

دانش اگر در ثریا هم باشد مردمانی از سرزمین پارس بر آن دست خواهند یافت.

پیامبر اعظم (ص)

شرح جامع و نمونه سوالات

دروس تخصصی نرم‌افزار (زبان‌های برنامه‌سازی)

«مهندسی کامپیوتر»

مولف: مریم موسوی ایرایی

انتشارات ارشد

پیشگفتار ناشر

خداوند بزرگ را شاکریم که فرصتی عنایت فرمود تا بتوانیم قدمی در جهت گسترش علم و فرهنگ کشور عزیزمان برداریم و مسیری هموار را در اختیار مخاطبان و داوطلبان علاقمند به ادامه تحصیل قرار دهیم. از آنجائی که درحوزه کتاب‌های درسی و کمک درسی کارشناسی ارشد کتاب‌های زیادی به رشته تحریر در آمده است با این حال وجود یک مجموعه جامع، دقیق، کامل و دارای رفرنس(منابع) مستند احساس می‌شود.

همیشه دغدغه داشتیم که ای کاش می‌شد تمامی منابع پراکنده و گاهاً دست نایافتنی کنکور در یک مجموعه به گونه‌ای جذاب و شیوا آماده می‌شد و همچنین سوالات دوره‌های قبل کنکور نیز به گونه‌ای تشریحی پاسخ داده می شدند، که خدشه‌ای بر آنها وارد نشود و خیال داوطلبان عزیز آنقدر آسوده باشد که اطلاعات مندرج در پاسخنامه‌ها را برای بهره‌گیری در آزمون خود بکار گیرند. این کار با یاری خداوند بزرگ و همکاری نخبه‌ترین دانشجویان و اساتید کشور در انتشارات ارشد هماهنگ گردید و در تمامی مقاطع تحصیلی از کاردانی به کارشناسی، کارشناسی ارشد و دکتری در حوزهٔ آزمون‌های سراسری – آزاد و پیام نور بسته‌های آموزشی شامل کتاب‌های بانک سوالات و کتاب‌های شرح جامع درس آماده گردید. در تهیه مجموعه‌های فوق‌الذکر از فارغ‌التحصیلان کارشناسی ارشد و دکتری که حائز رتبه‌های تک رقمی کنکور می‌باشند، استفاده کرده‌ایم، تا شما عزیزان بتوانید در راستای افزایش توان علمی خود در جهت نیل به مقصود خویش نایل آیید. بدون تردید می‌توان گفت که برای اولین بار در نظام آموزشی کشور مجموعه‌ای به این سبک عرضه می‌شود که در کیفیت و کمیت بی‌نظیر بوده و تمامی نیازهای دانشجویان و داوطلبان عزیز را پوشش می‌دهد. امید است که استفاده از این مجموعه عزیزان را در جهت موفقیت در آزمون یاری رسانده باشد. در پایان لازم به ذکر است که بی نقصی محض از آن خداوند متعال می‌باشد، بنا براین از تمامی خوانندگان و داوطلبان عزیز خواهشمندیم نظرات سازنده خود را در مورد این کتاب به سایت انتشارات ارشد اعلام نمایند تا ما را در جهت هر چه بهتر آماده کردن این محصولات یاری رسانند.

با تشکر

ناشـــــر

www.arshadbook. ir

پیشگفتار مولف

به نام آنکه جان را فکرت آموخت　　چراغ دل به نور جان برافروخت

همواره یکی از مؤثرترین شیوه‌های آشنایی برای هر آزمون هماهنگ، مطالعه در آزمون‌های پیشین به عمل آمده بوده است. در همین راستا، برآن شدیم که مجموعه‌ای از سلسله آزمون‌های زبانهای برنامه‌سازی کارشناسی ارشد را تدوین کرده تا قدمی هر چند اندک در جهت کامیابی شما داوطلب گرانقدر برداشته باشیم.

این مجموعه متشکل از تست‌های آزمون‌های پیشین و تست‌های تألیفی که فراخور آزمون کارشناسی ارشد می‌باشد گردآوری شده است

در خاتمه بر خود می‌دانم از مدیریت محترم انتشارات ارشد و عزیزان پرتلاش واحد انتشارات تشکر و قدردانی نمایم.

همچنین از خوانندگان اهل کمال می‌خواهیم با گوشزد کردن خطاهایم در این مجموعه ناچیز نگارنده را مورد لطف و راهنمایی خود قرار دهند.

با احترام

مریم موسوی ایرایی

mi.maryam@gmail.com

فهرست مطالب

عنوان **صفحه**

پیشگفتار ناشر ۳
پیشگفتار مولف ۴

فصل اول: اصول طراحی زبانهای برنامه سازی

۱-۱ مقدمه ۱۳
۱-۱-۱ برای بهبود توانایی خود در توسعه الگوریتمهای کارآمد ۱۳
۱-۱-۲ استفاده بهینه از زبان برنامه نویسی موجود ۱۳
۱-۱-۳ می توانید با اصطلاحات مفید ساختارهای برنامه نویسی آشنا شوید ۱۳
۱-۱-۴ انتخاب بهترین زبان برنامه سازی ۱۳
۱-۱-۵ طراحی زبان جدید ساده می‌شود ۱۴
۱-۲ تعریف زبان برنامه سازی ۱۴
۱-۳ ویژگیهای زبان خوب ۱۴
۱-۳-۱ وضوح و سادگی زبان (Clarity and Simplicity) ۱۴
۱-۳-۲ قابلیت تعامد (Orthoganility) ۱۴
۱-۳-۳ قاطعیت ۱۴
۱-۳-۴ طبیعی بودن زمان برای کاربردهایش ۱۴
۱-۳-۵ پشتیبانی از انتزاع (Support of abstraction) ۱۴
۱-۳-۶ سهولت کنترل صحت برنامه(ease of Verification) ۱۵
۱-۳-۷ محیط برنامه نویسی (Programming environment) ۱۵
۱-۳-۸ قابلیت حمل برنامه (Portability) ۱۵
۱-۳-۹ هزینه استفاده ۱۵
۱-۳-۹-۱ هزینه تولید برنامه ۱۵
۱-۳-۹-۲ هزینه کامپایل برنامه ۱۵
۱-۳-۹-۳ هزینه اجرای برنامه ۱۵
۱-۳-۹-۴ هزینه نگهداری برنامه ۱۶
۱-۳-۱۰ قابلیت اطمینان (Reliability) ۱۶
۱-۴ محیط هدف (Target environment) ۱۶
۱-۴-۱ محیط پردازش دسته ای(Batch processing target environment) ۱۶
۱-۴-۲ محیط محاوره ای (Interactive target environment) ۱۶

عنوان **صفحه**

۱-۴-۳ محیط توکار (Embedded target enviroment) ۱۷
۱-۵ استانداردسازی زبانها ۱۷
۱-۵-۱ استاندارد مالکیت (Proprietary Standard) ۱۷
۱-۵-۲ استاندارد اجماع (Consensus Standard) ۱۷
۱-۵-۳ عوامل مؤثر دراستانداردها ۱۷
۱-۶ مدل‌های محاسباتی زبانهای برنامه سازی ۱۸
۱-۶-۱ زبانهای دستوری (imperative languages) ۱۸
۱-۶-۲ زبانهای تابعی (applicative languages) ۱۹
۱-۶-۳ زبانهای قانونمند(Rule – Based languages) ۲۰
۱-۶-۴ مدلهای محاسباتی ترکیبی (Hybrid Computational Models) ۲۱

فصل دوم: اصول پیاده‌سازی زبان‌های برنامه‌سازی

۲-۱ مقدمه ۲۹
۲-۲ عملکرد کامپیوتر ۲۹
۲-۲-۱ کامپیوترهای سخت‌افزاری ۲۹
۲-۲-۲ کامپیوترهای میان‌افراز ۳۰
۲-۲-۳ کامپیوترهای شبیه‌سازی شده نرم‌افزاری ۳۰
۲-۲-۴ کامپیوترهای ترکیبی ۳۱
۲-۳ سلسله مراتب کامپیوترهای مجازی ۳۱
۲-۴ مقایسه کامپایل و تفسیر ۳۲
۲-۴-۱ هزینه تولید کد ۳۳
۲-۴-۲ هزینه اجرای برنامه ۳۳
۲-۴-۳ بهینه‌سازی ۳۳
۲-۴-۴ هزینه حافظه مصرفی ۳۳
۲-۴-۵ قابلیت اطمینان ۳۳
۲-۴-۶ انعطاف‌پذیری ۳۳
۲-۵ انقیاد (binding) ۳۴
۲-۵-۱ زمان توصیف زبان (specification of language) ۳۵
۲-۵-۲ زمان تعریف زبان (declaration of language) ۳۵
۲-۵-۳ زمان پیاده‌سازی زبان (implementation of language) ۳۵
۲-۵-۴ زمان ترجمه زبان (translation of language) ۳۵
۲-۵-۵ زمان اجراء زبان (running of language) ۳۶

فصل سوم: اصول ترجمه زبانهای برنامه‌سازی

۳-۱ مقدمه ۴۵

عنوان **صفحه**

۲- ۳ معیارهای عمومی Syntax زبان ۴۵
۱- ۲- ۳ قابلیت خوانایی (readability) ۴۵
۲- ۲- ۳ قابلیت نوشتن (writability) ۴۵
۳-۲-۳ سهولت بازرسی (ease of verification) ۴۵
۴-۲- ۳ سهولت ترجمه (ease of translation) ۴۶
۵-۲-۳ عدم وجود ابهام (lack of ambiguites) ۴۶
۳-۳ مراحل ترجمه زبان‌ها ۴۷
۱- ۳- ۳ تحلیل لغوی (lexical analysis) ۴۷
۲- ۳- ۳ تحلیل نحوی (syntactic and lyses) ۴۷
۳-۳-۳ تحلیل معنایی (Semantic and lyses) ۴۷
۴- ۳- ۳ بهینه‌سازی (optimization) ۴۸
۵-۳-۳ تولید کد (code generation) ۴۸
۴-۳ مدلهای مفهومی زبان‌های برنامه‌سازی (programming languages semantic model) ۵۰
۱-۴-۳ استفاده از زبانهای طبیعی ۵۰
۲-۴-۳ استفاده از گرامر ۵۰
۳-۴-۳ مدلهای دستوری یا عملیاتی ۵۰
۴-۴-۳ مدلهای تابعی ۵۰
۵-۴-۳ مدلهای مشخصه ۵۰

فصل چهارم: انواع داده اولیه

۱-۴ انواع داده اولیه ۵۹
۲-۴ مشخصات انواع داده اولیه(specificaton of elementary data type) ۶۰
۱-۲-۴ صفات نوع داده (attributes) ۶۰
۲-۲-۴ مقادیر نوع داده (Values) ۶۰
۳-۲-۴ عملیات (operations) ۶۰
۳-۴ اعلان انواع داده اولیه (declaration of elementary data types) ۶۱
۴-۴ پیاده‌سازی انواع داده اولیه …… (Implementation of ۶۲
۵-۴ کنترل نوع داده اولیه …… (type- checking / association ۶۲
۱-۵-۴ کنترل نوع ایستا (Static) ۶۲
۲-۵-۴ کنترل نوع پویا (Dynamic) ۶۳
۶-۴ تبدیل نوع (Data type conversion) ۶۳
۱-۶-۴ تبدیل نوع اجباری (coercion conversion) ۶۳
۲-۶-۴ تبدیل نوع اختیاری (explicit conversion) ۶۴
۷-۴ عملیات انتساب (assignnment operation) ۶۴
۱-۷-۴ انتساب معمولی (Direct assignment) ۶۴

عنوان **صفحه**

۴-۷-۲ انتساب اشاره‌گری (pointer assignment) ۶۵
۴-۸ انواع داده اسکالر (Scaler data types) ۶۵
۴-۸-۱ نوع داده صحیح (integer) ۶۵
۴-۸ -۲ نوع داده ممیز ثابت (fixed point) ۶۶
۴-۸-۳ اعداد حقیقی ممیز شناور ۶۷
۴-۸-۴ اعداد موهومی ۶۷
۴-۸-۵ اعداد گویا ۶۷
۴-۸-۶ نوع داده شمارشی (enumeric) ۶۷
۴-۸-۷ نوع داده بولی (Boolean) ۶۸
۴-۸-۸ کاراکترها (characters) ۶۸

فصل پنجم: بسته بندی

۵-۱ انواع داده ساخت یافته ۷۹
۵ -۱ -۱ مشخصات انواع داده ساخت یافته ۷۹
۵-۱-۲ عملیات روی انواع داده ساخت یافته ۸۰
۵-۱-۳ اعلان نوع داده ساخت یافته ۸۰
۵-۱-۴ پیاده‌سازی نوع داده ساخت یافته ۸۰
۵-۱-۵ کنترل نوع داده‌های ساخت یافته ۸۱
۵-۲ طراحی پیاده‌سازی تعدادی از انواع داده ساخت یافته ۸۱
۵-۲-۱ پیاده‌سازی ساختمان‌داده همگن ۸۱
۵-۲-۲ پیاده‌سازی ساختمان داده ناهمگن ۸۳
۵-۲-۳ رکورد با طول متغیر ۸۳
۵-۲-۴ اشاره‌گرها ۸۵
۵-۲-۵ لیست‌های پیوندی ۸۷
۵-۲-۶ مجموعه‌ها ۸۹
۵-۳ اشیای دادهٔ اجرایی ۸۹
۵-۴ انتزاع (abstraction) ۹۰
۵-۵ ویژگی‌های کلی انتزاع ۹۰
۵-۶ تعریف نوع ۹۰
۵-۷ هم ارزی اعلان ۹۱
۵-۷-۱ تساوی نوع ۹۱
۵-۷-۲ هم ارزی نام ۹۲
۵-۷-۳ هم ارزی ساختاری ۹۲
۵-۷-۴ تساوی اشیای داده ۹۳

عنوان **صفحه**

فصل ششم: کنترل ترتیب اجراء

۶-۱ مقدمه ۱۰۳
۶-۲ ترتیب اجراء در عبارات محاسباتی ۱۰۳
۶-۲-۱ نشانه گذاری prefix ۱۰۴
۶-۲-۲ ارزیابی عبارات prefix ۱۰۴
۶-۲-۳ نشانه گذاری postfix ۱۰۴
۶-۲-۴ ارزیابی عبارات postfix ۱۰۵
۶-۲-۵ نشانه گذاری infix ۱۰۵
۶-۲-۶ ارزیابی عبارات infix ۱۰۵
۶-۲-۷ شرکت پذیری ۱۰۵
۶-۲-۷-۱ زبان APL ۱۰۵
۶-۲-۷-۲ زبانForth ۱۰۵
۶-۳ نمایش زمان اجراء در عبارات محاسباتی ۱۰۶
۶-۴ ارزیابی نمایش درختی عبارات ۱۰۶
۶-۴-۱ مسئله ۱: قواعد ارزیابی یکنواخت ۱۰۶
۶-۴-۲ مسئله۲: اثرات جانبی ۱۰۶
۶-۴-۳ مسئله۳: شرایط خطاء ۱۰۷
۶-۴-۴ مسئله۴: عبارت بولین مدار کوتاه ۱۰۷
۶-۵ کنترل ترتیب بین دستورات ۱۰۷
۶-۵-۱ شکل‌های مختلف کنترل ترتیب سطح دستور ۱۰۷
۶-۵-۲ کنترل ترتیب ضمنی ۱۰۸
۶-۵-۳ مزایا و معایب goto ۱۰۸
۶-۵-۴ دستورات شرطی ۱۰۸
۶-۶ برنامه‌های بنیادی ۱۱۰

فصل هفتم: کنترل زیربرنامه

۷-۱ تعریف زیربرنامه ۱۱۹
۷-۱-۱ مشخصات و پیاده سازی زیربرنامه ۱۱۹
۷-۱-۲ تعریف و سابقه فعالیت زیر برنامه ۱۱۹
۷-۱-۳ زیربرنامه کلی (generic subprogram) ۱۲۰
۷-۱-۴ کنترل ترتیب زیربرنامه ۱۲۰
۷-۱-۵ پیاده‌سازی ساختار کنترلی فراخوانی- برگشت ۱۲۱
۷-۱-۶ زیربرنامه‌های بازگشتی ۱۲۲
۷-۱-۷ اعلان پیشرو در پاسکال ۱۲۲

عنوان **صفحه**

۷-۲ صفات کنترل داده‌ها ۱۲۳
۷-۲-۱ اسامی و محیطهای ارجاع ۱۲۳
۷-۲-۲ انواع اسامی در برنامه ۱۲۴
۷-۲-۳ وابستگی‌ها و محیط‌های ارجاع ۱۲۴
۷-۲-۴ نام مستعار برای اشیای داده (aliasing) ۱۲۴
۷-۲-۵ حوزه ایستا و پویا ۱۲۶
۷-۲-۵-۱ ساختار بلوکی ۱۲۶
۷-۲-۵-۲ پیاده‌سازی قاعده حوزه پویا ۱۲۶
۷-۳ پارامترها و انتقال پارامترها ۱۲۷
۷-۳-۱ پارامترها و تناظر بین پارامترها ۱۲۷
۷-۳-۲ روشهای انتقال پارامترها ۱۲۸
۷-۳-۳ پیاده‌سازی انتقال پارامتر ۱۲۹
۷-۳-۴ مثالی از انتقال پارامترها ۱۲۹
۷-۴ محیط‌های مشترک صریح ۱۳۰

فصل اول

اصول طراحی زبانهای برنامه سازی

عناوین اصلی

- مقدمه
- تعریف زبان برنامه‌سازی
- ویژگیهای زبان خوب
- محیط هدف
- استاندارد سازی زبانها
- مدل‌های محاسباتی زبانهای برنامه‌سازی

فصل اول

اصول طراحی زبانهای برنامه سازی

۱-۱ مقدمه

هدف از درس طراحی و پیاده سازی زبانهای برنامه سازی مقایسه زبانهای برنامه سازی برای رسیدن به مقاصد زیر است:

۱-۱-۱ برای بهبود توانایی خود در توسعه الگوریتمهای کارآمد

بسیاری از زبانها ویژگیهایی دارند که اگر به خوبی مورد استفاده قرار گیرند به نفع برنامه نویس است ولی اگر به صور نامناسب مورد استفاده قرار گیرند، وقت زیادی صرف برطرف کردن خطا های منطقی برنامه می شود. نمونه ای از آن بازگشتی است که اگر به خوبی مورد استفاده قرار گیرد منجر به پیاده سازی کارآمد الگوریتم می شود.

۱-۱-۲ استفاده بهینه از زبان برنامه نویسی موجود

با درک این نکته که چگونه ویژگیهای یک زبان پیاده‌سازی می شوند، توانایی شما در نوشتن برنامه های کارآمد افزایش می یابد. به عنوان مثال، اگر بدانید که ساختمان داده های گوناگون چگونه پیاده سازی می شوند، می توانید برنامه های کارآمدی ایجاد نمائید که از این مفاهیم استفاده نمایند.

۱-۱-۳ می توانید با اصطلاحات مفید ساختارهای برنامه نویسی آشنا شوید

درک تکنیکهای برنامه نویسی اهمیت فراوانی دارد، زیرا برای استفاده از یک ساختار برنامه نویسی که در یک زبان موجود نیست، برنامه نویس باید آن را با استفاده از عناصر اولیه موجود در زبان پیاده سازی نماید.

۱-۱-۴ انتخاب بهترین زبان برنامه سازی

آشنایی با چند زبان برنامه سازی باعث می شود که زبان مناسبی برای پروژه خاصی انتخاب شود. به عنوان مثال، کاربردهایی که نیاز به محاسبات عددی دارند، بهتر است از Fortran,C یا Ada استفاده کنند. در کاربردهای تصمیم گیری مثل هوش مصنوعی از Lisp , ML یا Prolog استفاده می شود.

۱-۱-۵ طراحی زبان جدید ساده می‌شود

یک ویژگی در دو زبان مختلف ممکن است به روشهای متفاوتی پیاده سازی شود و در نتیجه هزینه استفاده از آنها متفاوت باشد. به عنوان مثال، تمام زبانها عمل جمع را انجام می دهند ولی هزینه آنها در زبان C , Cobol و Smalltalk ممکن است با هم متفاوت باشد.

۱-۲ تعریف زبان برنامه سازی

هر نشانه گذاری برای توصیف الگوریتم‌ها و ساختمان داده ها را می توان یک زبان برنامه سازی نامید.

۱-۳ ویژگیهای زبان خوب

۱-۳-۱ وضوح و سادگی زبان (Clarity and Simplicity)

زبان برنامه سازی باید از لحاظ مفهوم ساده و روشن باشد و مشخص باشد برای چه منظوری ایجاد گردیده است. زبان از نظر Syntax نیز باید خوانا باشد تا برنامه های نوشته شده قابل تصحیح باشند. از نظر Syntax زبان Pascal شفافیت بالایی دارد ولی زبانی مثل Lisp یا APL چنین ویژگی ندارند.

۱-۳-۲ قابلیت تعامد (Orthoganility)

منظور از تعامد قابلیت ترکیب ویژگیهای مختلف زبان است. به عنوان مثال، ترکیب عبارات محاسباتی و عبارات شرطی. وقتی ویژگیهای یک زبان متعامد باشد، یادگیری زبان ساده بوده، نوشتن برنامه ها راحت تر است. جنبه منفی تعامد این است که گرچه ممکن است برنامه شامل ترکیبی از ویژگی‌هایی باشد که از نظر منطقی روشن نباشند و اجرای آن نیز کارآمد نباشد ولی برنامه بدون خطاء کامپایل گردد.

۱-۳-۳ قاطعیت

برنامه سازی باید فاقد ابهام باشد، منظور از ابهام این است که از یک Syntax به دو مفهوم رسید. برای مثال در Fortran عبارت f (i , j) هم برای اعمال تابع f برروی متغیرهای i و j کاربرد دارد و هم برای دسترسی به عنصر (i , j) از ماتریس f . اما همین ابهام در Pascal وجود ندارد. (i , j) برای تابع f [i , j] برای ماتریس به کار می رود.

۱-۳-۴ طبیعی بودن زمان برای کاربردهایش

معمولاً زبانها هرچه با زمینه کاریشان سازگارتر باشند بهترجواب خواهند داد. برای مثال می توانیم واحدها و مکانیزم کنترل به راحتی با C هم بنویسیم ولی بهتر است از OCCAM که برای این منظور طراحی شده است استفاده کنیم.

۱-۳-۵ پشتیبانی از انتزاع (Support of abstraction)

مهمترین خصوصیات انتزاع، پنهان کردن داده ها است. مثلاً زیر برنامه ها نوعی انتزاع عملیاتی هستند. ساده ترین انتزاع داده‌ای رکوردها هستند. افزایش انتزاع در زبانهای برنامه نویسی باعث بالارفتن قابلیت اطمینان می گردد. منظور از انتزاع پنهان سازی داده ها در یک مفهوم است. انتزاع دارای طیفی است. زبان ماشین از حداقل انتزاع برخوردار است. زبان Pascal

انتزاع داده ای را توسعه داده و انتزاع داده ای در Ada به اوج خود می رسد و زبانهایی چون Java و C++ هم این خصوصیت را از Ada به ارث برده اند. به عنوان مثال، دستورات public و Private در C++ نمونه ای از پیاده سازی انتزاع است.

۱-۳-۶ سهولت کنترل صحت برنامه(ease of Verification)

یعنی با دیدن Source برنامه بتوانیم تطابق آن را با الگوریتم مورد نظر چک کنیم این اشکال زدایی در مرحله Syntax (قبل از کامپایل کردن برنامه) است و با debug فرق می کند. کنترل صحت برنامه یا بازرسی رسمی صورت می گیرد و یا با استفاده از داده های آزمایشی خروجی و ورودی آن را امتحان می کنند.

۱-۳-۷ محیط برنامه نویسی (Programming environment)

عوامل زیادی را می توان به عنوان محیط برنامه نویسی در نظر گرفت، وجود پیاده سازی قابل اعتماد، کارآمد و با مستندسازی خوب از عوامل مهم هستند. ویراستارهای خاص و بسته های تست برنامه موجب تسریع در ایجاد و عیب‌یابی برنامه می شوند.

۱-۳-۸ قابلیت حمل برنامه (Portability)

یک معیار مهم برای پروژه های برنامه نویسی قابلیت انتقال یک برنامه از یک سیستم کامپیوتری به سیستم کامپیوتری دیگر است. در مرحله طراحی زبان، قابلیت حمل هدف واقع می شود. هر چه زبان به سخت افزار ماشین نزدیک باشد قابلیت حمل آن کمتر است. به عنوان مثال، Java کاملاً مستقل از لایه های پیاده سازی است و نیاز به هزینه واسط برای برقراری ارتباط با کامپیوترها دارد، افزایش قابلیت حمل باعث کاهش کارآیی و انعطاف پذیری زبان‌ها می‌گردد.

۱-۳-۹ هزینه استفاده

۱-۳-۹-۱ هزینه تولید برنامه

زمان و حافظه مصرفی و پیچیدگی الگوریتم از جمله موارد مهم هزینه تولید برنامه است .

۱-۳-۹-۲ هزینه کامپایل برنامه

در کامپایل از Source code به object code می رسیم هرچه فاصله این دو کد کمتر باشد، هزینه کامپایل نیز کمتر می گردد. زبانهای ویژوال هزینه کامپایل بیشتری نسبت به زبانهای غیر ویژوال نظیرشان دارند.

۱-۳-۹-۳ هزینه اجرای برنامه

این هزینه مربوط به پیاده سازی زبان است. برای محاسبه هزینه اجراء ۲ روش وجود دارد:

الف) مقایسه زمان اجراء.

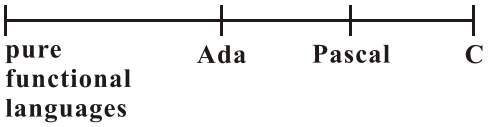
ب) محاسبه هزینه اجراء با توجه به پیاده سازی که با توجه به این که دسترسی به پیاده سازی دقیق زبانها غیرممکن است کار دشواری است.

۱-۳-۹-۴ هزینه نگهداری برنامه

نگهداری برنامه شامل هزینه های توسعه برنامه، ترمیم خطاها پس از اجرای برنامه، تغییر نیازمندیهای برنامه به دلیل تغییر در سخت افزار و سیستم عامل است .

۱-۳-۱۰ قابلیت اطمینان (Reliability)

وقتی می گوئیم یک زبان برنامه سازی قابل اعتماد است، منظور این است که اگر برنامه ای به این زبان نوشته شود تحت هر شرایطی به یک ورودی مشخص یک خروجی واحد بدهد. عواملی در افزایش قابلیت اطمینان زبانها مؤثرند مانند استاندارد سازی زبانها، استفاده از برخی feature های سودمند مانند انتزاع حفاظت شده، کاهش وابستگی داده ای. میزان قابلیت اطمینان زبانهای برنامه سازی دارای طیفی به شکل زیر است:



شکل ۱-۱ طیف میزان قابلیت اطمینان زبانهای برنامه‌سازی

۱-۴ محیط هدف (Target environment)

محیطی که زبان برنامه سازی درآن به کار گرفته می شود و برنامه در آن محیط نوشته و اجراء شود را محیط هدف گویند.

۱-۴-۱ محیط پردازش دسته ای(Batch processing target environment)

قدیمی ترین محیط برنامه سازی است که محیط پردازش درآن batch است یعنی فرض می شود کاربر وجود ندارد. داده ها مدتی روی هم انباشته می شوند و دریک مدت زمان معین باید اجراء گردد. زبان fortran از این محیط استفاده می نماید. ویژگیهای این محیط به شرح زیر است :

الف) از آنجا که در این محیط برنامه ساز حضور ندارد بنابراین باید زبان برنامه سازی درعملیات روی فایلها قدرتمند باشد.

ب) تشخیص و برطرف نمودن خطاهای برنامه بسیار مشکل است یک راه بسیار ساده دربرخورد باخطا تمام کردن برنامه است درغیر اینصورت زبان باید هوشمند باشد تا بتواند درغیاب کابر خطاها را برطرف نماید.

ج) زمان انجام کارها در این محیط اهمیتی ندارد بنابراین زبان به کار برده شده در این محیط، زمان مشخصی نخواهد داشت.

۱-۴-۲ محیط محاوره ای (Interactive target environment)

در این محیط کاربر حضور دارد، ولی واکنش بلادرنگ نیست، ویژگیهای این محیط عبارتند از:

الف) فرض براین است که از طریق مانیتور با کاربر و برنامه ساز تعامل وجود دارد بنابراین در پیاده سازی زبان باید ابزارهایی برای انجام تعامل درنظر گرفته شود.

ب) برخورد با خطاها واستثناها آسانتر است. هنگام روبرو شدن با مشکل از خود کاربر می پرسد که چه کاری را انجام دهد.

۱-۴-۳ محیط توکار (Embedded target enviroment)

این محیط توکار است یعنی نرم افزارهای کاربردی قرار است درمحیط های توکار استفاده شوند. به عنوان مثال، داخل یک اتومبیل یا داخل یک هواپیما. اولین زبانی که در پاسخگویی به این محیط ها ایجاد گردید، Ada است. برخی ویژگیهای این محیط عبارتند از:

الف) دراین محیط با سیستمهایی سروکار داریم که تحمل خطاء را ندارند بنابراین قابلیت اطمینان زبان اهمیت فوق العاده ای دارد. برای رسیدن به این خصوصیت طراح و پیاده ساز باید مبانی نظری زبان‌هایشان را تغییر بدهند.

ب) نیاز به پردازش بلادرنگ وجود دارد. زبان نیاز به سیستم عامل ندارد و باید خیلی سریع پاسخ دهد.

ج) مسأله برنامه نویسی توسعه یافته مطرح است .

د) اهمیت عملیات روی فایلها بسیار پائین است .

ه) مدیریت استثناها و خطاها از اهمیت بالایی برخوردار است .

۱-۵ استانداردسازی زبانها

هدف از استانداردسازی زبانها رسیدن به قابلیت اطمینان و قابلیت جابجایی برنامه های نوشته شده با آنها است. ما درحقیقت زبانهای برنامه سازی را استاندارد می کنیم نه کامپایلر زبانها را.

کامپایلر معادل پیاده سازی زبانها است. کامپایلری که بتواند برنامه نوشته شده به زبان استاندارد را اجراء کند و جواب صحیح بدهد، کامپایلر پیرو نامیده می شود کامپایلرهای پیرو معمولاً دستوراتی بیشتر از زبان استاندارد شده را پشتیبانی می کنند که این باعث توسعه زبان می شود، البته برنامه نویسی باید مواظب این دستورات اضافی باشد.

۱-۵-۱ استاندارد مالکیت (Proprietary Standard)

استانداردی است که پیاده ساز زبان با توجه به این که به ساختار زبان دسترسی دارد ارائه می نماید. در این نوع استاندارد نوع ماشینها و سیستمها عاملهای پشتیبان زبان اعلام می شوند. دلیل ارائه این استاندارد این است که اگر چنین کاری انجام شود، قابلیت حمل برنامه و جابجایی برنامه ها نوشته شده با این زبان کاهش می یابد. این استاندارد هیچگونه ضمانت اجرایی ندارد.

۱-۵-۲ استاندارد اجماع (Consensus Standard)

در این استاندارد گروهی روی زبان اجماع نظر می کنند و طراح و برنامه ساز مجبور به استفاده از استاندارد هستند. به عنوان مثال، مؤسسه های ANSI در آمریکا، BSI در بریتانیا و ISO بین المللی فارغ از این که چه کسی زبان را نوشته است به استاندارد کردن زبانها می پردازند. این شرکتها روی برنامه های نوشته شده با این زبانها ضمانت اجرایی دارند.

۱-۵-۳ عوامل مؤثر دراستانداردها

برای استفاده مؤثر از استانداردها سه مسئله را باید در نظر بگیریم:

الف) زمان شناسی (timeliness): چه زمانی زبان را استاندارد می کنیم؟

ب) اطاعت و پیروی (Conformance): منظور از این که برنامه باید از استاندارد پیروی کند و کامپایلر باید استانداردی را کامپایل کند، چیست؟

ج) کهنگی (Obsolescence): چه زمانی استاندارد کهنه می شود و چگونه باید اصلاح شود؟

۱-۶ مدل‌های محاسباتی زبانهای برنامه سازی

مدل محاسباتی زبان قبل از طراحی زبان تعیین شود. مدل محاسباتی یکی از مفاهیم اساسی هر زبان برنامه سازی و پشتوانه نظری آن است. شناخت بهتر مدل محاسباتی به شناخت کاملتر زبان می انجامد و تنوع زبانها ناشی از تنوع مدلهای محاسباتی است. براین اساس زبانها را به ۴ دسته تقسیم می کنند :

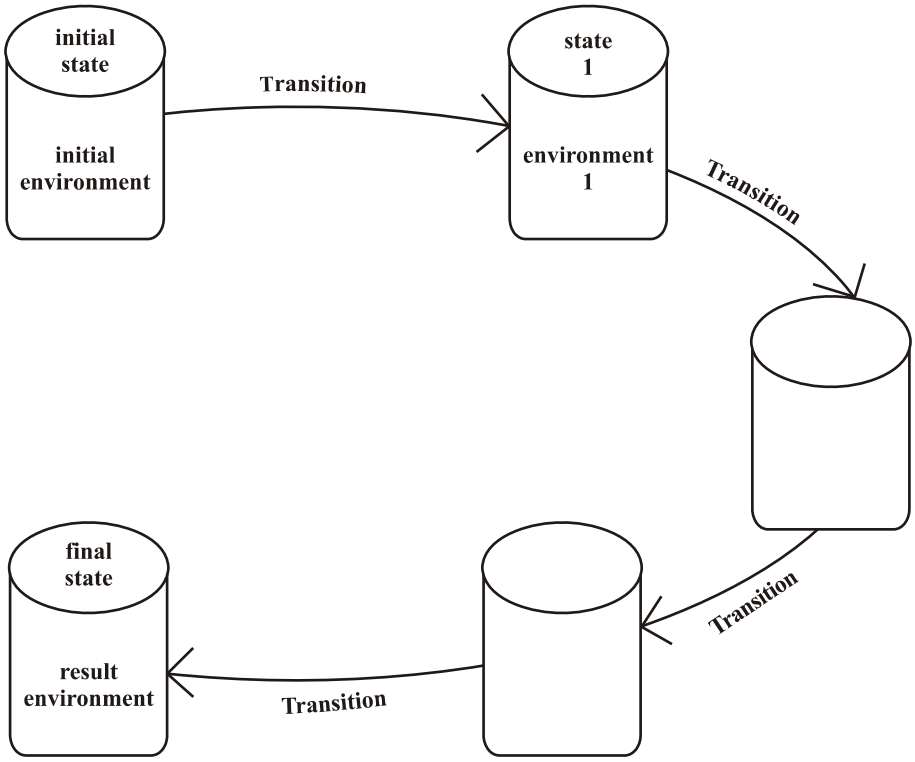
۱-۶-۱ زبانهای دستوری (imperative languages)

این نوع زبانها خصوصیاتی دارند که عبارتند از:

الف) اساس زبان را دستورات تشکیل می دهند.

ب) عمدتاً رویه ای هستند و الگوریتمها را مرحله به مرحله دنبال می کنند.

ج) مدلهای محاسباتی این زبانها مبتنی بر تئوری ، FSM(Finite state machine) است. در این تئوری یک حالت اولیه (initial state) داریم که با شروع برنامه در آن حالت قرار می گیریم. هردستوری که اجراء می شود ما را از حالتی به حالت دیگر می برد (transition) و سرانجام به حالت نهایی (final state) می رسیم. در این مدل حتماً یک شرط خاتمه لازم است به همین علت است که در C و pascal نمی توانیم حلقه بی پایان داشته باشیم.



شکل ۲-۱: مدل زبانهای دستوری

در پیاده سازی زبان تعریف هر State دقیقاً مشخص باشد. هر State و environment (مجموعه داده های ساخت یافته که در رابطه با پیاده سازی مطرح هستند) برنامه را مشخص می کند.

اجرای هر دستور update کردن environment و در نتیجه ورود به یک state جدید است. در final state که برنامه در آن به پایان می رسد، حاوی مجموعه اطلاعات ساخت یافته است که برنامه برای رسیدن به آن اجراء گردیده است.

مزیت این مدل محاسباتی در این است که درهر مقطع می توانیم، environment را چک کنیم. در این مدل بازگشت به حالات قبلی غیر ممکن است و با هر transition به یک state جدید می رویم. بنابراین حلقه نداریم و برای پیاده سازی دستور loop آن را باز می کنیم و هر بخشی loop را یک حالت جدید در نظر می گیریم.

مدل محاسباتی مدل اجراء است یعنی در دستورات شرطی دو شاخه خروجی از یک حالت نداریم، به عنوان مثال در خروجی دستور if تنها یکی از شاخه های then یا else را بر می گزینیم و دیگری را به فراموشی می سپاریم و این یکی از معایب این مدل می باشد.

کامپیوترهایی که از این مدل پشتیبانی می کنند، سیستمهای ون نیومن و ماشینهای تورینگ هستند. نمونه زبانهای این مدل محاسباتی C , Pascal و fortran هستند. تفاوت این زبانها در این است که transition ها را با چه دستوراتی انجام می دهند.

۲-۶-۱ زبانهای تابعی (applicative languages)

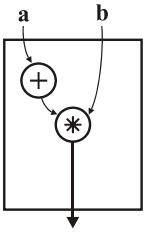
دراین روش به جای transitionها، عملکرد برنامه دنبال می شود و باید به این سؤال پاسخ داده شود: چه عملی باید برروی حالت اولیه ماشین انجام شود تا با دستیابی به مجموعه اولیه متغیرها و ترکیب آنها پاسخ مناسب داده شود؟ بنابراین در این زبانها، توابع به عنوان اصل پیاده سازی و مدل محاسباتی پشتیبان هستند. به جای ملاحظه حالتهای متوالی محاسبات ماشین تبدیلات تابعی متوالی را درنظر می گیریم، که باید برروی داده های اولیه انجام شوند تا به پاسخ خود برسیم. وقتی به پاسخ خود رسیدیم، آن را برروی داده های اولیه اجراء می کنیم تا نتیجه نهایی بدست آید.

$$\text{Function}_n (\ldots \text{Function}_2 (\text{Function}_1 (\text{data}))\ldots)$$

ساختار برنامه های این زبانها تابعی یا مبتنی بر جریان داده (dataflow) است. در زبانهای تابعی الگوریتم برنامه تقسیم می شود به چگونگی تعریف توابع و چگونگی اعمال توابع برنامه نوشته شده به زبانهای تابعی به عنوان یک عبارت اجراء می شود. (شکل ۱-۲) .

Header
Function Definition Section
Function Application

شکل ۳-۱: Data Flow paradigm



شکل ۴-۱ Functional Languages Paradigm

ماشینهایی طراحی شده اند که زبانهای dataflow را اجراء کنند، اما همین زبانها می توانند روی کامپیوترهای معمولی نیز اجراء شوند به عنوان مثال در نوعی پیاده سازی این زبانها جدولی وجود دارد که هر گره و ورودیها و خروجیهای آن را مشخص می کند و برروی کامپیوترهای تک پردازنده اجراء می شود.

زبان Lisp یکی از قدیمی ترین و مهمترین زبانهای تابعی است.

۳-۶-۱ زبانهای قانونمند(Rule – Based languages)

این زبانها مبتنی بر قواعدی هستند که ما تعریف می کنیم. در این مدل محاسباتی اساسی کارگزاره ها (Productions) هستند یعنی شرطی (Condition) را قرار می دهیم که اگر برقرار بود، کاری (action) انجام شود، مثلاً در زبانهای Imperative گزاره Case نمونه هایی از این مدل هستند.

enabling condition 1 → action 1

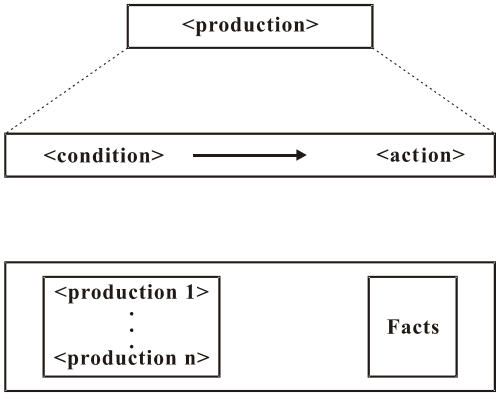
enabling condition 2 → action 2

...

enabling condition n → action n

در این زبانها یک مجموعه از Rules (قوانین) و یک مجموعه از Facts (حقایق) داریم. به عنوان مثال، در روابط خانوادگی ،رابطه خواهر و برادری یک Rule است و fact در آن وجود ندارد ولی سارا خواهر علی است یک fact است، آنچه در این جا مطرح است تشکیل این جدول صحت است که با گرفتن facts و به کاربردن Rules به پاسخ مطلوب برسد. همانطور که می بینید اساس محاسبه Rules است.

تئوری پشتیبان این زبانها فرضیه های منطقی (logic theory) است. ساختار برنامه های این زبانها به صورت، شکل ۴-۱ است.



شکل ۴-۱ Program Structure of Rule-Based Languages

۴-۶-۱ مدلهای محاسباتی ترکیبی (Hybrid Computational Models)

زبانهایی وجود دارند که مدل محاسباتی آنها ترکیبی از دو مدل محاسباتی است که مانند زبانهای شیء گرا که مدل محاسباتی آنها ترکیب مدل محاسباتی زبانهای Imperative و زبانهای applicative است. البته این ترکیب به این معنی نیست که تمام خصوصیات هردو را دارد. در زبانهای شیءگرا پایه مدل محاسباتی زبانهای Imperative است و دربعضی جاها مبحث هایی از مدل محاسباتی Applicative وارد آن می شود تا قابلیت اطمینان زبان را بالا ببرد. نمونه این ترکیب زبان C++ است .

البته باید توجه داشت که به هیچ وجه نمی توان تمام مدلهای محاسباتی را ترکیب نمود و یک مدل محاسباتی جامع به دست آورد.چون در این صورت به تناقض بر می خوریم. مثلاً در زبانهای مانند Pascal نیاز به حافظه اصلی ضروری است درحالیکه در زبانهای مبتنی بر جریان داده اصلاً داده وجود ندارد بنابراین نیاز به حافظه اصلی وجود ندارد.

نکات کلیدی

- ویژگیهای یک زبان خوب عبارتند از : وضوح و سادگی زبان، قابلیت تعامد، قاطعیت، طبیعی بودن، پشتیبانی از انتزاع، سهولت کنترل صحت برنامه، محیط برنامه نویسی،قابلیت حمل برنامه و قابلیت اطمینان.
- هزینه استفاده از زبان شامل هزینه تولید برنامه،هزینه کامپایل برنامه، هزینه اجرای برنامه وهزینه نگهداری برنامه است.
- انواع محیط هدف عبارتند از : پردازش دسته‌ای، محاوره‌ای و تو کار.
- استاندارد سازی زبانها به دو روش انجام می‌شود: استاندارد مالکیت و استاندارد اجماع.
- عوامل مؤثر در استانداردها عبارتند از: زمان شناسی،پیروی و کهنگی.
- انواع مدل‌های محاسباتی عبارتند از: دستوری،تابعی، قانونمند و ترکیبی.
- در محیط‌های توکار قابلیت اطمینان اهمیت زیادی دارد چون این سیستمها تحمل خطاء ندارند.
- زبانهای تابعی خالص بالاترین سطح قابلیت اطمینان را دارند.
- در محیط‌های توکار قابلیت اطمینان اهمیت زیادی دارد.
- در محیط‌های توکار پردازش بلادرنگ لازم است.
- در محیط های دسته‌ای به دلیل عدم حضور کاربر برخورد با استثناها و خطاها مشکل است و مسئله زمانبندی اهمیت کمی دارد.
- منظور از قابلیت جا به جایی قطع وابستگی برنامه به Platform است.
- منظور از قابلیت اطمینان این است که برنامه نوشته شده به یک ورودی مشخص تحت هر شرایطی خروجی واحد دهد.
- مدل محاسباتی زبانهای شیءگرا ترکیب دو مدل دستوری و تابعی است.
- امکان ایجاد یک مدل محاسباتی جامع برای همهٔ زبانها وجود ندارد.

تست‌های آخر فصل

۱- کدام یک از معیارهای زیادی برای انتخاب زبان مناسب در کاربردهای توکار (embedded systems) اهمیت بیشتری دارد؟ (سراسری ۸۲)

۱) یکنواختی (Uniformity)
۲) قابلیت توسعه (Extensibility)
۳) عمومیت (Generality)
۴) قابلیت اطمینان (Reliability)

۲- کدام یک از موارد ذیل از اهداف زبان الگول (Algol) نیست ؟ (تألیفی)

۱) نشانه های الگول باید به ریاضیات استاندارد نزدیک باشد.
۲) الگول باید برای توصیف الگوریتم ها مفید باشد .
۳) برنامه ها در الگول باید به زبان ماشین ترجمه شوند .
۴) الگول باید به معماری یک ماشین مقید (وابسته) باشد.

۳- یک سیستم کامپیوتری که برای کنترل بخشی از یک سیستم بزرگ مثل ماشین آلات صنعتی ، هواپیما، ماشین تراش، اتومبیل یا مترو به کار می رود؟ (تألیفی)

۱) دسته ای (batch)
۲) محاوره ای (interactive)
۳) اشتراک زمانی (timesharing)
۴) سیستم‌های تعبیه شده (embedded systems)

۴- در کدام گزینه هر سه زبان، در کاربردهای تصمیم گیری مثل هوش مصنوعی استفاده می شود؟ (تألیفی)

۱) Ada , C++ و Smalltalk
۲) Prolog , Lisp و ML
۳) Lisp , C++ و Java
۴) XML , APL و Perl

۵- کدام مورد دلایل مطالعه زبانهای برنامه سازی است ؟ (تألیفی)

۱) استفاده بهینه از زبان برنامه نویسی موجود
۲) شناختن ساختارهای مفید زبانهای برنامه نویسی
۳) انتخاب بهترین زبان برنامه سازی برای یک پروژه خاص
۴) همه موارد فوق صحیح است

۶- اولین مدل از زبانهای برنامه سازی کدامیک از مدل‌های زیر است ؟ (تألیفی)

۱) زبان‌های مبتنی بر اعداد
۲) زبان‌های شیء گرا
۳) زبانهای تجاری
۴) زبانهای منطقی

۷- کدامیک از زبان‌های زیر برنامه نویسی اعلانی (declarative Programming) را حمایت می‌کند؟ (تألیفی)

۱) Prolog
۲) C ++
۳) Java
۴) Ada

۸- بانک اطلاعاتی Prolog شامل حقایق و قواعد می باشد. کدامیک از موارد زیر می تواند یک حقیقت باشد؟ (تألیفی)

۱) علی پدر حسین است .
۲) اگر علی پدر حسین و حسن پدر رضا باشد آنگاه علی پدربزرگ رضا می باشد.