

## \* تفاوت کامپایلر و مفسر:

مفسر (Interpreter)

کامپایلر (Compiler)

۱- اجرای خط به خط

۱- اجرای کل برنامه

۲- زمان تحلیل کم است ولی زمان اجرای حالت کلی زیاد است.

۲- زمان ترجمه و تحلیل برنامه زیاد است ولی زمان اجرای حالت کلی کم است.

\* ۳- اشکال زدایی راحت تر است چرا که ترجمه آن زمانی که خط وجود ندارد، اتفاق افتاده است.

\* ۳- پیام خطا بعد از ترجمه کل برنامه اتفاق می افتد و بنابراین خطایابی و دیباگ کردن آن ممکن است نیاز به تلاش بیشتری داشته باشد.

۴- حافظه پهنایب کمتری در مقایسه با کامپایلر دارد، چرا که object code ای تولید نمی شود.

۴- کامپایلر نیازمند حافظه برای تولید object code است.

۵- مفسر برای اهداف فوری، آسب پذیرتر است.

۵- برای اهداف آمیخته، کامپایلر سودمندتر است. علت: تولید آبجکت کواد نسخه تفسیر آنجا در مقایسه با سورس کد

\* پیرو بند ۳ در جدول فوق، در ادامه انواع ظاهرا در یک برنامه را به طور مختصر شرح

می کنیم.



Bug → Error  
اشغال خطا

2019

فروردین ۹۸

Apr 14



۱۴۴۰

یکشنبه

۸ شعبان

- \* انواع خطاها در یک برنامه
- ① خطاهای نحوی (syntax errors)
  - ② خطاهای زمان اجرا (Runtime errors)
  - ③ خطاهای معنایی (semantic errors)

\* خطاهای نحوی: به خطاهای اطلاق می شود که مربوط به ساختار برنامه و قوانین مربوط به آن است.

مثلاً در زبان فلسفی دانش یک جمله با حرف بزرگ شروع می شود و با نقطه ختم می شود.

this sentence contains a syntax error.

آنها شروع با حرف بزرگ

هم در این خطا نحوی است.

② زبان های برنامه نویسی مختلف و کد های مختلف در میان خطاهای زمان اجرا

عبرآ بیان می دارد خطا در زمان اجرا  
خطا در خطای خط مربوط به آن  
راشتر نشان می دهد

\* خطاهای زمان اجرا: خطاهای هستند که از زمان اجرای مشاهده نمی شوند. به این نوع

خطاها اصطلاحاً استثنا (Exception) نیز اطلاق می شود و هر که

می تواند یک اتفاق بد را یک استثنا هستند.

مثلاً: تقسیم یک عدد صحیح بر صفر  $(n/m)$  که تعریف شده است.

صفر m در طول اجرا مقدار صفر را

به خود گرفته است.

روز بزرگداشت عطار نیشابوری



\* خطاهای معنایی: برنامه یا موفقیت اجرا می شود و کامپیوتر هم خطایی تولید نمی کند اما خروجی مورد نظر نیز حاصل نمی شود.

به عبارت دیگر: برنامه ای که شما نوشته اید، آن برنامه ایست که انتظار دارید نوشته باشید.

مثال:  $(a+b)*c$  ← انتظار شما  
 $a+b*c$  ← آنچه نوشته اید  
 $a*(b*c)$  ← آنچه برنامه با توجه به قوانین اجرای کند (اولت محکرها)

اهمیت رفع خطا (اشکال زدایی یا Debug): یکی از مهارت های مهم در حوزه

برنامه نویسی است که توأم با حوصله و جزئیات است

سبب بکارگذاشتن آن است که فرآیند را

دنبال می کند: پیدا کردن سرریزهای از خطا با حدس یا یقین

۱) تصحیح

۲) اجرای مجدد

۳) توانایی گام ها را بالا

برخی افراد، اشکال زدایی و برنامه نویسی را یکسان می دانند. مثال: کرنل لینوکس



2019

فروردین ۹۸

Apr 16

۲۷

۱۴۴۰

سه شنبه

۱۰ شعبان

## جمع شنبه - معرفی Loader, Linker

\* **Linker**: سورس کدها را لینک می‌کند تا یک فایل اجرایی باشد یا

تأثیر به دسترس به کارزار و مستقل باشند که بر مبنای وظایف  
طراحی و پیاده‌سازی شده‌اند.

وظیفه **Linker**: پیوند زدن کتابخانه‌ها و ماژول‌های

استفاده شده در برنامه با اصل برنامه را بر هم

دارد.

برنامه کابل شده + کتابخانه‌ها و ماژول‌های = کد قابل اجرا  
استفاده شده در برنامه

Executable code

