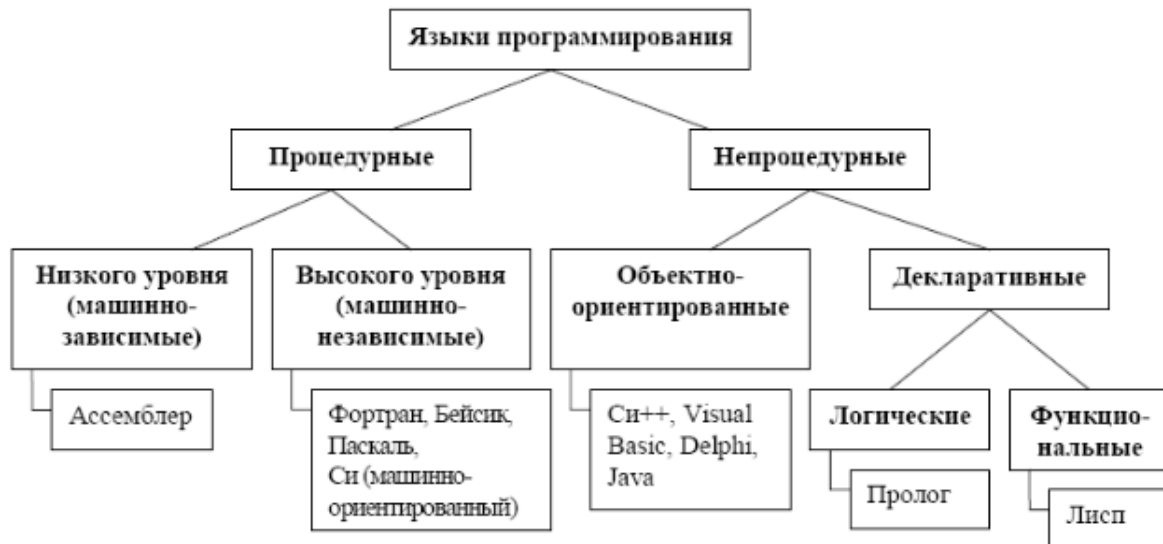


1. Назовите и дайте краткую характеристику основных классов языков программирования.



Все языки программирования разделяются на несколько классов:

Языки низкого уровня и языки высокого уровня

Языки низкого уровня - те языки программирования, которые близки к программированию непосредственно в машинных кодах процессора, но задающие минимальную абстракцию для хоть какого-то уровня читабельности кода. Используются преимущественно для узконаправленных задач вроде написания драйверов или прошивок микроконтроллеров.

Представителями являются:

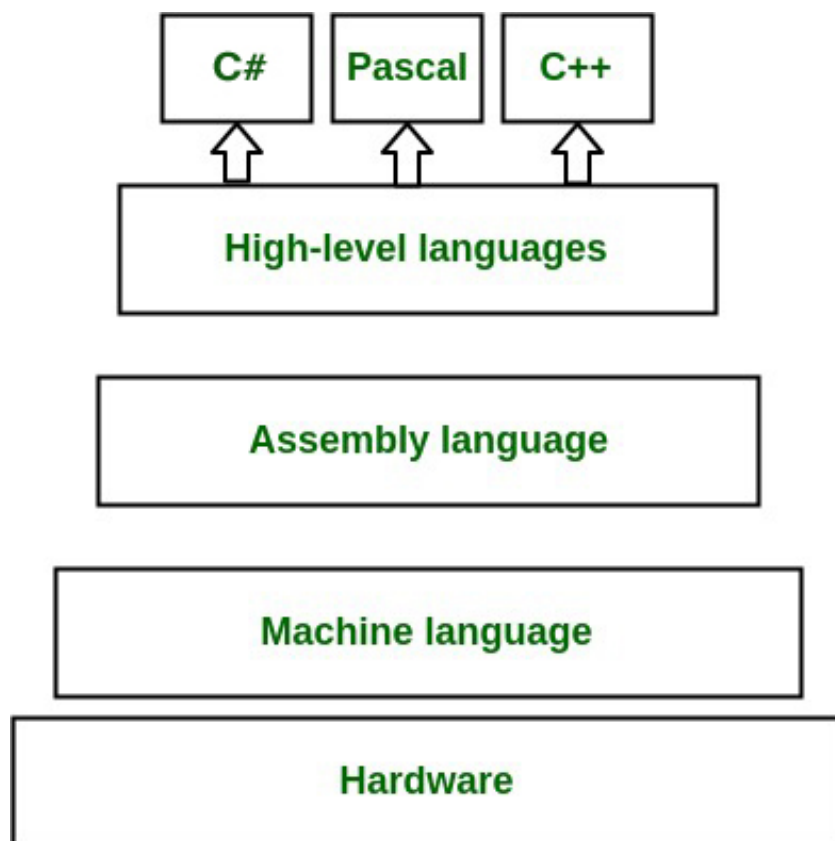
- Группа языков Ассемблера - транслирует данные в машинный код
- Машинный код — самый низкий уровень, 101011010101 и т д

Языки высокого уровня - все остальные языки, у которых уровень абстракции очень высокий. Очень удобны для программиста т.к. Исходный код таких языков читабелен, но как следствие, из-за специфики перевода на машинный код, производительность операций падает.

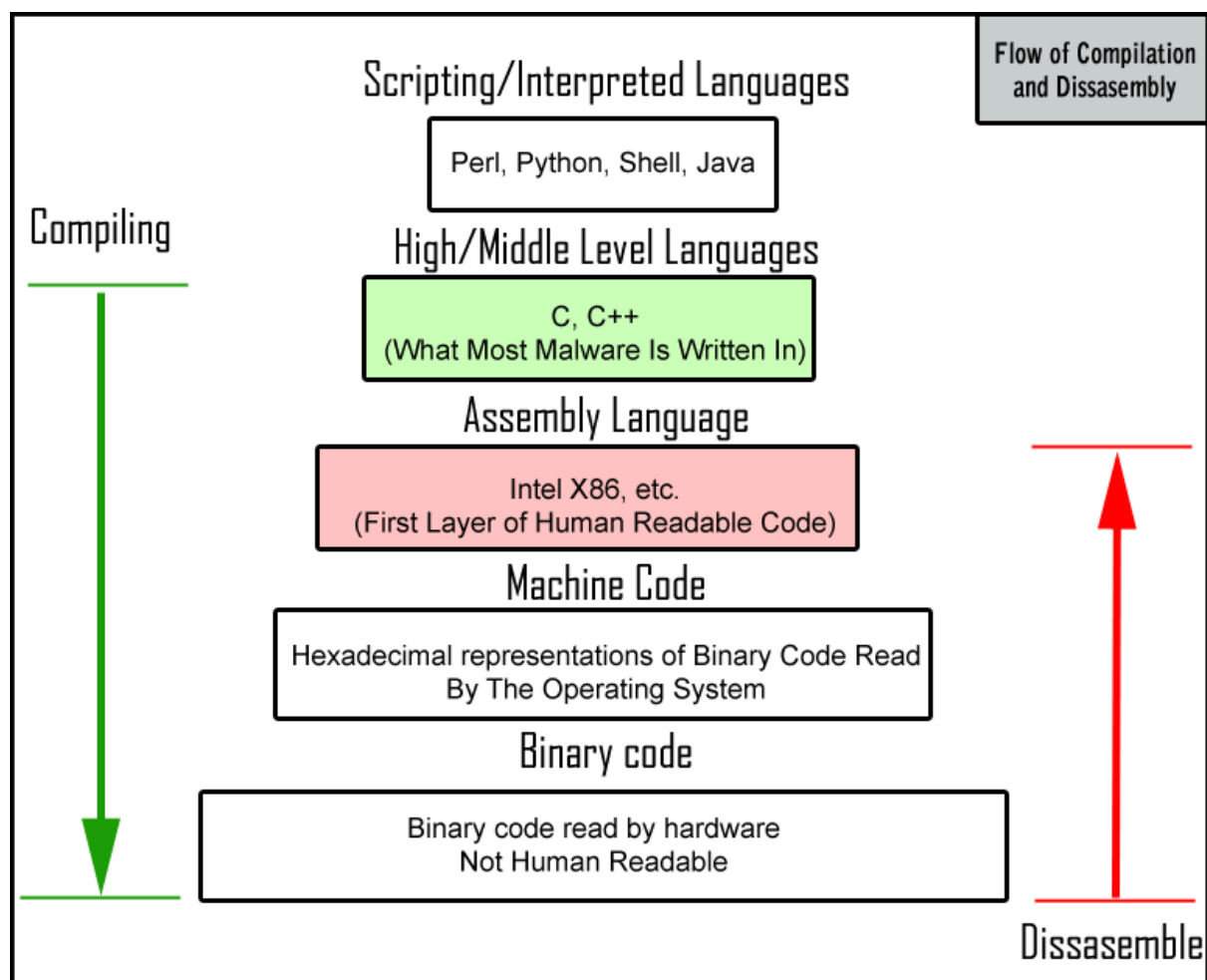
Особенность языка C

Язык Си (C) - НЕ ПУТАТЬ C C++ - транслируется в ассемблер, позволяет использовать ассемблерные вставки, находится уровнем выше него, но **является самым низким среди языков высокого уровня**. На нём реализовано ядро UNIX и СУБД MySQL с Oracle, поэтому иногда его относят к языкам низкого уровня, что, возможно, является ошибкой.

Суть трансляции/компиляции высокоуровневых языков программирования:



А для Python подходит даже вот эта картинка:



Машинозависимые языки и машиннезависимые языки программирования

Машинозависимые языки - те языки программирования, которые зависят от архитектуры процессора машины, например, **Ассемблер для ARM-процессора** несовместим с 86х архитектурой т.к. у 86 расширенное количество команд и в итоге программу требуется переписывать.

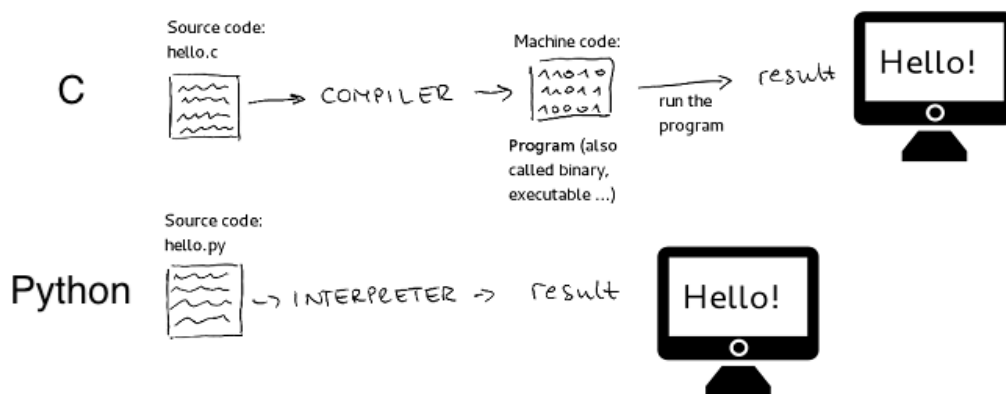
Машиннезависимые языки - те языки программирования, которые одинаково работают на всех архитектурах процессоров машин, как правило, еще и кроссплатформенны т.е. могут работать под разными ОС: Windows, Mac, Linux, iOS, Android

Примеры:

- C
- Python
- Java
- GoLang и т.д.

Компилируемые и интерпретируемые языки программирования

Различаются по способу преобразования исходного кода в машинный.



Компилируемые ЯП - их компилирует специальная программа-компилятор полностью читает весь исходный код программы, создает выходной бинарный файл .exe (если на Windows), содержащий инструкции и исполняет его. Компилируемые ЯП являются более быстрыми т.к. Исходный код читается сразу весь, в отличие от интерпретатора.

Примеры:

- C#
- C++
- GoLang
- C
- Java

Интерпретируемые ЯП - В отличие от компилируемых языков, интерпретируемым для исполнения программы не нужен машинный код. Вместо этого программу построчно исполняют интерпретаторы. Интерпретируемый код обычно медленнее. Никаких промежуточных файлов нет. В Python есть, конечно, .рус, но только для модулей, но быстрдействие он не особо повышает.

Примеры интерпретируемых ЯП:

- Python
- PHP
- Perl
- Ruby

Универсальные и узконаправленные ЯП

Универсальный язык программирования - язык, решающий широкий спектр задач, например, на Python можно работать с большим кол-вом данных за счёт библиотеки NumPy, компьютерным зрением за счёт OpenCV, проектированием REST API с помощью Flask и т.д.

Примеры помимо Python:

- Java
- C#
- GoLang

Узконаправленные языки программирования - используются для конкретных целей. Самый распространённый - SQL. Язык запросов для взаимодействия с SQL-подобными СУБД.

Объектно-ориентированные и функциональные ЯП

Объектно-ориентированный язык программирования - это язык, построенный на принципах объектно-ориентированного программирования, где в основе лежит объект и принципы ООП.

Примеры:

- C++
- C#
- Python
- Swift
- Objective-C

Функциональный ЯП - это язык, где в основе лежит функциональный подход для написания программы. Проще говоря, функция - это основной элемент языка.

Примеры:

- Erlang
- Lisp
- Scala

<http://www.apmath.spbu.ru/ru/education/final/question42new.pdf>