптимальные образы с учетом внешних факторов (погода, события) и индивидуального стиля.

Задачи проекта:

1. Проработка общей идеи и презентации (арт-стиля) проекта, выделяющихся на фоне существующих на рынке игр в жанре
2. Разработка бизнес-модели нейрофоторамок
3. Создание MVP: проверка дизайна и функционала умной рамки (при активной помощи ChatGPT).
4. Маркетинг и валидация: проведение фокус-групп, тестовые продажи, уточнение продуктового предложения на основе обратной связи.
5. Финансовое планирование и питчинг: подготовка инвест-презентации, расчёт себестоимости и потенциальной прибыльности, поиск грантов и инвестиций.
6. Углублённое обучение ChatGPT: участники получают навыки самостоятельного решения задач (код, копирайт, фин.модели) с поддержкой ИИ.

## Общая характеристика деятельности организации

Московский политехнический университет активно развивает практико-ориентированное обучение, предоставляя студентам возможность реализовывать собственные проекты в рамках учебного процесса. В университете особое внимание уделяется развитию проектной деятельности, направленной на формирование у студентов навыков командной работы, проектирования и практической реализации идей. В качестве заказчика проекта выступает сам Московский политех, который сравнительно недавно начал активно поддерживать студенческие инициативы через специализированные структуры, координирующие и сопровождающие выполнение учебных и исследовательских проектов.

На проектной деятельности мы разделились на множество команд, в конце выбиралось какая команда будет защищать свою тему на защите. Нашу команду выбрали кураторы для итоговой защиты. Тема нашего проекта: «Создание нейрофоторамки для умного дома».

По итогу кураторы выставили нашей команде максимальный балл за большой вклад в развитие проекта.

## Описание задания по проектной практике

### 1. Базовая часть задания:

* **Настроить Git и репозиторий** (на GitHub или GitVerse)
* **Оформить проектную документацию в формате Markdown**
* **Создать статический сайт проекта**, используя HTML/CSS или генератор Hugo, включив в него:
  + Домашнюю страницу с аннотацией проекта;
  + Страницу «О проекте»;
  + Страницу или раздел «Участники» с описанием личного вклада каждого участника группы в проект по «Проектной деятельности».
  + Страницу или раздел «Журнал» с минимум тремя постами (новостями, блоками) о прогрессе работы.
  + Страницу «Ресурсы» со ссылками на полезные материалы (ссылки на организацию-партнёра, сайты и статьи, позволяющие лучше понять суть проекта).
  + Медиа-контент: изображения, схемы и т.д.
* **Организовать взаимодействие с партнёрской организацией** (стажировка, мероприятие, онлайн-встреча), составить отчёт об этом опыте.

### 2. Вариативная часть задания:

В рамках вариативной части выбрана **практическая реализация технологии** — **создание Телеграмм бота для игры в шахматы на языке Python**.

Необходимо:

* Выполнить все пункты базовой части.
* Изучить структуру и механику классической игры «Шахматы».
* Проанализировать принципы разработки Телеграмм ботов на Python и выбрать соответствующий стек технологий.
* Реализовать игру «Шахматы» с нуля, оформив подробную техническую документацию:
  + пошаговое руководство по созданию игры;
  + фрагменты исходного кода;
  + визуальные материалы: схемы, диаграммы (в том числе UML) и иллюстрации.
* Разместить исходный код и документацию в репозитории.
* Провести творческую модификацию проекта — например, изменить визуальный стиль.
* Составить **финальный отчёт**, включающий:
  + хронологию выполнения всех этапов;
  + описание творческой модификации и полученных навыков.

## Описание достигнутых результатов по проектной практике

В ходе работы по выполнению проектной практики была выполнена вариативная часть задания.

**В ходе вариативной части задания:**

1. В ходе выполнения проекта были значительно углублены знания и практические навыки работы с языком программирования Python. Ранее в рамках учебной программы не рассматривался курс на данном языке программирования. При создании Telegram-бота для игры в шахматы на Python необходимо в первую очередь изучить особенности шахматного движка и его интеграции с Telegram API. Важно выбрать подходящую шахматную библиотеку, такую как python-chess, которая предоставляет все необходимые функции для работы с шахматной доской, валидации ходов и определения игровых состояний. Необходимо разобраться с генерацией шахматных нотаций (FEN, PGN) для сохранения и отображения позиций, а также реализовать механизм обработки шахматных ходов, передаваемых пользователями в текстовом формате (например, "e2-e4"). Особое внимание стоит уделить визуализации шахматной доски, которая может быть реализована через отправку изображений сгенерированной доски или использованием Unicode-символов для текстового представления.
2. Для эффективной разработки бота необходимо освоить специализированные Python-библиотеки, такие как python-telegram-bot (PTB) или aiogram, которые значительно упрощают взаимодействие с Telegram API. Библиотека python-telegram-bot предлагает синхронный подход, в то время как aiogram построен на асинхронном программировании и предоставляет более современные возможности. Также стоит рассмотреть библиотеку Telebot (pyTelegramBotAPI), которая отличается простотой использования и хорошо подходит для начинающих разработчиков. Важно изучить основные классы и методы этих библиотек, чтобы понимать, как обрабатывать входящие сообщения и отправлять ответы. Особое внимание следует уделить асинхронным возможностям aiogram, так как они позволяют создавать более производительных ботов.
3. Архитектура шахматного бота должна включать систему хранения игровых состояний для каждого активного матча, что особенно важно при игре между двумя пользователями. Необходимо разработать механизм очереди ходов, который будет корректно обрабатывать последовательность действий игроков и предотвращать попытки сделать ход не в свою очередь. Для многопользовательских игр потребуется реализовать систему приглашений и лобби ожидания, а также механизм таймаутов для автоматического завершения игры при неактивности одного из участников. Важно предусмотреть различные режимы игры, включая классические шахматы, блиц и игру против ИИ разного уровня сложности, что потребует интеграции с шахматными движками типа Stockfish.
4. Завершающим этапом стало составление подробной документации, в которой описаны все ключевые шаги по созданию проекта. Документ содержит хронологически выстроенные этапы разработки, начиная с подготовки среды и заканчивая визуальными доработками. Основной целью создания такой документации было обеспечение понятного и доступного руководства, по которому другой начинающий разработчик, только осваивающий язык Python, мог бы воспроизвести технологию и самостоятельно создать аналогичный проект. Это позволяет использовать данный труд как учебное пособие по разработке графических приложений на Python.

**Основные классы и их логика:**

1. **Класс Board (Доска)**  
   Ядро шахматной логики. Реализует:

* 8x8 матрицу для хранения фигур
* Стандартную начальную расстановку фигур
* Механизм выполнения ходов с полной валидацией
* Обнаружение специальных ситуаций (шах, мат, пат)
* Обработку особых правил: рокировки, взятия на проходе, превращения пешки
* Проверку доступных ходов для определения мата

1. **Базовый класс Piece (Фигура)**  
   Абстрактный базовый класс для всех фигур содержит:

* Цвет фигуры (белые/черные)
* Общие проверки для всех ходов
* Механизм проверки не приводит ли ход к шаху
* Абстрактный метод move\_to() для реализации в наследниках

1. **Классы фигур (наследники Piece)**  
   Каждый реализует уникальную логику:

* **Pawn (Пешка)**: особые правила первого хода, взятия по диагонали, превращения
* **King (Король)**: ход на 1 клетку, проверка шаха, логика рокировки
* **Queen (Ферзь)**: комбинация ладьи и слона
* **Rook (Ладья)**: движение по прямым
* **Bishop (Слон)**: движение по диагоналям
* **Knight (Конь)**: ход буквой "Г"

1. **Класс Game (Игра)**  
   Обрабатывает игровой процесс:

* Состояние текущей партии
* Поиск соперников через очередь
* Взаимодействие с пользователем
* Координация работы всех компонентов
* Отправка визуализации доски

1. **Класс ImageBoard (Визуализация)**  
   Генерирует графическое представление:

* Загружает изображения фигур и доски
* Рассчитывает позиции фигур на доске
* Накладывает фигуры на основу доски
* Сохраняет результат в PNG формате

1. **Класс DBParser (Работа с БД)**  
   Управляет хранением данных:

* Текущие активные игры
* Очередь ожидающих соперника
* Рейтинги и статистика игроков
* Сохранение/загрузка состояния доски

Особенность архитектуры - четкое разделение ответственностей:

* Шахматная логика изолирована в Board/Piece
* Визуализация - в ImageBoard
* Работа с Telegram API - в главном модуле
* Сохранение состояния - в DBParser

# Заключение

В ходе выполнения практики были получены знания по работе с самыми разными технологиями. Были получены знания по Python в реализации технологии, которая представляет из себя Телеграмм бота для игры в Шахматы.

## **Список литературы**

1. Официальная документация Python ([python.org](https://www.python.org/doc/))
2. "Создание чат-ботов для начинающих: Telegram и Python" (Артём Демиденко) – Введение в разработку ботов с примерами кода.
3. Chess Programming Wiki ([chessprogramming.org](https://www.chessprogramming.org/)) – База знаний по алгоритмам шахматных программ