

Очеев Е.В.

студент 4 курса, кафедра ИТиВС

ФГАОУ ВО «МГТУ «СТАНКИН»

Россия, г. Москва

PYTHON В СОВРЕМЕННОЙ РАЗРАБОТКЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

***Аннотация.** Рассмотрена роль Python в современной разработке программного обеспечения. На основе русскоязычных научных публикаций показано применение языка в backend-разработке, микросервисах и машинном обучении. Выделены преимущества, связанные с развитой экосистемой и скоростью реализации прикладной логики, а также ограничения производительности и сопровождения. Сделан вывод о необходимости выбирать Python с учетом архитектуры и характера программной задачи.*

***Ключевые слова:** программирование; программное обеспечение; Python; backend; веб-разработка; машинное обучение.*

Ocheev E.V.

fourth-year student, Department of IT and Computer Systems

Moscow State University of Technology «STANKIN»

Russia, Moscow

PYTHON IN MODERN SOFTWARE DEVELOPMENT

***Abstract.** The article examines the role of Python in modern software development. Russian-language scientific publications demonstrate its use in backend development, microservices, and machine learning. The study identifies advantages related to the ecosystem and development speed as well as performance and maintenance limitations. The choice of Python should therefore depend on software architecture and task requirements.*

***Keywords:** programming; software; Python; backend; web development; machine learning.*

ВВЕДЕНИЕ

Современная разработка программного обеспечения требует одновременно сокращать сроки реализации функций и сохранять возможность дальнейшего сопровождения системы. В этих условиях язык программирования оценивается не только по синтаксису, но и по экосистеме библиотек, средствам интеграции и поддерживаемым архитектурным подходам. Python применяется в разных классах задач и особенно заметен там, где серверная логика связана с обработкой данных. Практика создания веб-приложений на Django показывает возможность использовать Python для маршрутизации запросов, взаимодействия с базой данных, аутентификации и реализации прикладных функций в единой серверной среде [1, с. 334-337].

Цель исследования состоит в определении роли Python в современной прикладной разработке и условий его рационального использования. Методическую основу составили анализ и сопоставление четырех русскоязычных научных публикаций 2023-2024 гг. В качестве основных направлений рассмотрены backend-разработка, микросервисная интеграция и машинное обучение. Такой подход позволяет оценивать Python не как универсально лучший язык, а как технологический инструмент, эффективность которого зависит от типа программной системы.

ОСНОВНЫЕ НАПРАВЛЕНИЯ ПРИМЕНЕНИЯ PYTHON

В backend-разработке важным преимуществом Python является наличие фреймворков, в которых типовые задачи серверного приложения уже представлены готовыми механизмами. Это уменьшает объем повторяющегося кода и позволяет разработчику сосредоточиться на предметной логике. Для полнофункциональных веб-систем такой подход

реализуется, например, через Django. Для более компактных сервисов применяется модель API, когда отдельный компонент предоставляет ограниченный набор функций другим частям системы.

Второе направление связано с машинным обучением и анализом данных. В исследовании К.А. Конева библиотеки Python используются для реализации дерева решений в системе поддержки принятия решений промышленного предприятия [2, с. 696-701]. Значение Python в таком сценарии состоит в возможности объединить подготовку данных, применение алгоритма и получение результата в одном программном контуре. Это сокращает расстояние между экспериментальной моделью и ее прикладным использованием.

Особенно показателен микросервисный сценарий. В информационной системе анализа отзывов модуль определения эмоциональной окраски текста вынесен в отдельный компонент на Python, а взаимодействие с основной системой организовано через FastAPI [3, с. 13-16]. Подобная архитектура показывает, что Python может быть не единственным языком большого продукта, а специализированным интеграционным слоем для аналитических функций. Такая роль соответствует современной практике разделения систем на независимые сервисы.

Широкая область применения связана с особенностями самого языка: высокоуровневым синтаксисом, сравнительно компактной записью алгоритмов и большим количеством готовых библиотек. Обобщенный анализ Python отмечает его пригодность для веб-разработки, анализа данных, автоматизации и интеллектуальных приложений [4, с. 1114-1126]. В результате один и тот же язык может использоваться на этапах прототипирования, реализации серверного API и подготовки аналитического модуля.

ОГРАНИЧЕНИЯ И ИНЖЕНЕРНЫЙ ВЫБОР

Универсальность Python не означает одинаковой эффективности во всех задачах. При интенсивных вычислениях дополнительные накладные расходы высокоуровневого исполнения могут становиться существенными. На практике это ограничение часто компенсируется использованием оптимизированных библиотек или вынесением критических участков в специализированные компоненты. Поэтому при проектировании следует разделять удобство разработки прикладной логики и требования к предельной производительности.

В крупных проектах возрастают требования к структуре модулей, тестированию, версиям зависимостей и контролю типов. Простота синтаксиса не заменяет архитектурной дисциплины. Рациональным является применение Python там, где преимущества развитой экосистемы, быстрой реализации и интеграции превышают издержки сопровождения. Рассмотренные исследования показывают такой баланс в веб-системах, микросервисах и задачах машинного обучения [1, с. 334-337; 2, с. 689-703; 3, с. 13-16].

ЗАКЛЮЧЕНИЕ

Python сохраняет значимую роль в современной разработке благодаря сочетанию понятной прикладной модели, развитой экосистемы и удобства интеграции с инструментами анализа данных. Наиболее характерными областями являются backend-разработка, API и микросервисы, а также машинное обучение. При этом выбор языка должен учитывать профиль нагрузки, масштаб кодовой базы и требования к сопровождению. Таким образом, устойчивость Python определяется не отсутствием ограничений, а возможностью использовать его сильные стороны в тех архитектурных задачах, где скорость разработки и интеграция важнее предельной низкоуровневой производительности.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Баранова Е.М., Баранов А.Н., Борзенкова С.Ю., Шепаров И.С., Богданов А.М. Разработка безопасного веб-приложения для организации интернет-мероприятий // Известия Тульского государственного университета. Технические науки. 2024. № 9. С. 334-337. DOI: 10.24412/2071-6168-2024-9-334-335.

2. Конев К.А. Машинное обучение для поддержки принятия решений в сфере качества на промышленном предприятии // Экономика. Информатика. 2023. Т. 50, № 3. С. 689-703. DOI: 10.52575/2687-0932-2023-50-3-689-703.

3. Криворогов Д.Д., Низамов Т.Д., Фазлыев А.А., Ходырев А.Н., Шушарин Д.В., Глазкова А.В. Разработка сервиса для сбора и анализа отзывов на элективные дисциплины // Вестник НГУ. Серия: Информационные технологии. 2023. Т. 21, № 3. С. 5-19. DOI: 10.25205/1818-7900-2023-21-3-5-19.

4. Мадиярбекова А. Python как высокоуровневый язык программирования // Вестник науки. 2024. Т. 4, № 11 (80). С. 1114-1126. URL: <https://www.xn----8sbempclcwd3bmt.xn--p1ai/article/19068> (дата обращения: 09.08.2026).