زبان‌های برنامه‌سازی

دکتر حبیب ایزدخواه

مهندس عمران یونسی مهندس عزیز حنیفی

امروزه کتاب‌خوانی و علم‌آموزی، نه تنها یک وظیفه‌ی ملی، که یک واجب دینی است.

مقام معظم رهبری

در عصر حاضر یکی از شاخصه‌های ارزیابی رشد، توسعه و پیشرفت فرهنگی هر کشوری میزان تولید کتاب، مطالعه و کتاب‌خوانی مردم آن مرز و بوم است. ایران اسلامی نیز از دیرباز تاکنون با داشتن تمدنی چندهزارساله و مراکز متعدد علمی، فرهنگی، کتابخانه‌های معتبر، علما و دانشمندان بزرگ با آثار ارزشمند تاریخی، سرآمد دولت‌ها و ملت‌های دیگر بوده و در عرصه‌ی فرهنگ و تمدن جهانی به‌سان خورشیدی تابناک همچنان می‌درخشد و با فرزندان نیک‌نهاد خویش هنرنمایی می‌کند. چه کسی است که در دنیا با دانشمندان فرزانه و نام‌آور ایرانی همچون ابوعلی سینا، ابوریحان بیرونی، فارابی، خوارزمی و ... همچنین شاعران برجسته‌ای نظیر فردوسی، سعدی، مولوی، حافظ و ... آشنا نباشد و در مقابل عظمت آنها سر تعظیم فرود نیاورد. تمامی این افتخارات ارزشمند، برگرفته از میزان عشق و علاقه فراوان ملت ما به فراگیری علم و دانش از طریق خواندن و مطالعه منابع و کتاب‌های گوناگون است. به شکرانه‌ی الهی، تاریخ و گذشته ما، همیشه درخشان و پربار است. ولی اکنون در این زمینه در چه جایگاهی قرار داریم؟ آمار و ارقام ارائه‌شده از سوی مجامع و سازمان‌های فرهنگی در مورد سرانه‌ی مطالعه‌ی هر ایرانی، برایمان چندان امیدوارکننده نمی‌باشد و رهبر معظم انقلاب اسلامی نیز از این وضعیت بارها اظهار گله و ناخشنودی نموده‌اند.

کتاب، دروازه‌ای به سوی گستره‌ی دانش و معرفت است و کتاب خوب، یکی از بهترین ابزارهای کمال بشری است. همه‌ی دستاوردهای بشر در سراسر عمر جهان، تا آنجا که قابل کتابت بوده است، در میان دست‌نوشته‌هایی است که انسان‌ها پدید آورده و می‌آورند. در این مجموعه‌ی بی‌نظیر، تعالیم الهی، درس‌های پیامبران به بشر، و همچنین علوم مختلفی است که سعادت بشر بدون آگاهی از آنها امکان‌پذیر نیست. کسی که با دنیای زیبا و زندگی‌بخش کتاب ارتباط ندارد بی‌شک از مهم‌ترین دستاورد انسانی و نیز از بیشترین معارف الهی و بشری محروم است. با این دیدگاه، به‌روشنی می‌توان ارزش و مفهوم رمزی عمیق در این حقیقت تاریخی را دریافت که اولین خطاب خداوند متعال به پیامبر گرامی اسلام(ص) این است که «بخوان!» و در اولین

سوره‌ای که بر آن فرستاده‌ی عظیم‌الشأن خداوند، فرود آمده، نام «قلم» به تجلیل یاد شده‌است: «إِقْرَأْ وَ رَبُّکَ الْاَکْرَمُ. اَلَّذی عَلَّمَ بِالْقَلَم» در اهمیت عنصر کتاب برای تکامل جامعه‌ی انسانی، همین بس که تمامی ادیان آسمانی و رجال بزرگ تاریخ بشری، از طریق کتاب جاودانه مانده‌اند.

دانشگاه پیام‌نور با گستره‌ی جغرافیایی ایران‌شمول خود با هدف آموزش برای همه، همه‌جا و همه‌وقت، به‌عنوان دانشگاهی کتاب‌محور در نظام آموزش عالی کشورمان، افتخار دارد جایگاه اندیشه‌سازی و خردورزی بخش عظیمی از جوانان جویای علم این مرز و بوم باشد. تلاش فراوانی در ایام طولانی فعالیت این دانشگاه انجام پذیرفته تا با بهره‌گیری از تجربه‌های گرانقدر استادان و صاحب‌نظران برجسته کشورمان، کتاب‌ها و منابع آموزشی درسی شاخص و خودآموز تولید شود. در آینده هم، این مهم با هدف ارتقای سطح علمی، روزآمدی و توجه بیشتر به نیازهای مخاطبان دانشگاه پیام‌نور با جدیت ادامه خواهد داشت. به‌طور قطع استفاده از نظرات استادان، صاحب‌نظران و دانشجویان محترم، ما را در انجام این وظیفه‌ی مهم و خطیر یاری‌رسان خواهد بود. پیشاپیش از تمامی عزیزانی که با نقد، تصحیح و پیشنهادهای خود ما را در انجام این وظیفه‌ی خطیر یاری می‌رسانند، سپاسگزاری می‌نماییم. لازم است از تمامی اندیشمندانی که تاکنون دانشگاه پیام‌نور را منزلگه اندیشه‌سازی خود دانسته و ما را در تولید کتاب و محتوای آموزشی درسی یاری نموده‌اند، صمیمانه قدردانی گردد. موفقیت و بهروزی تمامی دانشجویان و دانش‌پژوهان عزیز آرزوی همیشگی ما است.

دانشگاه پیام‌نور

فهرست مطالب کتاب

پیشگفتار

امروزه کمابیش استفاده از هر کاربرد در سایه امکانات سیستم کامپیوتری میسر است. یک سیستم کامپیوتری از مجموعه سخت‌افزار و نرم‌افزار تشکیل شده‌است. هر نرم‌افزار و برنامه کاربردی باید به یک‌زبان برنامه‌سازی نوشته شود تا قابلِ‌تبدیل به زبان ماشین جهت اجرا روی سخت‌افزار باشد. بنابراین، یک‌زبان برنامه‌سازی وسیله ارتباطی بین یک برنامه‌نویس و سیستم کامپیوتری است. باتوجه به این مورد اهمیت زبان‌های برنامه‌سازی مشخص می‌شود. هدف از این کتاب بررسی روش‌های طراحی و پیاده‌سازی زبان‌های برنامه‌سازی است. در این کتاب علاوه‌بر بررسی روش‌های پیاده‌سازی زبان‌های برنامه‌سازی، نحوه به‌کاربردن این روش‌ها در تعدادی از زبان‌های معروف را موردبررسی قرار خواهیم داد. این کتاب به‌صورت زیرسازماندهی شده‌است:

در فصل اول به بیان اصول طراحی زبان‌های برنامه‌سازی پرداخته شده‌است.

در فصل دوم اثرات معماری ماشین بر زبان‌های برنامه‌سازی موردبررسی قرار گرفته است.

در فصل سوم گسترش و روند تکاملی زبان‌های برنامه‌سازی آورده شده‌است.

در فصل چهارم نحو و معنای رسمی زبان مورد بررسی قرار گرفته است.

در فصل پنجم انواع داده اولیه که در یک‌زبان برنامه‌سازی می‌تواند وجود داشته شده‌است مورد بررسی قرار گرفته است.

در فصل ششم انواع داده‌های ساخت‌یافته مورد بررسی قرار گرفته است.

در فصل هفتم روش‌های تعیین ترتیب اجرای دستورها و روش‌های پیاده‌سازی آن‌ها موردبررسی قرار گرفته است.

در فصل هشتم روش‌های کنترل زیربرنامه‌ها موردبررسی قرار گرفته است.

در فصل نهم برخی مباحث پیشرفته در طراحی زبان‌های برنامه‌سازی شیءگرا موردبررسی قرار گرفته است.

آنچه که غیرقابل‌انکار است تأثیر مستقیم کتاب معروف و منبع درسی بسیاری از دانشگاه‌های جهان تألیف آقای Terrence W. Pratt در تدوین کتاب حاضر است. همچنین از مطالب و مثال‌های ارائه‌شده در کتاب تألیفی آقای John C. Mitchell و همچنین کتاب علیرضا خلیلیان استفاده فراوانی کرده‌ایم.

فصل اول

اصول طراحی زبان‌های برنامه‌سازی

هدف کلی

با دلایل مطالعه زبان‌های برنامه‌نویسی و صفات یک‌زبان خوب آشنا خواهید شد.

هدف‌های یادگیری

در پایان فصل، دانشجو با مفاهیم زیر آشنا خواهد شد:

۱. تفاوت زبان طبیعی با زبان برنامه‌سازی را بیان کند.
۲. هدف‌ها و دلایل مطالعه زبان‌های برنامه‌سازی را نام ببرد.
۳. صفات یک‌زبان خوب را بیان کند.
۴. انواع روش‌های برنامه‌نویسی را توضیح دهد.
۵. مدل‌های زبان را تشریح کند.
۶. گسترش و روند تکاملی زبان‌های برنامه‌سازی را توضیح دهد.
۷. استانداردسازی زبان را توضیح دهد.

مقدمه

زبان یک سیستم ارتباطی و مجموعه‌ای از نشانه‌های قراردادی معین (یعنی نحو یا دستور زبان) است که برای انتقال پیام استفاده می‌شود. اگر این سیستم ارتباطی بتواند

بین انسان‌ها ارتباط برقرار کند به آن زبان طبیعی[1] گفته می‌شود و چنانچه این سیستم جهت ارتباط بین انسان و ماشین استفاده شود به آن زبان مصنوعی[2] می‌گویند. در زبان‌شناسی، دستور زبان یا نحو مجموعه‌ای از قوانین ساختمانی در زبان است که ساختِ جمله‌ها، عبارت‌ها، و کلمه‌ها را در هرزبانی معیّن می‌کند.

هر زبان برنامه‌سازی[3] یک‌زبان مصنوعی است که همانند زبان‌های طبیعی شامل کاراکترها، نمادهای خاص و قواعد است، به این قواعد که برای ایجاد دستورالعمل‌ها یا دستورها به‌کار گرفته می‌شوند نحو[4] یا دستور زبان گفته می‌شود. زبان‌های برنامه‌نویسی با زبان‌های طبیعی تفاوت دارند و آن اینکه زبان‌های طبیعی فقط برای فعل‌وانفعالات بین انسان‌ها به‌کار می‌روند، درحالی‌که زبان‌های برنامه‌نویسی به انسان‌ها اجازه می‌دهد که ازطریق دستورات با ماشین‌ها ارتباط برقرار کنند. تعداد زیادی زبان‌های مختلف برنامه‌سازی وجود دارند که بعضی از آن‌ها جهت انجام مقاصد خاصی نظیر کنترل ربات‌های ماشینی ابداع شده‌اند و بعضی دیگر قابل انعطاف‌تر بوده و به‌عنوان سیستم ارتباطی همه منظوره جهت بسیاری از کاربردها مناسب هستند.

زبان برنامه‌سازی: هرگونه علامت‌گذاری جهت توصیف الگوریتم‌ها و ساختمان داده‌ها

زبان برنامه‌نویسی یا زبان کامپیوتری یک تکنیک ارتباطی استاندارد برای بیان دستورالعمل‌ها به یک رایانه است. یک‌زبان، به برنامه‌نویس اجازه می‌دهد که با دقت مشخص کند که رایانه روی چه داده‌ای عمل کند، این داده چگونه ذخیره یا منتقل شود، و تحت شرایط مختلف کدام الگوریتم روی آن اعمال شود.

۱-۱ اهداف مطالعه زبان‌های برنامه‌سازی

اهداف و دلایل متعددی برای بررسی زبان‌های برنامه‌سازی می‌تواند وجود داشته باشد. بنابراین در ابتدا لازم است مشخص شود که چرا ما به مطالعه زبان‌های برنامه‌سازی نیاز داریم. هدف ما در این کتاب، آشنایی دانشجویان با روش‌های برنامه‌سازی، مفاهیم و قابلیت‌های زبان‌های برنامه‌سازی مختلف است. به‌طورخلاصه یادگیری مفاهیم و قابلیت‌های زبان‌های برنامه‌سازی مختلف به دانشجویان این امکان را می‌دهد که: با

1. Natural Language
2. Artificial Language
3. Programming Language
4. Syntax

ویژگی‌های زبان‌های مختلف جهت ایجاد برنامه‌های کارا آشنا شوند، بتوانند زبان‌های جدید را راحت‌تر یاد گیرند و همچنین بتوانند از قابلیت‌های خوب آن‌ها استفاده کنند و یا بتوانند بهترین زبان ممکن را برای نوشتن یک برنامه خاص انتخاب کنند. این موارد و سایر موارد در ادامه توضیح‌داده شده‌است (Terrence W. Pratt, 2000).

الف) افزایش توانایی به‌منظور توسعه الگوریتم‌های کارآمد: زبان‌ها ویژگی‌هایی دارند که اگر به خوبی مورداستفاده قرار گیرند موجب افزایش سرعت اجرای برنامه می‌شوند و همچنین ازنظر راحتی پیاده‌سازی به‌نفع برنامه‌نویس است، ولی اگر به‌طور نامناسب مورداستفاده قرار گیرند وقت زیادی از برنامه‌نویس را می‌گیرند. برفرض‌مثال داخل برنامه‌ای به ساختمان داده‌ای خاص (مانند: لیست پیوندی[1] و ...) نیاز داریم؛ اگر بدانیم زبان پیاده‌سازی ما از این ساختمان داده[2] خاص پشتیبانی می‌کند و چگونه، دراین‌صورت پیاده‌سازی برنامه راحت‌تر خواهد بود. به‌عنوان‌مثال دیگر، اگر برنامه‌نویس از تکنیک بازگشتی[3]، مزایا و مشکلات آن اطلاع داشته باشد می‌تواند تصمیم بگیرد چه موقع از آن‌ها استفاده کند یا نکند.

با مطالعه زبان‌های برنامه‌سازی مختلف و آگاهی از ویژگی‌های آن‌ها با محدودیت‌های کمی ازنظر تفکری و روش‌های پیاده‌سازی در تولید نرم‌افزار خواهیم داشت. یعنی برنامه‌نویسان با آگاهی از ساختارهای برنامه‌نویسی گوناگون، ازطریق مطالعه زبان‌های برنامه‌سازی مختلف، می‌توانند با ذهن بازتری به فرایند برنامه‌نویسی بپردازند.

نکته مهم این است که در اغلب موارد می‌توان ساختارهای موجود در یک‌زبان را به‌کمک سایر ویژگی‌های زبان دیگر که فاقد آن ساختارها است شبیه‌سازی کرد. برای مثال، در زبان C آرایه‌های انجمنی[4] که در زبان Perl وجود دارد، به‌طورمستقیم وجود ندارد. اما برنامه‌نویسی که از وجود چنین ویژگی آشنا است می‌تواند به‌کمک سایر ساختارهای زبان C آن را شبیه‌سازی‌کرده و از مزایای آن استفاده کند.

---

1. Linked List
2. Data Structure
3. Recursive
4. Associative Arrays

پرسش تحقیق	الف) درمورد نخ‌ها[1] و برنامه‌نویسی چندنخی[2] در زبان‌های برنامه‌سازی تحقیق کنید. ب) زبان‌هایی مثل Java یا C# امکان استفاده مستقیم از برنامه‌نویسی چندنخی را می‌دهند، اما زبان C این چنین امکانی را نمی‌دهد. تحقیق کنید که چگونه می‌تواند چندنخی را در زبان C شبیه‌سازی کرد و از مزایای آن استفاده کرد.
پرسش تحقیق	در زبان‌های شیءگرا مفهومی به نام کلاس وجود دارد که در زبان‌های ساخت‌یافته مانند C این مفهوم وجود ندارد. به‌نظر شما چگونه می‌توان مفهوم کلاس را در زبان C شبیه‌سازی کرد. آیا تمام مزایایی که کلاس در زبان‌های شیءگرا دارد کلاس شبیه‌سازی‌شده شما هم می‌تواند داشته باشد.

ب) استفاده بهینه از زبان‌های برنامه‌سازی موجود: با درک اینکه چگونه ویژگی‌های یک‌زبان پیاده‌سازی می‌شوند توانایی شما در نوشتن برنامه‌های کارآمد افزایش می‌یابد. به‌عنوان‌مثال، دربرخی از زبان‌ها، رشته‌ها را هم می‌توان با استفاده از آرایه‌ها پیاده‌سازی کرد و هم به‌عنوان نوع داده مستقل وجود دارد که می‌توان استفاده کرد. اگر بدانید آرایه‌ها و رشته‌ها در زبان برنامه‌نویسی موردنظرتان چگونه ایجاد و پیاده‌سازی می‌شوند می‌توانید از زبان برنامه‌نویسی به‌طوربهینه استفاده کنید.

ج) امکان یادگیری آسان‌تر زبان جدید: با مطالعه زبان‌های برنامه‌نویسی متفاوت، امکان یادگیری زبان جدید ساده‌تر خواهد شد. به‌عنوان‌مثال در زبان Pascal دستور شرط به‌صورت if (condition) then نوشته می‌شود در بیشتر زبان‌ها نیز دستور شرط با if نشان‌داده می‌شود. پس اگر ما در یک‌زبان مفهوم شرط را بدانیم در یادگیری زبان جدید با آن مفهوم مشکلی نخواهیم داشت. به‌عنوان‌مثال دیگر، اگر منطق شیءگرایی را در یک‌زبان یاد بگیریم یادگیری سایر زبان‌های شیءگرا راحت‌تر می‌شود.

د) ساده‌ترشدن طراحی زبان جدید: بعضی مواقع به‌همراه با ساختن خود برنامه، نوع و عملیات مربوطه را هم تعریف می‌کنیم و به‌گونه‌ای کار می‌کنیم مثل اینکه درحال طراحی یک‌زبان جدید هستیم. برفرض‌مثال کاربر ممکن است یک‌نوع جدید به نام عدد مختلط[3] تعریف کند و عملیات ضرب و جمع و تفریق و غیره را برای آن درنظر بگیرد. دراین‌حالت برنامه‌نویس مثل طراح زبان جنبه‌های جدیدی به زبان اضافه می‌کند. با آشنایی با گستره‌ای از ساختارها و روش‌های پیاده‌سازی زبان‌های برنامه‌نویسی موجود

1. Thread
2. Multi Threading
3. Complex Number

ما می‌توانیم راحت‌تر دست به ایجاد واسط‌های کاربری[1] بزرگ بزنیم؛ برنامه‌هایی مانند ویرایش‌گرهای متن[2]، بسته‌های گرافیکی و سایر برنامه‌های بزرگ که یکی از ویژگی‌های آن‌ها محاوره با کاربر است.

و) آشنایی با برخی ساختارهای مفید برنامه‌نویسی: اگر برنامه‌نویس فقط با یک‌زبان آشنایی داشته باشد در جستجو برای حل مسئله، فقط به توانایی‌های زبانی فکر می‌کند که با آن آشنا است و این یک محدودیت است. با مطالعه زبان‌های برنامه‌سازی متعدد، دامنه لغات یک برنامه‌نویس افزایش می‌یابد. به‌عنوان‌مثال یک ساختار کنترلی به نام همروند[3] در خیلی از برنامه‌ها مفید است ولی زبان‌های محدودی این قابلیت را پشتیبانی می‌کنند. به‌عنوان‌مثال دیگر، بعضی ساختارها مانند توابع بازگشتی در همهٔ زبان‌ها وجود ندارد. آشنایی با زبان‌های برنامه‌نویسی مختلف ما را با ساختارهای مختلف آشنا می‌کند چون کسی که فقط به یک‌زبان آشنایی داشته باشد ذهنش محدود به‌همان زبان خواهد بود.

پرسش تحقیق الگوریتم‌های بازگشتی به‌طورمعمول بسیار کندتر از الگوریتم‌های تکراری معادل خود کار می‌کنند ولی ازنظر برنامه‌نویسی راحت‌تر از معادل تکراری خود هستند. مثال هایی بزنید که کجا بهتر است از الگوریتم بازگشتی و کجا از معادل تکراری استفاده شود.

و) انتخاب بهترین زبان برنامه‌سازی: زبان‌های مختلف برای کارهای متفاوتی طراحی شده‌اند پس آشنایی با زبان‌های برنامه‌نویسی مختلف باعث انتخاب بهترین زبان برای کاربرد موردنظر ما می‌شود. آگاهی از زبان‌های مختلف باعث می‌شود که برای مسئله خاص بتوان زبان بهتری را انتخاب کرد به‌عنوان‌مثال:

– کاربردهایی که نیاز به محاسبات عددی دارند بهتر است از Fortran استفاده شود.
– برای کاربردهای هوش مصنوعی از Lisp یا Prolog و جهت کاربردهای اینترنت از Perl یا Java مناسب هستند.
– برای کاربردهایی که نیاز به تصمیم‌گیری است زبان‌هایی مثل ML،Lisp یا Prolog مناسب‌تر هستند.

پرسش تحقیق فرض کنید که هدف ما طراحی یک نرم‌افزاری است که یک چراغ راهنمایی واقع در تقاطع یک خیابان را به‌طور خودکار کنترل کند. تحقیق کنید کدام‌یک از زبان‌های برنامه‌سازی موجود برای این‌کار مناسب‌تر هستند و چرا؟

1. User Interface
2. Text Processor
3. Concurrent

۱-۲ روند توسعه زبان‌های برنامه‌نویسی

می‌توان روند توسعه نحوه برنامه‌نویسی را از ابتدا تاحال‌حاضر به پنج نسل مختلف تقسیم‌بندی کرد که عبارت‌اند از:

نسل اول: در دهه ۱۹۵۰ برنامه‌نویسی کامپیوترهای اولیه به‌طورخاص ÜNIVAC 1 و IBM 701 توسط تغییر سیم‌ها و تنظیم هزاران کلید و سویچ انجام می‌شد. دربرخی موارد این تنظیمات برروی کاغذهای طومارگونه و یا کارت‌های سوراخ‌شده نوشته می‌شدند که به کامپیوتر می‌گفتند چه کاری را به چه صورت و در چه زمانی انجام دهد. به‌منظور اجرای یک نرم‌افزار، برنامه‌نویس باید اطلاعات جامع و کاملی از کامپیوتر موردنظر می‌داشت. یک اشتباه کوچک منجر به شکست در کل برنامه کامپیوتری می‌شد.

نسل دوم: در این نسل هدف رهایی از پیچیدگی‌های زبان ماشین برای برنامه‌نویسی بود. نتیجه این تلاش‌ها تولد نسل دوم زبان‌های برنامه‌نویسی یعنی زبان اسمبلی در اواسط دهه ۱۹۵۰ شد. در این نسل از نمادها[1] به‌جای دستورات ماشین استفاده می‌شد.

نسل سوم: هدف در زبان‌های برنامه‌سازی نسل سوم، این است که قابلیّت استفاده از زبان‌ها ساده‌تر شوند؛ یعنی به‌کارگیری آن راحت‌تر باشد. برای این منظور، در اواخر دهه ۱۹۵۰ مفسرها[2] و کامپایلرها پا به‌عرصه ظهور گذاشتند. شروع این نسل با ارائه زبان برنامه‌نویسی FORTRAN در سال ۱۹۵۳ توسط IBM بود. به‌طبع آن در سال ۱۹۵۹ زبان برنامه‌نویسی COBOL به‌منظور استفاده در دنیای نرم‌افزارهای تجاری عرضه شد. زبان‌های سطح بالای برنامه‌نویسی مانند BASIC، PASCAL، ALGOL، PL/I و C در این دوره معرفی شدند. بیشتر زبان‌های امروزی (C++، C# و جاوا)، نیز زبان‌های نسل سوم هستند. اغلب زبان‌های نسل سوم (3GL)، برنامه‌نویسی ساخت‌یافته را پشتیبانی می‌کنند.

نسل چهارم: در ادامه هدف زبان‌های نسل سوم، هدف این نسل از زبان‌های برنامه‌سازی نیز حرکت به‌سمت برنامه‌نویسی آسان و رفتن به‌سطح انتزاع بالاتری نسبت به زبان‌های نسل سوم بود. زبان‌های این نسل برنامه‌نویس را قادر می‌سازند تا کارهای

---

1. Symbols
2. Interpreters

سطح بالاتر و بیشتری را توسط کد کمتری انجام دهد. هر دستور از زبان‌های این نسل معادل صدها دستور از زبان‌های نسل سوم است. طرح‌های مربوط به زبان‌های نسل چهارم و پنجم، بیشتر در حلّ مسائل و مهندسی سیستم‌ها تمرکز یافته‌اند. امروزه گروهی از تولیدکنندگان نرم افزار، ابزارهای ایجاد کاربردهای گوناگونی را تولید و معرفی می‌کنند که به آن‌ها زبان‌های نسل چهارم[1] (4GL) گفته می‌شود. یک‌نرم‌افزار به زبان نسل چهارم، با نرم‌افزار سیستم مدیریت پایگاه داده‌ها[2] جهت ذخیره، پردازش و بازیابی داده‌های موردنیاز جهت نیازمندی‌های کاربر به‌صورت محاوره‌ای عمل می‌کند. یک‌زبان نسل سوم یک‌زبان رویه‌ای است، یعنی زبانی است که در استفاده از آن برنامه‌نویس باید مراحل رویه‌های پردازشی لازم جهت به‌دست‌آوردن نتیجه دلخواه را برای کامپیوتر مشخص کند. ولی یک زبان نسل چهارم زبان غیررویه‌ای است که به کاربر اجازه می دهد تنها مشخص کند خروجی چگونه باشد، بدون اینکه نیاز به‌شرح کلیه جزئیات جهت پردازش و تولید نتیجه باشد.

نسل پنجم: این نسل از زبان‌های برنامه‌نویسی هنوز در مرحله تئوری هستند و تا به امروز نمونه عملی از آن‌ها ساخته نشده‌است. بسیاری از تلاش‌ها به‌دلیل محدودیت سخت افزارها، عملیاتی نشده اند. استفاده از زبان طبیعی و روزمره برای تفهیم کارها به کامپیوترها از ویژگی‌های بارز این نسل به‌شمار می‌رود. استفاده از شبکه‌های عصبی[3] و هوش مصنوعی و همچنین استفاده از عامل‌ها[4] به‌منظور انجام کارها در کامپیوتر از دیگر ویژگی‌های این نسل از زبان‌ها هستند.

۱-۳ مدل‌های زبان (الگوهای مختلف برنامه‌نویسی یا مدل‌های محاسباتی زبان)

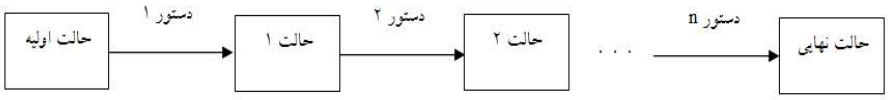
برای بررسی ساختار زبان‌ها، سه مدل محاسباتی معمول وجود دارد که هریک به‌روش خاصی اشاره می‌کنند که بعد از تعریف حالت یک برنامه به بررسی آن‌ها می‌پردازیم.

حالت یک برنامه: به وضعیت فعلی و رفتار برنامه در هرلحظه از اجرای برنامه، حالت یک برنامه گفته می‌شود. به‌طور خاص، حالت اولیه نشان‌دهندهٔ لحظه شروع

1. Fourth Generation Language
2. Data Base Management System
3. Neural Networks
4. Agents

برنامه و حالت نهایی نشان دهنده لحظه خاتمه اجرای برنامه است. با درنظرگرفتن تعریف حالت یک برنامه، انواع مدل‌های یک‌زبان به‌صورت زیر است:

۱. زبان‌های دستوری[1]: در این مدل از زبان‌ها با اجرای هر دستور حالت کامپیوتر تغییر می‌کند.



در این مدل، برنامه متشکل از مجموعه‌ای دستورات است که به‌صورت پشت سرهم اجراشده و اجرای هر دستور باعث تغییر در یک یا چندخانه از حافظه می‌شود. به‌طورکلی این مدل از سخت‌افزار پیروی می‌کند به‌عنوان‌مثال نمونه‌هایی از این مدل می‌توان به موارد زیر اشاره کرد.

Fortran و C, Pascal, Ada, COBOL, Algol

۲. زبان‌های شیءگرا[2]: در این مدل به هر موجودیت ممکن (که به شیء[3] معروف است) به‌صورت‌مستقل نگریسته می‌شود و هر شیء دارای یک‌سری خصوصیات[4] داده‌ای و دارای تعدادی متد[5] برای دستکاری آن‌ها است. در این مدل ابتدا اعمال ساده روی اشیای تعریف‌شده و سپس اعمال و اشیای پیچیده‌تر با بهره‌گیری از امکانات ارث‌بری و سایر امکانات موجود ایجاد می‌شوند. اشیا داده‌ای پیچیده از اشیای داده‌ای ساده خود زبان ساخته می‌شوند و متدهایی طراحی می‌شوند تا برروی آن‌ها اعمال شوند. این مدل با ساختن اشیا دقیق کارایی زبان‌های دستوری را به‌دست‌آورده و با ساختن دسته‌ای از توابع قابلیت انعطاف و اعتماد برنامه‌نویسی تابعی را کسب می‌کنند.

۳. زبان‌های اعلانی[6]: در این مدل‌ها منطق و محاسبات بدون شرح چگونگی انجام آن‌ها بیان می‌شود. در این مدل‌ها تلاش می‌شود که با توصیف عملیات موردنیاز به‌جای توضیح چگونگی انجام عملیات، اثرجانبی برنامه‌ها را کاهش و یا به کل از میان برداشت. این مدل‌ها ازلحاظ اولویت‌های عملیاتی در تناقض کامل با شیوه

---

1. Imperative Programming Languages
2. Object-oriented Programming Languages
3. Object
4. Properties
5. Methods
6. Declarative Programming Languages

برنامه‌نویسی دستوری است. ازآنجاکه این زبان می‌تواند تا حد چشمگیری نوشتن برنامه های موازی برای رایانش موازی را آسان و ساده کند توانسته توجه زیادی را به خود معطوف سازد.

الف) زبان‌های تابعی[1]: در این مدل به‌جای اینکه تغییر حالت‌های کامپیوتر به‌ازای دستورات را درنظر بگیریم به اعمال کلی که ما را از وضعیت شروع به‌وضعیت پایان می‌رساند توجه می‌کنیم.

function n (… function 2 (function 1 (data)) …)

در این مدل به‌جای مشاهده تغییر حالت، عملکرد برنامه دنبال می‌شود. در اینجا این سؤال باید پاسخ داده شود که چه عملی باید برروی حالت اولیه ماشین انجام شود تا با دستیابی به مجموعه اولیه متغیرها و ترکیب آن‌ها پاسخ مناسب داده شود وتوسعه برنامه با ایجاد توابعی از توابع ایجادشده قبلی صورت می‌گیرد. زبان‌های زیر از نمونه زبان‌های تابعی هستند که از این مدل پشتیبانی می‌کنند.

Lisp, APL و ML

ب) زبان‌های مبتنی بر قاعده[2]: شرایطی را بررسی می‌کنند و درصورت برقراربودن آن‌ها فعالیتی را انجام می‌دهند. مانند مدل دستوری است اما دستورات به‌صورت پشت‌سرهم اجرا نمی‌شوند. در این مدل اجرای اعمال مختلف درصورت پدیدآمدن شرطی خاص صورت می‌گیرد. نحو چنین زبان‌هایی به‌صورت زیر است:

Enabling condition ⟶ action 1
Enabling condition ⟶ action 2
.
.
.
Enabling condition ⟶ action n

Prolog متداول‌ترین زبان مبتنی بر قاعده است.

در اینجا این موضوع را خاطر نشان می‌شویم که چگونگی استفاده از زبان به برنامه‌نویس بستگی دارد و برنامه‌نویس می‌تواند از مدل مبتنی بر قاعده به‌گونه‌ای استفاده کند که دستورات به‌صورت پشت‌سرهم اجرا شوند.

---

1. Applicative Programming Languages
2. Rule-based Programming Languages