

به نام خدا

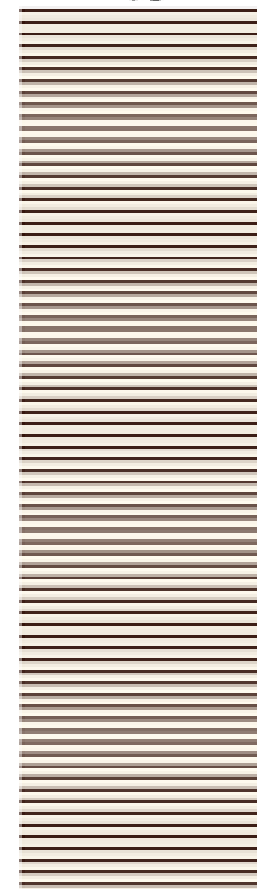


طراحی زبان‌های برنامه‌سازی

مرجان عظیمی
Azimi.marjan@gmail.com

مرجع

“Programming language: design and implementation”, T.W. Pratt and M.V. Zelkowitz



ارزشیابی



امتحان پایان ترم	۳۰٪
امتحان میان ترم	۳۰٪
تمرین	۱۰٪
امتحانک	۵٪
ارایه	۲۰٪
فعالیت کلاسی	۵٪

دلایل مطالعه زبان‌های برنامه‌سازی



- افزایش توانایی به منظور ایجاد و توسعه الگوریتم‌های کارا
- استفاده بهینه از زبان برنامه‌نویسی موجود
- انتخاب بهتر زبان برنامه‌نویسی
- یادگیری آسان‌تر زبان جدید
- ساده‌تر شدن طراحی زبان جدید

سیر تکاملی زبان‌ها



- زبان‌هایی با مبنای عددی: ALGOL, Fortran(F)
- زبان‌های تجاری: COBOL=(Common Business Oriented Language)
- زبان‌های هوش مصنوعی: SNOBOL, PROLOG, LISP(List Processing)
- زبان‌های سیستمی: C, Assembly

عوامل موثر در طراحی و پیاده‌سازی زبان‌های برنامه‌سازی



- سخت‌افزار و سیستم عامل (plat form)
 - روش‌های برنامه‌سازی (شی‌گرا، تابعی، دستوری)
 - کاربردها (صنعتی، تجاری، نظامی، ...)
 - روش‌های پیاده‌سازی (مثلا برای پیاده‌سازی آرایه از چه تکنیکی استفاده شود، مثلا از link list و یا فضای ثابت و پشت سر هم حافظه)
 - استانداردسازی
- ...
- محیط عملیاتی: محیطی که برنامه بر روی آن اجرا می‌شود.
 - محیط میزبان: محیطی که برنامه در آن ایجاد، تست و اشکال‌زدایی می‌شود.

تحول معماری‌های نرم‌افزار



• دوره کامپیوترهای بزرگ

- محیط دسته‌ای: یک برنامه به عنوان ورودی مجموعه‌ای از فایل‌های داده‌ها را دریافت می‌نماید و داده‌های آن‌ها را پردازش می‌کند و مجموعه‌ای از فایل‌های داده‌های خروجی را تولید می‌کند. مانند کارت‌های پانچ شده.

- محیط محاوره‌ای: یک برنامه مستقیماً با کاربر در ضمن اجرا از طریق صفحه نمایش ارتباط برقرار می‌کند.

✓ اثرات بر طراحی زبان: در دسته‌ای ورودی خروجی فایل مرسوم است ولی در محاوره‌ای ورودی و خروجی برخط است. در دسته‌ای هنگام بروز خطا کمکی از کاربر نمی‌توان گرفت. در دسته‌ای محدودیت زمانی وجود ندارد.

تحول معماری‌های نرم‌افزار



- دوره کامپیوترهای شخصی
 - کامپیوترهای شخصی
 - سیستم‌های تعبیه شده: سیستم کامپیوتری بخشی از یک سیستم بزرگتر است. در چنین سیستم‌هایی خرابی، قابل قبول نیست.
 - ✓ اثرات بر طراحی زبان: توسعه زبان‌های با گرافیک مناسب و رابط‌های کاربری مناسب. سیستم‌های تعبیه شده اغلب بدون سیستم‌عامل و ورودی و خروجی فایل هستند. اداره خطا در سیستم‌های تعبیه شده اهمیت بالایی دارد. این سیستم‌ها به صورت بلادرنگ کار می‌کنند.

تحول معماری‌های نرم‌افزار



• دوره شبکه

- سیستم‌های توزیع شده: استفاده از کامپیوترهای مرکزی در شرکت‌ها برای پردازش داده‌های مشترک، محاسبات کارگزار مشتری (مثال: سیستم رزرو هواپیما)

• اینترنت

- ✓ اثرات بر طراحی زبان: اطلاعات کارگزاران بزرگ در سطح جهانی، تعامل بین مشتری و کاربر، صفحات ایستا و پویای وب، امنیت صفحات، سرعت محدود، خطوط ارتباطی، قدرت پردازش سرور

دامنه کاربرد زبان‌ها



کاربردهای دهه ۱۹۶۰

- پردازش تجاری: COBOL (ورود سفارشات، کنترل فروش، مدیریت شخصی و ...)
- محاسبات علمی: FORTRAN (معادلات ریاضی)
- سیستمی: Assembly (سیستم عامل کامپیوتر، کنترل فرآیندهای بزرگ)
- هوش مصنوعی: LISP

دامنه کاربرد زبان‌ها



کاربردهای قرن ۲۱

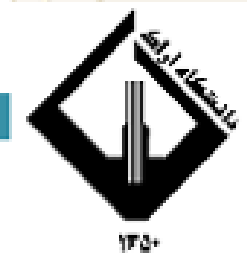
- تجاری: (C, C++, 4GL)
- علمی: (Java, C)
- سیستمی: (C, C++)
- هوش مصنوعی: (لیسپ و پرولوگ)
- انتشارات: (نرم افزارهای واژه پرداز)
- پردازش: اغلب از یک برنامه برای کنترل برنامه‌ی دیگر استفاده می‌شود.
مانند پاسخ خودکار به میل‌ها (پوسته یونیکس)
- کاربردهای جدید (مانند شی گراها...): مانند کاربرد ام‌ال در تحقیقات
زبان‌های برنامه‌سازی برای بررسی تئوری نوع

معیارهای یک زبان خوب



- وضوح، سادگی و یکپارچگی :
 - ✓ جامعیت مفهومی : داشتن حداقل تعداد مفاهیم مختلف به همراه قوانینی ساده برای ترکیب آنها
 - ✓ خوانایی برنامه : تفاوت‌های معنایی، منعکس کننده تفاوت‌های نحوی باشد.
- قابلیت تعامد : امکان ترکیب ویژگی‌های مختلف زبان
 - ✓ مثال : ترکیب عبارت و ساختار شرطی
 - ✓ مزیت : یادگیری ساده‌تر زبان و نوشتن راحت‌تر برنامه
 - ✓ معایب : کامپایل بدون خطا در ترکیب‌هایی که منطق روشن و اجرای کارآمدی ندارند.

معیارهای یک زبان خوب



- طبیعی بودن برای کاربردها
✓ زبان‌ها باید ساختمان داده، عملگرها، دستورات کنترلی و نحو مناسب برای مسئله ای که باید حل شود را داشته باشند.
- پشتیبانی از انتزاع: امکان ایجاد ساختارهای داده‌ای جدید
- سادگی بررسی درستی برنامه: این کار به دو صورت رسمی مانند روش‌های ریاضی و غیر رسمی مانند تست با ورودی‌های مختلف انجام می‌شود.

معیارهای یک زبان خوب



- محیط برنامه نویسی: وجود ویراستارهای خاص، امکانات نگهداری و اصلاح نسخه‌های متفاوت

- قابلیت حمل برنامه: امکان استفاده از برنامه‌های آن زبان بر روی ماشین‌های مختلف

- هزینه استفاده

هزینه اجرا: میزان استفاده از برنامه نظیر زمان و حافظه

هزینه ترجمه: زمان کامپایل برنامه

هزینه ایجاد و آزمایش: در زمان کوتاهی بتوان برنامه مورد نظر را ایجاد و تست کرد.

هزینه نگهداری: ساده بودن تغییرات احتمالی در آینده

نحو و معنا (Syntax & Semantic)



- نحو : ساختار دستوری مربوط به نحوه نوشتن آن‌ها را مشخص می‌کند.
- معنا: مفهومی است که به ساختارهای نحوی زبان داده می‌شود.

مدل‌های زبانی



• زبانهای دستوری یا رویه‌ای (imperative)

- در این زبان‌ها برنامه، شامل دنباله‌ای از دستورات است و اجرای هر دستور باعث می‌شود تا مترجم مقدار ثبات‌های **cpu** یا محلی از حافظه را تغییر دهد. در این زبان‌ها باید به طور دقیق چگونگی تبدیل ورودی‌ها به خروجی‌ها مشخص شود.

- مانند C++ , C و پاسکال و ...

Statement1;

Statement2;

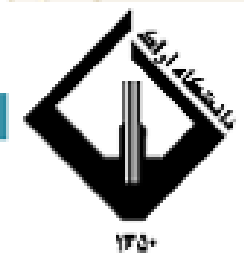
.

.

.

Statement;

مدل‌های زبانی



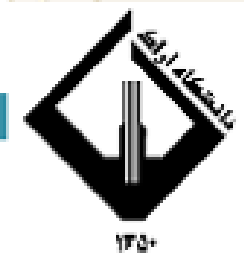
- زبانهای تابعی یا کاربردی (applicative)

- توجه به عملی که باید انجام شود بدون توجه به چگونگی انجام آن عمل.
- مانند ام ال و لیسپ

`functionn(...(function2(function1(data)) ...)`

▪

مدل‌های زبانی



• زبان‌های قانونمند (rule-based)

- شرایطی را بررسی می‌کنند و در صورت برقرار بودن آنها فعالیت را انجام می‌دهند.

- مانند PRPLOG

Enabling condition₁ → action₁

Enabling condition₂ → action₂

•

•

•

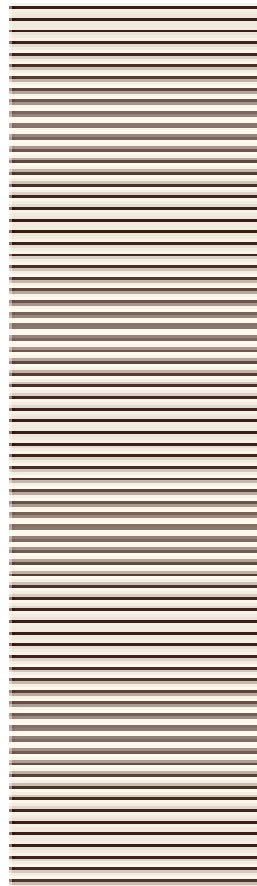
Enabling condition_n → action_n

مدل‌های زبانی



۱۳۵۰

- برنامه‌نویسی شی گرا (object-oriented):
 - اولویت داشتن داده‌ها نسبت به عملیات مورد نیاز بر روی آن‌ها. در این زبان‌ها اشیا مورد نیاز در برنامه ساخته شده به گونه‌ای که عملیات مورد نیاز بر روی آن‌ها در درون هر شی گنجانده شده است.
 - مثال: java, c++
 - زبانی مانند c++ ترکیبی از یک زبان تابعی، دستوری و شی‌گرا محسوب می‌شود.



۱۱