

Введение в языки программирования





Что такое язык программирования

формальный язык,
предназначенный для записи
компьютерных программ

Набор

- лексических
- синтаксических
- семантических

правил

Классификация

- Языки низкого и высокого уровня
- Безопасные и небезопасные языки
- Компилируемые и интерпретируемые языки
- Особые категории языков

Низкий уровень

Ограничения на абстракции данных, сильная статическая типизация, отсутствие промежуточной среды выполнения, прямой доступ к памяти.

Примеры: C, C++, Assembler (язык Ассемблера)



Средний уровень

Фокус на абстракциях, сильная статическая типизация, среда выполнения, ограничения на прямой доступ к памяти.

Примеры: Java, C#.

Высокий уровень

Сильное абстрагирование,
динамическая и/или слабая
типизация, полностью независимое
управление памятью и/или наличие
среды выполнения.

Примеры: Python, Ruby, JavaScript,
Common Lisp.



Парадигма программирования


совокупность идей и понятий, определяющих стиль написания компьютерных программ (подход к программированию)

Декларативная

Императивная

По типизации

- Статическая и динамическая типизация
- Сильная и слабая типизация (также иногда говорят строгая и нестрогая)
- Явная и неявная типизация



Статическая и динамическая типизация

Статическая - конечные типы переменных и функций устанавливаются на этапе компиляции.

В динамической - во время выполнения программы.

Статическая: C, Java, C#;

Динамическая: Python, JavaScript, Ruby.




Сильная и слабая типизация

Сильная типизация не позволяет смешивать в выражениях различные типы и не выполняет автоматические неявные преобразования.

Языки со слабой типизацией выполняют множество неявных преобразований автоматически

Сильная: Java, Python, Haskell, Lisp;

Слабая: C, JavaScript, Visual Basic, PHP



Явная и неявная ТИПИЗАЦИЯ

Явно-типизированные языки отличаются тем, что тип новых переменных / функций / их аргументов нужно задавать явно. Соответственно языки с неявной типизацией перекладывают эту задачу на компилятор / интерпретатор.

Явная: C++, D, C#

Неявная: PHP, JavaScript, Python, Ruby, Perl, Lua, Swift

Python

высокоуровневый язык
программирования общего
назначения, ориентированный на
повышение производительности
разработчика и читаемости кода

Система типов сильная,
динамическая

Тип исполнения: интерпретируемый,
компилируемый в байт-код

Достоинства

- встроенные структуры данных, словари, кортежи;
- простой и удобный синтаксис;
- большое количество библиотек;
- мощные интерфейсы к конкретным ОС;
- переносимость кода между платформами: автоматическую генерацию документации на модули и возможность написания самодокументированных программ;
- поддержку процедурного, функционального и объектного стилей программирования; встроенную поддержку Unicode и большое количество национальных кодировок.

Недостатки

- низкая скорость выполнения
- динамическая типизация (?)

Области применения

- Разработка веб-приложений
- Анализ данных и машинное обучение
- Введение в программирование
- Быстрое прототипирование идей в бизнесе за счёт обилия готовых библиотек, низкого порога вхождения в язык и высокой продуктивности программистов, пишущих на Python
- Написание скриптов (сценариев)

Синтаксис языка



Документация

<https://pythonworld.ru/>

<https://www.python.org/doc/>

<https://ru.wikibooks.org/wiki/Python>