تمرین چهارم محمد شیخ پور

اگر انسان بخاهد با کامپیوتر صحبت بکند نیاز به مترجم دارد. نام این مترجم بین انسان و کامپیوتر، «کامپایلر» و «مفسر» است . کامپایلر و مفسر یک نرم‌افزاری هستند که کدهای برنامه نویسی را به «زبان ماشین» که برای کامپیوتر قابل فهم باشد تبدیل می‌کند . در زبان برنامه نویسی سطح پایین نیاز به زبان ماشین نداریم اما در زبان سطح بالا نیاز داریم

**کامپایلر چیست؟**

، یک نرم‌افزاری است که کدهای برنامه نویسی را به «زبان ماشین» تبدیل می‌کند.

در کامپایلر باید قبل از اجرای برنامه، کدها یکبار به زبان ماشین کامپایل شوند تا برنامه به درستی اجرا شود. هر تغییر کوچکی در کدها اعمال شود، باید یکبار کدها از اول کامپایلر شوند. برای اجرا ساخته می‌شود

چه زبان‌هایی کامپایلری هستند؟

1. C
2. ++C
3. #C
4. Java
5. Go

## مفسر چیست؟

## مانند کامپایلر، نرم‌افزاری است که کدهای برنامه نویسی را از زبان سطح بالای قابل فهم برای انسان به زبان قابل فهم برای کامپیوتر یا «زبان ماشین» تبدیل می‌کند

چه زبان‌هایی مفسری هستند؟

1. JavaScript
2. PHP
3. Python
4. Ruby

## تفاوت میان مفسر و کامپایلر چیست ؟

## در کامپایلر، پیش از اجرای برنامه باید کدها به «زبان ماشین» ترجمه شوند اما در مفسر موقع اجرای برنامه به‌صورت خط به خط به «زبان ماشین» ترجمه می‌شوند. در کامپایلرها قبل از اجرای برنامه وقتی کدها کامپایل می‌شوند یک فایل خروجی (مانند exe) به ما تحویل داده می‌شود اما در مفسر، فایل خروجی به ما تحویل داده نمی‌شود بنابراین برای هر بار اجرا، باید خط به خط کدها به زبان ماشین ترجمه شوند. در صورتی که در کامپایلر یک فایل نهایی ایجاد می‌شود که در اجرای بعدی نیازی به ترجمه کد نیست.

1. در زبان‌های کامپایلری قبل از اجرای برنامه، همه‌ی کدها به یکباره به «زبان ماشین» تبدیل می‌شوند. اما در زبان‌های مفسری موقع اجرای برنامه، کدها به‌صورت خط به خط به زبان ماشین ترجمه می‌شوند.
2. در زبان‌های کامپایلری بعد از کامپایل شدن یک فایل خروجی (مانند exe) ساخته می‌شود که به زبان ماشین ترجمه شده است و در دفعات بعدی نیازی به کامپایل و ترجمه دوباره کدها نیست. اما در زبان‌های مفسری هیچ فایل خروجی ساخته نمی‌شود و همیشه موقع اجرا، این کدها به‌صورت خط به خط به «زبان ماشین» ترجمه می‌شوند.
3. اگر خطایی در کد زبان‌های برنامه نویسی کامپایلری رخ بدهد، کامپایلر با خطای Compile error مواجه می‌شود و فایل خروجی ساخته نمی‌شود و نمی‌توان آن را اجرا کرد. اگر در کد زبان‌های برنامه نویسی مفسری خطایی رخ دهد، چون به‌صورت خط به خط این ترجمه انجام می‌شود، برنامه تا زمانی که خطایی در کد وجود نداشته باشد اجرا شده و موقع رسیدن به آن خط از کد که خطا وجود دارد، برنامه متوقف می‌شود.
4. سرعت اجرای برنامه در زبان‌های کامپایلری به دلیل اینکه خروجی کامپایل شده برای دفعات بعدی نیاز به ترجمه نیس، بالاتر از سرعت اجرای برنامه در زبان‌های مفسری است چون که همیشه موقع اجرا این ترجمه باید انجام شود و مقداری زمان‌بر است.
5. پیدا کردن باگ (Bug) در زبان‌های برنامه نویسی مفسری آسان‌تر از زبان‌های برنامه نویسی کامپایلری است، چون در زبان‌های مفسری به‌صورت خط به خط ترجمه انجام می‌شود و در هر خط از کد، خطا (Bug) وجود داشته باشد، برنامه متوقف شده و خطا در آن خط قابل مشاهده است اما در زبان‌های کامپایلری چون کل کدها کامپایل می‌شوند، موقع خطا، متوجه نمی‌شویم که در کدام یک از خط‌ها باگ وجود دارد.
6. زبان مفسری وابسته به سیستم‌عامل نیست اما زبان کامپایلری به سیستم‌عامل وابسته است. در زبان مفسری کدها چون در هر اجرای برنامه ترجمه می‌شوند، در هر سیستم‌عاملی قابل اجرا هستند. اما خروجی زبان کامپایلری، در همان سیستم‌عاملی قابل اجرا است که در همانجا کامپایل شده است. اگر بخواهیم در سیستم‌عامل دیگری استفاده کنیم، باید در همان سیستم‌عامل، کدها دوباره کامپایل شوند.

## منبع :

## https://developersho.com/blog/compiler-vs-interpreter#:~:text=%D9%BE%D8%B3%20%D8%AA%D9%81%D8%A7%D9%88%D8%AA%20%DA%A9%D8%A7%D9%85%D9%BE%D8%A7%DB%8C%D9%84%D8%B1%20%D8%A8%D8%A7%20%D9%85%D9%81%D8%B3%D8%B1,%C2%AB%D8%B2%D8%A8%D8%A7%D9%86%20%D9%85%D8%A7%D8%B4%DB%8C%D9%86%C2%BB%20%D8%AA%D8%B1%D8%AC%D9%85%D9%87%20%D9%85%DB%8C%E2%80%8C%D8%B4%D9%88%D9%86%D8%AF.