



فصل اول

مطالعه زبان های برنامه سازی



Directed by: Mahdi Sadeghizade
Email: Mahdi_sadeghe@yahoo.com
Site: www.Sadeghizadeh.ir

چرا زبانهای برنامه سازی را مطالعه می کنیم ؟

1. برای گسترش مفاهیم زبانی (شناسایی قابلیت های هر زبان)
2. برای گسترش فرهنگ لغت زبان برای ساختن اجزاء برنامه (ساختن اجزا مورد نیاز)
3. برای اینکه زبان بهتری را انتخاب کنیم (چه زبانی برای برطرف کردن نیاز ما مناسب است)
4. برای اینکه زبان جدید را ساده تر یاد بگیریم
5. برای اینکه یک زبان جدید طراحی کنیم

تاریخچه زبانهای برنامه سازی

عوامل مؤثر بر پیدایش و طراحی زبانها :

- *سخت افزار و سیستم عامل* دو جنبه ی مهم در پیدایش و طراحی زبانها بوده اند .
- *روشهای برنامه سازی* تغییر می کردند و باعث تغییر زبانها می شدند .
روش رویه ای ← روش ساختیافته (تابعی) ← روش
- *کاربردها* (آموزشی، علمی، تجاری، ...)
- *روشهای پیاده سازی* : سازگار با محیط پیاده سازی باشد .
- *مطالعات تئوریک* (مثل زبان α که پیاده سازی نشد اما اصول آن در پایگاه داده به کار می رود)
- *استاندارد سازی* : معیارهای استانداردسازی با توجه به شرایط زمانی و تکنولوژی تغییر می کند .

انواع استاندارد سازی :

خصوصی (مخصوص سازمان خاص) }
عمومی (همه سازمانها از آن حمایت می کنند) }

مسائل مهم در استاندارد سازی:

شرایط زمانی }
انطباق و توافق }
کهنگی (از رده خارج شدن) }

مدلهای زبانهای برنامه سازی :

• <i>زبانهای دستوری یا رویه ای (Imperative)</i>	
Statement 1; Statement 2; <input type="checkbox"/>	برنامه یک سری از دستورات پشت سر هم است که از بالا به پایین اجرا می شوند .
مانند : C, C++, Fortran, Algol, Cobol, PL/1, Ada, Smaltalk, Pascal	

• زبانهای کاربردی یا تابعی (Functional)
توابع در کنار هم برنامه را تشکیل می دهند و قابلیت فراخوانی تودرتو دارند . Function _n (...Function ₁ (data)...)
مانند : Schema , ML , Lisp

• مدل مبتنی بر قانون (Rule-Base)
برنامه ها به صورت مجموعه ای از قوانین پایه ای تجزیه مساله ، می باشند . ساختمان این زبان مشابه ساختار if است که در صورت رخداد شرایطی ، قانونی اجرا می شود . Enabling condition ₁ → action ₁ ≡ Action ₁ if enabling condition ₁
مانند : Prolog

• زبانهای شیء گرا (Object Oriented)
برنامه از تعدادی ماژول تشکیل شده که هر ماژول مشابه یک برنامه در زبان C می باشد . مبحث اصلی در این نوع زبان پشتیبانی از کلاس است .
مانند : C++, Java, Visoual

کاربردهای زبانهای برنامه سازی:

- **تجاری** : مانند طراحی پایگاه داده . (Cobol, C, PL/1)
 - **علمی** : مثل کارهای محاسباتی که فرمولهای زیادی دارند . (Basic, C++, C, Fortran)
 - **سیستمی** : یعنی چیزی شبیه به سیستم عامل بنویسیم . (C, C++, Assembly)
 - **هوش مصنوعی** : بازی های فکری مثل شطرنج . (Prolog, Lisp)
 - **انتشارات (Publishing)** : Tex, Postscript ، واژه پردازها .
 - **پردازشی (Process)** : کارهای پردازشی بالا ، یعنی بیشتر زمان روی CPU و سیستم فایل درگیر است . (Perl, TCL, Unix)
 - **آموزشی** : برای آشنایی با برنامه نویسی . (Basic, Pascal, C)
 - **New** : اغلب کاربردهای بالا را پشتیبانی می کنند و به محاوره ای نزدیک هستند . (ML, Smaltalk,)
- (Ifel

زمان ایجاد زبانهای برنامه سازی :

YEAR	LANGUAGE
1950- 55	Assembly.
1956- 60	Fortran, Algol 58, Algol 60 , Cobol , Lisb.
1961- 65	FortranIR, Cobol 61, Algol 61, Snobol, APL.
1966- 70	Fortran66, Cobol65, Algol68, Snobol 4, Simula 67, Basic, APL 360.
1971- 75	Pascal, Cobol74, APL(standard), C.
1976- 80	Ada, Fortran 77(standard).
1981- 85	Prolog.
1986- 90	C++,Fortran90.

ویژگی های یک زبان خوب :

به طور کلی عوامل زیر باعث می شود که یک زبان نسبت به زبانهای دیگر ارجحیت داشته باشد :

1. وضوح (Clarity)، سادگی (Simplicity)، واحدمند بودن (یکپارچگی) مفاهیم زبان (Unity) :
 زبان برنامه سازی باید چهار چوبی برای تفکر در مورد اینکه یک الگوریتم چگونه پیاده سازی می شود فراهم نماید . هر یک از اجزاء داخل این چهار چوب یک مفهوم را دربردارد . هر چه تعداد اعضاء کمتر باشد ، سادگی بیشتری خواهیم داشت . هر چه درک این مفاهیم روان تر باشد ، وضوح بیشتری داریم. هر چه ارتباط بین اجزاء زبان مشخص تر باشد و Abstraction (میزان تجرید) قوی تر باشد ، زبان برنامه سازی واحدمندتر است .

2. وضوح در Syntax برنامه :

هر چه وضوح بیشتری در متن برنامه و یا Syntax زبان باشد زمان برنامه سازی بهتر است .

3. طبیعی بودن با کاربرد یا تطبیق بیشتر زبان با کاربرد :

زبان باید بتواند ساختار منطقی الگوریتم ها را تحت کاربرد مشخصی منعکس نماید .

4. پشتیبانی از تجرید (Support for abstraction):

هر چه طبقه بندی اجزاء قوی تر انجام شود ، زبان برنامه سازی بهتر خواهد بود . مانند لایه بندی برنامه.

ایجاد Ada از Pascal و C++ از C به دلیل ایجاد قابلیت پشتیبانی از تجرید می باشد .
 اضافه شدن مفهوم Package به Pascal و مفهوم Class به C .

5. سادگی تغییر دادن برنامه (سهولت در بازبینی برنامه) :

هر چه زبان در امر بازبینی برنامه ، برای رفع خطاها و ایجاد تغییرات در برنامه ، سهولت بیشتری را ایجاد نماید آن زبان بهتر خواهد بود.

6. محیط برنامه سازی :

هر چه امکانات محیط برنامه سازی نظیر Compiler ، Linker و Debugger بیشتر باشد آن زبان بهتر است .

7. انتقال پذیری برنامه :

هر چه انتقال برنامه از سخت افزار یا سیستم عامل به سخت افزار یا سیستم عامل دیگری راحت تر انجام شود زبان برنامه سازی بهتر است.

8. هزینه استفاده از زبان :

الف) هزینه اجرا :

هزینه اجرای برنامه شامل استفاده از منابع کامپیوتر مخصوصاً CPU, Ram, Disk می باشد. مثلاً اجرای یک برنامه در زبان نباید باعث هدر رفتن وقت CPU شود یا به عبارت دیگر حداکثر کارایی از CPU گرفته شود .

ب) هزینه کامپایل :

هزینه ترجمه برنامه شامل استفاده از منابع کامپیوتر مخصوصاً CPU, Ram, Disk می باشد.

در زبانهای آموزشی مثل C, Pascal, Fortran ترجمه کارآمد از اجرای کارآمد مهم تر است زیرا در این زبانها برنامه مدام در حال کامپایل شدن می باشد .

ج) هزینه ایجاد ، تست و نگهداری : شامل استفاده از منابع کامپیوتر و نیروی انسانی .

د) هزینه پشتیبانی :

شامل تطبیق برنامه با شرایط محیطی، ایجاد امکانات جدید در نرم افزار و برطرف کردن اشکالات برنامه می باشد .

هر چه هزینه های استفاده از یک زبان برنامه سازی کمتر باشد آن زبان برنامه سازی بهتر خواهد بود.

9. امکان ترکیب ویژگی های زبان یا قابلیت تعامد :

خوبی این مزیت این است که اولاً یادگیری زبان را راحت تر می کند ، ثانیاً نوشتن برنامه به آن زبان ساده تر می باشد و عیب آن این است که ویژگی های با منطق غیره همسان وجود دارد ؛ یعنی بسته به محل کاربرد آن ویژگی منطق آن متفاوت خواهد بود . البته مشکلی برای کامپایل ایجاد نمی کند یعنی برنامه کامپایل می شود اما زمان آن بالا می باشد که باعث می شود کارایی اجرای برنامه پایین بیاید.

✓ بررسی ویژگی های بالا برای زبان Pascal :

+ موارد مثبت گزینه های : 1 ، 2 ، 5 (چون نحو ساده ای دارد)
- موارد منفی گزینه های : 3 (فقط آموزشی) ، 4 ، 6 (نسبت به زبانهای پیشرفته امروزی) ، 7 (فقط محیط آموزشی) ، 9 (زبان خشکی است)

✓ بررسی ویژگیهای بالا برای زبان C :

+ موارد مثبت گزینه های : 1 (نسبت به پاسکال وضوح و سادگی کمتری دارد اما از لحاظ یکپارچگی با هم معادلند . در کل بهتر از پاسکال است) ، 2 ، 3 (اکثر کاربردها را پوشش می دهد)
5 ، 7 (بسیاری از سیستم ها از آن حمایت می کنند) ، 8 و 9 (بهتر از پاسکال)
- موارد منفی گزینه های : 4 ، 6

✓ بررسی ویژگیهای بالا برای زبان C++ :

موارد مثبت گزینه های : 1.2.3.4.5.6.7.8.9

C++ فقط محیط خوبی ندارد که با توجه به پیشرفت های حالا خوب است .