

Python: ООП и архитектура приложений

Научитесь развивать приложение без цепочки случайных правок и хрупких зависимостей.

120

академических часов

40 + 80

занятия + самоподготовка

8 + 1

проектов модулей + итог

Зачем проходить этот курс

- Разберётесь, где нужны классы, а где достаточно простой функции.
- Освоите SOLID и паттерны через изменения требований к одному приложению.
- Получите архитектурный кейс с тестами и объяснением принятых решений.

Кому подойдёт

Python-разработчикам, уверенно пишущим функции и скрипты, которым пора перейти к поддерживаемым приложениям.

Что нужно на старте

Нужны функции, коллекции, модули, исключения, основы Git и тестирования. Подойдёт подготовка на уровне курса Python с нуля.

Онлайн • Живые занятия • Группа до 8 человек

Понятный путь к результату

40 ак. ч. с преподавателем + 80 ак. ч. самостоятельной работы. Один академический час - 45 минут. Практика и проекты входят в указанный объём.

Модуль	Занятия	Самост.
01. Объектная модель и окружение	5	10
02. Инкапсуляция и жизненный цикл	5	10
03. Протоколы объектов и итерация	5	10
04. Функции и декораторы	5	10
05. SOLID и внедрение зависимостей	6	12
06. Наследование и прикладные паттерны	5	10
07. События и слабая связанность	5	10
08. Рефакторинг и защита архитектуры	4	8

Как устроена практика

На занятиях разбираются подход и ключевые решения. После каждого модуля вы завершаете мини-проект, проверяете его по критериям и оформляете результат. Итоговый проект объединяет эти работы; это один развиваемый кейс, а не дополнительная нагрузка сверх 120 часов.

Инструменты курса

Python 3 • dataclasses • typing / Protocol • abc • functools • pytest • Git • SOLID • Factory • Observer

Ведущий преподаватель

Сергей Валерьевич Герасименко - технический директор и основатель IT-компаний «ТЕРАКОР» и «ТЕРАКОР Digital», ведущий преподаватель и практикующий эксперт.

Изучаете. Применяете. Проверяете.

МОДУЛЬ 01 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно

Объектная модель и окружение

- Структура пакета, окружение и тестовый запуск
- Классы, экземпляры, self и модель ссылок
- dataclasses и различие данных и поведения
- Доменные сущности и простые инварианты

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Модель заказа

Создать Order и OrderItem с расчётом суммы и примерами использования.

Готово, когда: Состояние экземпляров независимо; недопустимое количество отклоняется; сумма проверена.

МОДУЛЬ 02 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно

Инкапсуляция и жизненный цикл

- Свойства, методы изменения состояния и @property
- Публичный контракт и внутренние детали реализации
- Валидация и доменные исключения
- Name mangling без иллюзии защиты доступа

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Жизненный цикл заказа

Реализовать создание, подтверждение и отмену заказа с ограничениями переходов.

Готово, когда: Нельзя подтвердить пустой заказ или изменить отменённый; ошибки объясняют причину.

Изучаете. Применяете. Проверяете.

МОДУЛЬ 03 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно

Протоколы объектов и итерация

- `__str__`, `__repr__`, `__eq__` и понятное поведение
- Контейнерные протоколы и итераторы
- Генераторы для последовательной обработки
- Композиция вместо ненужного наследования

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Каталог и выборка заказов

Создать коллекцию с поиском и ленивой выборкой по условию.

Готово, когда: Повторный обход работает предсказуемо; выборка не меняет исходную коллекцию.

МОДУЛЬ 04 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно

Функции и декораторы

- Функции первого класса и замыкания
- `functools.wraps` и параметры декоратора
- Логирование, измерение времени и ограничения кэширования
- Тестирование обёрток и сохранение контракта

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Набор диагностических декораторов

Добавить измерение времени и безопасный журнал вызовов без изменения бизнес-функций.

Готово, когда: Метаданные сохранены; исключения не скрыты; чувствительные аргументы не попадают в журнал.

Изучаете. Применяете. Проверяете.

МОДУЛЬ 05 / 6 ак. ч. занятий + 12 ак. ч. самостоятельно

SOLID и внедрение зависимостей

- SRP и разделение бизнес-логики и инфраструктуры
- OCP, LSP и ISP на изменении требований
- DIP, ручной DI, Protocol и ABC
- Тестовые двойники и проверка контрактов

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Сервис заказов с заменяемым хранилищем

Отделить use cases от хранения и подставить in-memory реализацию в тестах.

Готово, когда: Сервис не создаёт хранилище внутри себя; тесты бизнес-правил работают без внешней БД.

МОДУЛЬ 06 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно

Наследование и прикладные паттерны

- Композиция, наследование, super и обзор MRO
- Strategy для изменяемых правил
- Factory для выбора реализации
- Builder только для сложной пошаговой сборки

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Модуль расчёта стоимости

Добавить стратегии скидок и фабрику выбора правила по конфигурации.

Готово, когда: Новая стратегия подключается без правок существующих; проверены минимум два сценария.

Изучаете. Применяете. Проверяете.

МОДУЛЬ 07 / 5 ак. ч. занятий + 10 ак. ч. самостоятельно

События и слабая связанность

- Observer и внутрипроцессный Pub/Sub
- События, подписчики и явные контракты
- Обработка ошибок и повторных вызовов
- Границы локальной шины и отличие от брокера сообщений

ПРОЕКТ ПОСЛЕ МОДУЛЯ / Событийные уведомления

Подключить журнал и уведомления к событию подтверждения заказа.

Готово, когда: Подписчики тестируются отдельно; поведение при сбое задано; повтор не создаёт лишнее действие.

МОДУЛЬ 08 / 4 ак. ч. занятий + 8 ак. ч. самостоятельно

Рефакторинг и защита архитектуры

- Регрессионные тесты перед изменениями
- Сборка пакета и конфигурация приложения
- README, схема зависимостей и решения
- Демонстрация расширения без переписывания ядра

ПРОЕКТ ПОСЛЕ МОДУЛЯ / OrderDesk

Собрать приложение, добавить новый обработчик события и объяснить архитектуру.

Готово, когда: Тесты проходят; зависимости направлены к контрактам; показано изменение требований.

Ваш главный кейс в портфолио

OrderDesk: расширяемая система заказов

Приложение управляет заказами, рассчитывает стоимость по сменным правилам и отправляет события подключаемым обработчикам. Хранилище и уведомления передаются через явные зависимости.

Критерии готовности

- Бизнес-правила отделены от интерфейса и хранения.
- Добавлены две стратегии расчёта и два обработчика событий.
- Контракт хранилища проверен на основной и тестовой реализации.
- Запрещённые переходы состояния и ошибки обработчиков проверены тестами.
- README объясняет архитектуру и демонстрирует добавление нового компонента.

Что останется после обучения

Репозиторий с исходным кодом, учебные данные или конфигурация стенда, автоматические проверки, инструкция запуска и описание решений. Это материал для демонстрации навыков на работе и техническом собеседовании.

Как представить результат

Покажите задачу, запустите основной сценарий, продемонстрируйте обработку ошибки и объясните один архитектурный выбор. Отдельно зафиксируйте, что можно улучшить следующим шагом.

Итоговый кейс собирается из проектов модулей. Его завершение и подготовка защиты включены в часы последнего модуля.

Что вы сможете делать

- Моделировать предметную область и защищать инварианты объектов.
- Выбирать композицию, наследование и протоколы под конкретную задачу.
- Подменять внешние зависимости и тестировать бизнес-логику отдельно.
- Добавлять способы хранения и уведомления без переписывания ядра.

Практическое обучение в ТЕРАКОР

Живой онлайн-формат позволяет задавать вопросы и разбирать решения. Программа сочетает занятия с преподавателем и самостоятельную практику. После успешного прохождения - сертификат ТЕРАКОР и оформленный проект.

Перед записью

Обсудите исходный уровень, рабочую задачу и время для практики. Для уверенного результата важно проходить модули последовательно и завершать проекты. Курс развивает конкретные навыки; профессиональный уровень зависит также от предыдущего опыта и объёма самостоятельной работы.

Начните с консультации по программе

Подберём стартовый уровень и уточним формат обучения.

Открыть страницу курса → | info@terakor.ru

+7 (960) 352-38-05 • terakor.ru

Сроки старта, стоимость и состав выбранного тарифа уточняются при записи.

Программа оптимизирована на основе страницы курса ТЕРАКОР, изученной 29.09.2026. Это предлагаемая редакция для обновления курса и его описания.

Источник: terakor.ru/catalog/python/python-II/