Тип работы: Курсовая работа

Предмет: Программирование

1.1 Обоснование выбора темы
Выбор темы "Класс как тип данных в интернет-программировании" обусловлен активным развитием объектно-ориентированного программирования и его важной ролью в современной веб-разработке. Классы в языке программирования Python играют ключевую роль в построении сложных структур данных и обеспечивают гибкость и повторное использование кода. Исследование этой темы позволит глубже понять, как классы используются для создания высокоуровневых веб-приложений, а также выделить особенности их применения в контексте веб-разработки.
1.2 Актуальность проблемы
Современный мир веб-разработки сталкивается с растущей сложностью проектов, требующих эффективных средств структурирования и организации кода. Классы, как ключевой элемент объектно-ориентированного программирования, предоставляют разработчикам мощный инструмент для создания модульных и гибких приложений. Исследование актуальности использования классов в веб-разработке позволит выявить новые тренды и оптимальные подходы к проектированию веб-приложений.
1.3 Цель и задачи исследования
Цель: Исследование классов как типа данных в интернет-программировании направлено на выявление принципов и методов их эффективного использования для создания высококачественных и поддерживаемых веб-приложений.
Задачи:
• Изучить основы объектно-ориентированного программирования и его применение в веб-разработке.
• Анализировать роль и структуру классов в языке программирования Python.
• Рассмотреть процесс создания новых классов, их свойства и методы в контексте веб-приложений.
• Сравнить классы с другими концепциями, такими как библиотеки, и выделить отличия и схожести.
• Исследовать практические аспекты использования классов в веб-разработке с примерами на Python.
1.4 Методология исследования
Для достижения поставленной цели и решения поставленных задач будет использовано комплексное методологическое подход. В рамках исследования предполагается:
• Систематизация и анализ существующей литературы, статей и руководств по объектно-ориентированному программированию и веб-разработке на Python.
• Эмпирический анализ примеров использования классов в веб-проектах, включая изучение открытых исходных кодов приложений.
• Проведение сравнительного анализа между классами и библиотеками, сфокусированный на особенностях их использования в интернет-программировании.
• Разработка примеров кода и их апробация в современных веб-фреймворках.
Этот подход позволит получить всестороннее понимание роли классов в веб-разработке на Python и выявить оптимальные методы их применения.

II. Основные понятия и принципы объектно-ориентированного программирования
2.1 Объектно-ориентированное программирование в Python
Объектно-ориентированное программирование (ООП) представляет собой методологию, основанную на

использовании объектов, которые могут содержать данные в виде полей (или свойств) и кода в виде процедур (или методов). Python — мощный и гибкий язык программирования, поддерживающий принципы ООП.
В Python ООП включает в себя следующие основные принципы:
• Инкапсуляция: Механизм, который позволяет объединить данные (поле) и код, который работает с данными (метод), в единую единицу (класс). Инкапсуляция помогает управлять доступом к данным и защищать их от внешних изменений.
• Наследование: Возможность создавать новый класс на основе существующего (родительского) класса. Этот новый класс (потомок) может наследовать свойства и методы родительского класса, что способствует повторному использованию кода и созданию иерархии классов.
• Полиморфизм: Возможность использовать объекты различных типов с единым интерфейсом. Полиморфизм включает в себя перегрузку операторов и методов, что позволяет объектам разного типа взаимодействовать с едиными интерфейсами.

8.1 Литература
1. Lutz, M. (2013). "Learning Python." O'Reilly Media.
2. VanderPlas, J. (2016). "Python Data Science Handbook." O'Reilly Media.
3. Beazley, D. M. (2009). "Python Essential Reference." Addison-Wesley.
8.2 Онлайн-ресурсы
1. Python Official Documentation: https://docs.python.org
2. Flask Documentation: https://flask.palletsprojects.com
3. SQLAlchemy Documentation: https://docs.sqlalchemy.org
4. WTForms Documentation: https://wtforms.readthedocs.io
8.3 Ссылки на стандарты и документацию Python
1. PEP 8 -- Style Guide for Python Code: https://www.python.org/dev/peps/pep-0008
2. PEP 257 -- Docstring conventions: https://www.python.org/dev/peps/pep-0257
3. PEP 333 -- Python Web Server Gateway Interface (WSGI): https://www.python.org/dev/peps/pep-0333
4. PEP 484 -- Type Hints: https://www.python.org/dev/peps/pep-0484
5. PEP 572 -- Assignment expressions: https://www.python.org/dev/peps/pep-0572