

1. زبان‌های کامپایلری (Compiled Languages)

در زبان‌های کامپایلری، کل کد برنامه ابتدا توسط یک کامپایلر به کد ماشین تبدیل می‌شود و سپس اجرا می‌شود. کامپایلر تمام کد را یک‌باره می‌گیرد، آن را به زبان ماشین ترجمه می‌کند و در نهایت یک فایل اجرایی (مانند فایل‌های exe در ویندوز) ایجاد می‌کند. این فایل مستقل است و می‌تواند بدون نیاز به کامپایلر مجدداً اجرا شود.

مثال‌هایی از زبان‌های کامپایلری: C، ++C، Rust، Go

مزایا:

سرعت اجرای بالا، زیرا کد به صورت کامل قبل از اجرا به زبان ماشین ترجمه شده است.

معمولاً بهینه‌سازی بیشتری دارد زیرا کامپایلر می‌تواند کل کد را بررسی کرده و بهینه کند.

معایب:

خطاها تنها پس از کامپایل کل برنامه مشخص می‌شوند.

در صورت ایجاد تغییر در کد، باید مجدداً کامپایل شود.

2. زبان‌های مفسری (Interpreted Languages)

در زبان‌های مفسری، کد برنامه خط به خط توسط یک مفسر به زبان ماشین ترجمه و اجرا می‌شود. به عبارتی دیگر، مفسر کد را در لحظه تفسیر و اجرا می‌کند و نیازی به تولید فایل اجرایی ندارد.

مثال‌هایی از زبان‌های مفسری: Python، JavaScript، Ruby، PHP

مزایا:

انعطاف‌پذیری بالا برای توسعه‌دهندگان، زیرا می‌توانند کد را خط به خط اجرا و اشکال‌زدایی کنند.

مناسب برای اسکریپت‌نویسی و توسعه سریع، زیرا نیازی به کامپایل مجدد پس از هر تغییر نیست.

معایب:

سرعت اجرای پایین‌تر نسبت به زبان‌های کامپایلری، زیرا هر خط کد در لحظه اجرا به زبان ماشین ترجمه می‌شود.

معمولاً بهینه‌سازی کمتری نسبت به زبان‌های کامپایلری دارد.

زبان‌های ترکیبی

برخی زبان‌ها از هر دو روش استفاده می‌کنند، به این معنی که ابتدا کد به یک زبان میانی (مانند بایت‌کد) کامپایل شده و سپس توسط یک ماشین مجازی مفسری اجرا می‌شود. این روش برای حفظ مزایای هر دو روش طراحی شده است.

مثال‌ها: Java (که کد را به بایت‌کد کامپایل و سپس توسط ماشین مجازی جاوا اجرا می‌کند)، #C

شباهت‌ها

هر دو نوع زبان به کاربر اجازه می‌دهند تا برنامه‌ی خود را در یک زبان خاص بنویسند و آن را به زبانی قابل فهم برای کامپیوتر ترجمه کنند.

هر دو نوع زبان از ساختارهای کنترلی مشابهی برخوردار هستند که شامل حلقه‌ها، شرطی‌ها و توابع است.

در هر دو نوع زبان، برنامه‌های نوشته شده می‌توانند بر روی سیستم‌عامل‌های مختلف اجرا شوند.

هر دو نوع زبان برای تولید برنامه‌هایی با کارایی بالا و سرعت اجرای بالا طراحی شده‌اند.

هر دو نوع زبان می‌توانند برای ایجاد برنامه‌های بزرگ و پیچیده استفاده شوند.

هر دو نوع زبان در صورت نوشتن کد درست و بهینه، می‌توانند به نتایج مشابهی در اجرای برنامه‌ها دست پیدا کنند.

تفاوت‌ها

روش اجرا: زبان‌های مفسری کد را در زمان اجرا تفسیر و خط به خط آن را اجرا می‌کنند. در حالی که زبان‌های کامپایلری کد را در زمان کامپایل ترجمه و خروجی کامپایل شده را اجرا می‌کنند.

سرعت اجرا: بدون شک، زبان‌های کامپایلری در سرعت اجرای برنامه‌ها بهتر هستند. چرا که کامپایلر، کد را به صورت کامل به زبان ماشین ترجمه کرده و خروجی آن بهینه و سریع است. در مقابل، زبان‌های مفسری برای هر دستوری که باید اجرا شود، باید کدها را تفسیر و سپس آن را اجرا کنند که باعث کاهش سرعت اجرای برنامه می‌شود.

اشکال‌زدایی: اشکال‌زدایی برای زبان‌های کامپایلری ممکن است کمی سخت باشد، زیرا خطاهایی که در زمان کامپایل رخ می‌دهند، باید توسط کدگذاری برطرف شوند. در مقابل، زبان‌های مفسری به دلیل تفسیر دستورات، می‌توانند اشکال‌زدایی را راحت‌تر پیش ببرند.

پرتاب‌های استثناء (exception handling): برخلاف زبان‌های کامپایلری، زبان‌های مفسری به صورت پیش فرض پرتاب استثناء دارند. این به معنای این است که در صورت بروز خطا در هنگام اجرای برنامه، برنامه متوقف و خطایی به کاربر نمایش داده می‌شود.