

### **Инвариантная самостоятельная работа 3. Разработка схем использования технологий электронного обучения при организации и осуществлении корпоративного обучения.**

**Общая интеллект-карта для ТЭО:**



**Урок: Управление зависимостями и изоляция окружений в Python**  
**Общий хронометраж: ~90 минут**

#### **1. Вводная часть и мотивация (10 минут)**

**Формат:** интерактив (вопросы), обсуждение, вводная часть лекции

**Инструменты:** Zoom/Яндекс.Телемост + презентация

- Наводящие вопросы:
  - Почему иногда проект "не запускается" на другой машине?
  - Почему важно фиксировать зависимости и версии?
- Цель: показать значимость темы и пробудить интерес.
- Анонс: сегодня научимся управлять зависимостями с помощью современных инструментов.

#### **2. Теоретическая часть с демонстрациями в VS Code (45 минут)**

**Формат:** Zoom/Яндекс.Телемост + презентация + live-демо в VS Code + материалы на GitHub

**Платформа:**

- LMS: текст лекции и презентация

- GitHub: репозиторий с кодом параллельного показа live-демо в VS Code

## **2.1 Что такое зависимости? (5 мин)**

## **2.2 Что значит "управление зависимостями" и зачем это нужно? (5 мин)**

## **2.3 Какой подход самый надежный? (10 мин)**

## **2.4 Какими бывают зависимости? (5 мин)**

## **2.5 Обзор инструментов (30 мин)**

### **3. Видеофрагмент + круглый стол (15–20 минут)**

**Формат:** просмотр видеоролика + обсуждение в группе / чате / онлайн

- Просмотр короткого видеоролика (3–5 минут)  
Пример: передовой пакетный менеджер
- Обсуждение:
  - Какой пакетный менеджер лидирует на данный момент?
  - Кто его создал и какие еще приложения выпускала эта компания?
  - В чем преимущество uv?
  - Какие еще пакетные менеджеры упоминаются в видео?
  - Какой основной фактор для появления uv был назван в видео?
  - Почему packaging в питоне это сложная проблема для решения?
  - Почему пакетные менеджеры так важны для программистов?
  - Как вы думаете в каком направлении будут развиваться пакетные менеджеры?

### **4. Практическая часть: объяснение лабораторной работы (15–20 минут)**

**Формат:** демонстрация в LMS + GitHub + ответы на вопросы

- Цели:
  - Установить Poetry и настроить проект
  - Разделить зависимости на prod и dev
  - Установить зависимости и запустить проект в двух окружениях
  - Проанализировать дерево зависимостей
- Ключевые команды:
  - poetry init, add, install, lock, show --tree, run
- Вопросы и ответы:
  - Как запустить тесты?
  - Как работает pyproject.toml?

- Методические рекомендации размещены в курсе + репозиторий на GitHub:  
<https://github.com/vichnya/calc>

## 5. Заключение и рефлексия (5 минут)

**Формат:** короткое обсуждение, фиксация выводов

- Что запомнилось по инструментам?
- Где можно применить изученное на практике?
- Сложности при выполнении
- Напоминание об отчёте и дедлайне

## Схема использования технологий электронного обучения



## Технологии и инструменты, задействованные в процессе:

Компонент	Применение
LMS (Moodle)	Управление доступом, контентом, проверкой тестов и отчетов

Видеолекция с демонстрацией	Обучающее видео + демонстрация в реальном времени через VS Code
GitHub	Размещение исходного кода проекта, возможность форков
VS Code + терминал	Демонстрация команд <code>pip</code> , <code>venv</code> и др.
Интерактивные задания	Практика в LMS: загрузка отчетов, работа с кодом, ответ на контрольные вопросы
Обсуждение	Контакты для вопросов по теме