معرفی زبان های برنامه نویسی

بسم الله الرحمن الرحیم

درس طراحی و پیاده سازی زبان ها

موضوع پروژه:

معرفی انواع زبان های برنامه نویسی

کاری از :

سیوان گنجی

بهار ۹۲

فهرست منابع:

زبان های که برای معرفی انتخاب نموده ام عبارتند از :

۱. معرفی زبان برنامه نویسی Delphi
۲. معرفی زبان برنامه نویسی Fortran
۳. معرفی زبان برنامه نویسی C++
۴. معرفی زبان Basic
۵. معرفی زبان Visual Basic
۶. معرفی C#
۷. معرفی PhP
۸. معرفی Java

معرفی زبان های برنامه نویسی

.زبان Delphi

تاریخچه و معرفی:

دِلفی (Delphi) یا به تعبیری ویژوآل پاسکال – یک زبان برنامه‌نویسی است و بستری برای توسعه نرم‌افزار که شرکت بورلند آن را تولید کرده است . این زبان، در بد ر انتشار خود در سال ۱۹۹۵، به عنوان یکی از نخستین ابزارهایی مطرح شد که از توسعه نرم‌افزار بر مبنای متدولوژی RAD(Rapid Application Development)پشتیبانی می کردند؛ یعنی تولید و توسعه سریع برنامه‌های کاربردی این نرم افزار بر مبنای پاسکال شی گرا بوده و از این زبان مشتق شده است . البته بورلند نسخه ای از دلفی و سی‌پلاس‌پلاس‌بیلدر را برای لینوکس به نام کایلیکس (Kylix) ارائه کرد که مورد استقبال توسعه دهندگان نرم‌افزارهای لینوکس قرار نگرفت. نرم‌افزارهای دلفی در ابتدا به صورت مستقیم از کتابخانه های ویندوز و کتابخانه مخصوص خود به نام VCL استفاده می کرد، اما پس از نسخه ۶ دلفی، امکانات استفاده از دات نت هم به آن اضافه شد . در حال حاضر می توان دلفی را یکی از رایج‌ترین زبان‌های ممکن در ایران دانست .زبانِ دلفی که پیشتر بنام پاسکال شیءگرا (Object-Pascal) خوانده می‌شد و برای طراحی نرم‌افزاهای تحت ویندوز به کار می رفت، امروزه چنان توسعه یافته است که برای تولید نرم افزارهای تحت سیستم عاملِ لینوکس و دات‌نت نیز به کار می آید. بیشترین کاربرد دلفی در طراحی برنامه های رومیزی و پایگاه داده‌ها است، اما به عنوان یک ابزارِ «چند- منظوره»، برای طراحی انواع گوناگونی از پروژه‌های نرم‌افزاری نیز مورد استفاده قرار می گیرد.

دلفی در واقع یک کامپایلر پاسکال است. دلفی ۶ نسل جدید کامپایلر های پاسکال است که شرکت Borland از زمان ایجاد اولین نسخه پاسکال توسط Andres Hejlsberg در ۱۵ سال پیش به بازار عرضه کرد .

برنامه نویسی به زبان پاسکال در سالیان سال از استواری و ثبات، زیبایی و ظرافت و البته سرعت بالای کامپایل سود برده است. دلفی هم از این قاعده مستثنی نیست. کامپایلر دلفی ترکیبی از بیش از یک دهه تجربه طراحی کامپایلر پاسکال و معماری بهبود یافته کامپایلر های ۳۲ بیتی است. اگرچه قابلیت های کامپایلرها با گذشت زمان پیشرفت قابل توجهی داشته است ولی سرعت آن چندان کاهش نیافته و همچنان از سرعت بالایی برخوردار است. به علاوه استحکام و قدرت کامپایلر دلفی معیاری برای سنجش دیگر کامپایلر هاست .در اینجا به بررسی تفصیلی روند حرکتی دلفی در هر یک از نسخه های آن می پردازیم و مشخصات مهم آن را بررسی می کنیم .

تاریخچه زبان دلفی

معرفی زبان های برنامه نویسی

دلفی در واقع یک کامپایلر پاسکال است. دلفی ۶ نسل جدید کامپایلر های پاسکال است که شرکت Borland از زمان ایجاد اولین نسخه پاسکال توسط Andres Hejlsberg در ۱۵ سال پیش به بازار عرضه کرد .

برنامه نویسی به زبان پاسکال در سالیان سال از استواری و ثبات، زیبایی و ظرافت و البته سرعت بالای کامپایل سود برده است. دلفی هم از این قاعده مستثنی نیست. کامپایلر دلفی ترکیبی از بیش از یک دهه تجربه طراحی کامپایلر پاسکال و معماری بهبود یافته کامپایلر های ۳۲ بیتی است. اگرچه قابلیت های کامپایلرها با گذشت زمان پیشرفت قابل توجهی داشته است ولی سرعت آن چندان کاهش نیافته و همچنان از سرعت بالایی برخوردار است. به علاوه استحکام و قدرت کامپایلر دلفی معیاری برای سنجش دیگر کامپایلر هاست .در اینجا به بررسی تفصیلی روند حرکتی دلفی در هر یک از نسخه های آن می پردازیم و مشخصات مهم آن را بررسی می کنیم .

سال ۱۹۹۵ Delphi1

در زمان استفاده از سیستم عامل DOS برنامه نویسان مجبور بودند از بین زبان پر قدرت ولی کم سرعت Basic و زبان کارآمد ولی پیچیده و نامفهوم Assembly یکی را انتخاب کنند. پاسکال با ارائه یک زبان ساخت یافته و یک کامپایلر سریع و کم نقص این شکاف را پرکرد. برنامه نویسان Windows 3.1 هم با تصمیم گیری مشابهی رو برو شدند. یکی زبان قدرتمند و سنگین C++ و یکی زبان ساده و محدود کننده . Visual Basic

ارائه Delphi1 در این مورد هم راه حل خوبی برای برنامه نویسان بود. دلفی مجموعه متفاوتی برای برنامه نویسی بود . طراحی و توسعه برنامه های کاربردی، ایجاد DLL ها، پایگاههای داده و ... که یک محیط ویژوال وسیع را تشکیل می داد Delphi1 . اولین ابزار برنامه نویسی ویندوز بود که محیط طراحی ویژوال، کامپایلر بهینه کد برنامه و دسترسی قوی به پایگاههای داده را در یک جا جمع کرد که آن را به یکی از بهترین ابزارهای روش نوین توسعه سریع نرم افزار (Rapid Application Development)تبدیل کرد. این مجموعه قدرتمند باعث شد که در همان زمان بسیاری از برنامه نویسان زبانهای دیگر به Delphiروی بیاورند و این موفقیت بزرگی برای Borland به حساب می آمد. همچنین بسیاری از برنامه نویسان پاسکال دلفی را ابزاری یافتند که توسط آن هم از توانایی و تجربه خود در برنامه نویسی پاسکال استفاده می کردند و هم توانایی کار در ویندوز را به دست آوردند. همچنین زبانی که در آن زمان با نام پاسکال شیئی (ObjectPascal) در دانشگاهها ایجاد شده بود یک زبان بسیار خشک و محدود کننده بود که اصلاً حالت کاربردی پیدا نکرد .

ویژگیهای دلفی مثل طراحی ظاهری حساب شده و کاربر پسند آن باعث شد که زبان پاسکال شیئی عملاً از رده خارج شود. تیم طراحی VB در Microsoft قبل از حضور دلفی هیچ رقیب مهمی برای خود نمی دید VisualBasic .در آن زمان زبانی نا کارآ ، کم سرعت و کند ذهن بود Visual Basic 3 .در عمل اصلا توانایی رقابت با Delphi 1 را نداشت. در این سال شرکت Borland گرفتار یک سری مشکلات قضائی با شرکت Lotus بود که در نهایت هم متخلف شناخته شد. همچنین درگیری مشابهی هم با Microsoft بر سر تلاش در تغییر دادن فضای نرم افزار های Microsoft پیدا کرد. همچنین Borlandمشغول طراحی و فروش طرح Quatro به شرکت Novell و طراحی پایگاه های داده dBase و Paradox

بود که با استقبال قابل توجهی مواجه نشد .

در این زمان که Borland مشغول فعالیتهای قضایی و تجاری بود Microsft توانست گوی سبقت را از Borland برباید و قسمت اعظم بازار ابزار های برنامه نویسی تحت Windows را در اختیار بگیرد و سعی می کرد تا این طرز فکر را اشاعه دهد که چون Windows را طراحی کرده صلاحیت و توانایی تهیه بهترین ابزار های برنامه نویسی تحت آن را نیز در دست دارد. در این شرایط Borland با عرضه Delphi و نسخه جدید Borland C++ سعی کرد خدشه ای در فرمانروایی Microsoft وارد کند و سهمی در بازار بزرگ این محصولات داشته باشد .

سال ۱۹۹۶ Delphi2

یک سال بعد Delphi2 تمام مزایای نسخه قبلی را تحت سیستم های جدید ۳۲ بیتی (Windows 95,Windows NT) ارائه داد. همچنین Delphi2 با ارائه خصوصیات اضافه و کارکرد های قویتری نسبت به Delphi1 توانایی های خود را افزایش داد. (ازجمله ارائه کامپایلر ۳۲ بیتی که سرعت بالایی به نرم افزار ها می بخشید، کتابخانه بزرگ و کاملی از اشیای مختلف، شیوه جدید و تکامل یافته ای برای اتصال به پایگاه های داده مختلف، ادیتور پیشرفته، پشتیبانی از OLE ، توانایی وراثت در فرمهای ویژوال و سازگاری با پروژه های 16 بیتی Delphi2 .(Delphi1 به معیاری برای سنجش و مقایسه همه ابزارهای توسعه نرم افزار در آن زمان تبدیل شد .

در آن زمان با ارائه سیستم ۳۲ بیتیWindows95 جهش بزرگی در سیستم عامل Windows رخ داد و Borland بسیار مشتاق بود که Delphi را به بهترین ابزار برنامه نویسی سیستم جدید تبدیل کند. نکته این که در آن زمان به منظور تاثیر در افکار عمومی و تاکید بر قدرت Delphi در سیستم عامل ۳۲ بیتی قرار بود که نرم افزار با نام جدید Delphi32 به بازار عرضه شود ولی در آخرین مراحل به خاطر اینکه نشان دهند این زبان زبانی رشد یافته و تکامل یافته نسخه قبلی یعنی Delphi1 است نام Delphi2را برای آن انتخاب کردند .

Microsoftتلاش کرد که با Visual Basic 4 با Delphi مقابله کند ولی از ابتدا کیفیت پایین آن و ضعف آن در انتقال برنامه های ۱۶ بیتی به سیستم ۳۲ بیتی و بروز اشکالات ساختاری در طراحی آن موجب شکست زودهنگام Visual Basic 4 شد. در این زمان هنوز تعداد زیادی از برنامه نویسان به Visual Basic وفادار بودند .Borland هنچنین روشها و ابزارهای قدرتمندی همچون PowerBuilder برای طراحی نرم افزار های Client/Server ارائه داد ولی Delphi هنوز آن قدر قدرتمند نشده بود که بتواند نرم افزارهایی که جایی در بین توسعه گران پیدا کرده اند را براندازد .

سال ۱۹۹۷ Delphi3

از زمان تهیه و توسعه Delphi1 تیم توسعه Delphi در فکر گسترش و ایجاد یک زبان قدرتمند جهانی بود. برای Delphi2 این تیم تمام نیروی خود را صرف اعمال مربوط به انتقال تواناییها و کارکرد ها به سیستم ۳۲ بیتی و همچنین اضافه کردن خصوصیات Client/Server و پایگاه داده کرد .در زمان تهیه Delphi3 تیم توسعه فرصت لازم برای گسترش مجموعه

معرفی زبان های برنامه نویسی

ابزار موجود را یافت و در این راستا کیفیت و کمیت ابزارهای Delphi بهبود یافت. به علاوه راه حل هایی برای مشکلات عمده و قدیمی برنامه نویسان تحت ویندوز ارائه شد به ویژه استفاده از برخی فناوری های پیچیده و نا مفهوم (مثل COM و ActiveX وتوسعه نرم افزار های تحت Web وکنترل پایگاههای داده چند کاربره). روش نمایش کد برنامه همچنین توانایی کامل کردن خودکار کد (Code Completion) عملیات کد نویسی را راحت تر کرد. ضمن این که همچنان در بیشتر موارد اساس و متدولوژی برنامه نویسی مانند Delphi1 بود و بر پایبندی به قوانین اصولی Pascal تاکید می شد .

در این زمان رقابت شرکت های تولید کننده ابزار های برنامه نویسی بسیار تنگاتنگ شده بود Microsoft .با ارائه Visual Basic 5به پیشرفت های خوبی دست یافت ازجمله پشتیبانی قوی از COM و ActiveX و ایجاد برخی خصوصیات و تغییرات کلیدی و اساسی در کامپایلر VB. ضمن این در همین سال Borland با پشتوانه قوی Delphi و با استفاده از ساختار موفق آن ابزارهای دیگری همچون Forte و BC++ Builder به بازار عرضه کرد .

تیم Delphi در زمان طراحی Delphi3 چند تن از اعضای کلیدی خود را از دست داد Andres Hejlsberg .معمار اصلی Delphi در اقدام غیر منتظره ای Borland را ترک کرد و تصمیم گرفت به رقیب دیرینه یعنی Microsoft بپیوندد. اما حرکت تیم Delphi متوقف نشد و معاون Hejlsberg که سالها تجربه همکاری با او را داشت توانست رهبری این تیم را به خوبی در دست بگیرد. همچنین مسئول فنی تیم (Paul Gross) هم در اقدام مشابهی به گروه Microsoft ملحق شد. این تغییرات بیشتر به خاطر اختلافات شخصی بین افراد تیم بود و نه به خاطر مسائل حرفه ای .

سال 1998 Delphi4

Delphi4بیشتر بر روی راحتتر کردن کار با دلفی متمرکز شد .مرورگر روال ها (Module Explorer) بهبود یافت و مرور و ویرایش Unit ها را راحت تر کرد. کنترل کد و کامل کردن خودکار کلاسها این فرصت را به کاربر داد که فکر و زمان خود را روی ساختار اصلی برنامه بگذارد و در وقت صرفه جویی کند. طراحی رابط کاربر هم کاملاً عوض شد و بهبود یافت و اشکال زدا (Debugger) نیز پیشرفت قابل توجهی داشت Delphi4 .قابلیتهای برنامه نویسان را در استفاده از تکنولوژیهای چند منظوره خارجی مثل MIDAS ، DCOM، MISو Corba افزایش داد .

در این سال Delphi جایگاه خود را در رقابت با دیگران مستحکم کرده بود و کم کم به سمت دست یابی به سودآوری مالی مورد نظر خود پیش می رفت. در واقع در این زمان بود که حاصل کار سنگین چند ساله تیم نمایان می شد. بعد از سالها آزمایش Delphiشهرت و محبوبیت خاصی پیدا کرد و دیگر برنامه نویسان Delphi توانایی جدا شدن از آن را نداشتند. در این زمان Borlandبه کار سوّال برانگیزی دست زد و به منظور تبلیغ بیشتر و برتری در جنگ روانی با دیگر شرکتها نام Inprise را برای فعالیتهای تجاری خود برگزید .

ابزار های مربوط به فن آوری Corba را گسترش داد تا راه جدیدی برای سودآوری ایجاد کند. برای موفقیت در این زمینه Corbaنیاز به رابط کاربر قدرتمندی داشت که در کنار توانایی های آن کار کردن با آن نیز راحت باشد. دقیقاً همان کاری که

معرفی زبان های برنامه نویسی

در سالهای قبل در مورد COM و برنامه نویسی تحت Web انجام شده بود و به موفقیت دست یافته بود. با این وجود بنا به دلایل مختلفی این گسترش و توسعه Corba هیچ وقت تکامل و موفقیتی که مورد نظر بود را به دست نیاورد و بر خلاف تبلیغات و سرمایه گذاری های انجام شده فن آوری Corba تنها توانست نقش کوچکی در روند رو به جلوی Delphi ایفا کند .

سال ۱۹۹۹ Delphi5

Delphi5در برخی زمینه ها پیشرفت های قبلی را ادامه داده است. اولاً مسیری را که Delphi4 با اضافه کردن ویژگیهای زیادی شروع کرده بود ادامه داد Delphi4 .باعث شد کارهایی که قبلاً به صرف وقت زیادی احتیاج داشت بسیار سریعتر انجام شود Delphi .به شکل امیدوار کننده ای به برنامه نویس این امکان را می دهد که بیشتر به برنامه ای که میخواهد بنویسد توجه کند و نه به قواعد برنامه نویسی و نوشتن کد های تکراری و خسته کننده. این ویژگیهای سودمند شامل رابط کاربر بهبودیافته و سیستم اشکال زدایی (Debugger)توانمند ، امکانات برنامه نویسی تیمی و ابزار های ترجمه می شود .

ثانیا Delphi5 خصوصیات جدیدی را در بر می گیرد که توسعه برنامه های تحت وب را واقعاً راحت کرده است. این ویژگیها شامل طراح اشیای مربوط به ASP برای ساختن صفحات(Active Server Page) ، اشیایی موسوم به Internet Expressبرای پشتیبانی از XML و خصوصیات جدید MIDAS که آن را به یک ابزار همه کاره در پایگاه های داده تحت Webتبدیل کرد. در نهایت با صرف وقت ، هزینه و صبر زیاد توانست Delphi5 قدرتمند را عرضه کند. این فعالیت مدتها به طول انجامید و قبل از عرضه عمومی، Delphi5بارها در بازبینی ها و آزمایشهای داخلی قسمتهای مختلف آن تغییر کرد و بهبود یافت .

Delphi5در نیمه دوم سال ۱۹۹۹ به بازار عرضه شد و به نفوذ و تسلط بر بازار ادامه داد. در این زمان Visual Basic که کم کم به عضوی تحقیر آمیز برای Microsoft تبدیل می شد هم با پیشرفتهایی توانست در رقابت دوام بیاورد و از صحنه خارج نشود. در اقدام درست و به جایی نام Inprise دوباره به Borland بازگشت. این اقدام از سوی طرفداران و مشتریان قدیمی Borlandبا استقبال خوبی مواجه شد .

سال ۲۰۰۱ Delphi6

در هنگام تهیه Delphi6 ساختار Delphi در زمینه های مختلف شکل گرفته بود و به یک تکامل نسبی رسیده بود. این مسئله باعث شد که تیم طراحی بتواند وقت خود را بر روی طرحی که مدتها تنها در حد یک نظریه بود بگذارد و آن را بسیار زودتر از آن که انتظار می رفت عملی کند: گام نهادن به محیط های فراتر از Windows . بیشتر نیروی توسعه گران Delphi در این مدت صرف رهانیدن Delphi از بند Windows شد که این خود در درجه اول مبارزه ای آشکار با سلطه Microsoft بود و ثانیاً راه برنامه نویسان را به سوی فضا های دیگر برنامه نویسی باز کرد. در ابتدا این عمل ریسک بزرگی بود و بیم آن می رفت که جایگاه Delphi در Windows هم به خطر بیفتد ولی در نهایت به نقطه رشد و قوتی بدل شد که Delphi را به یکی از بهترین ابزار برنامه نویسی Multi Platform تبدیل کرد.تکنولوژی CLX روالهای مختلف Delphi را با) Kylix عضو

جدید خانواده Borland که در فضای Linux کار می کند) به اشتراک گذاشت و استفاده از سیستم بایت Java باعث شد که Delphi حتی از قید سخت افزار هم رها شود .

به نظر می رسد که این فعالیتها باعث ثبات Delphi در دنیای برنامه نویسان شود و نگرانی های Borland و برنامه نویسان که همیشه می ترسیدند که مبادا با ضعیف شدن Windows جایگاه خود را از دست بدهند حال به افتخار و آرامش برای آنان و نگرانی برای طرفداران Microsoft تبدیل شده است .

مدیریت حافظه در دلفی

تخصیص خودکار حافظه

وقتی شما از نوع‌های پایه (Integer ، real، wordو...) برای ایجاد متغییرهای خود استفاده می کنید، هیچ نگرانی درباره تخصیص حافظه آن وجود ندارد چون دلفی خودش آنرا تخصیص حافظه می کند و سپس آزاد میکند .

```
type
TDay = (Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday)
var
Name : String; {256 Bytes}
X, Y : Integer; {4 + 4 = 8 Bytes}
List : array [0..10] of Double; {8 * 11 = 88 Bytes}
Today : TDay; {1 Byte}
```

دراین نمونه پس از پایان برنامه، تمام حافظه تخصیص داده شده فراخوانی و آزاد می شود .

تخصیص حافظه دینامیکی

در این حالت برنامه نویس احتیاج دارد تا انباره حافظه را شخصا" تخصیص و آزاد کند .

نوع Pointer

اشاره‌گرها در دلفی می‌توانند شکل‌های مختلفی را در برگیرد. نخست، نوع اشاره‌گری که یک آدرس حافظه را برای نوع ویژه‌ای از داده، همانند صحیح، رشته و غیره نگه میدارد(Typed Pointer)

```
var
;Number : ^Integer
;Name : ^String
```

دوم، اشاره‌گرهای بدون نوع. اشاره‌گرهای بدون نوع (Untyped Pointers) خیلی به نوع معمولی خود شبیه هستند. اما محدودیت‌هایی مثل اینکه باید به نوع خاصی اشاره (Point) کند را ندارد .

```
var
;Something : Pointer
```

حال اشاره‌گر بدون نوع ما می تواند به هر نوعی از داده اشاره کند. برای تخصیص حافظه آن، از کمپلکس بیش از یک بیت استفاده میکنیم. برای مثال برنامه زیر کامپایل می شود ولی در زمان اجرا حافظه‌ای تخصیص نمیشود .

```
begin
New(Something);
Dispose(Something);
end;
```

برای تخصیص حافظه کامپایلر باید بداند که نوع داده ما برای تخصیص حافظه چیست :

```
type
IntPtr = ^Integer;

var
Something : Pointer;
begin
Something := New(IntPtr);
Integer(Something^) := 10;
Dispose(Something);
end;
```

تخصیص بلاکی از حافظه

ما می‌توانیم از اشاره به بلاک‌هایی از تخصیص حافظه در سیستم استفاده کنیم. این کار را با رویه‌های GetMem و FreeMemبرای تخصیص و آزاد سازی حافظه استفاده میکنیم.

```
var
Something : Pointer
```

```
begin
GetMem(Something, 100);
FreeMem(Something, 100);
end;
```

اشاره به حافظه از قبل تخصیص داده شده

هر دو نوع اشاره‌گرها می توانند به هر جایی از حافظه اشاره بکنند. این بدان معناست که آنها می‌توانند اشاره به فضای اشغال شده با داده‌هایی که در حال حاضر موجودند داشته باشند. این نمونه اشاره‌گر احتیاجی به تخصیص حافظه ندارد .

```
var
Something : Pointer;
MyString : PChar; // type PChar = ^Char;

begin
GetMem(Something, 100);
MyString := Something;
StrCopy(Something, 'Hello World');
FreeMem(Something, 100);
end;
```

حافظه Heap

Heapشامل قسمتی از حافظه موجود در یک برنامه است که آنرا حافظه پویا می نامیم .Heap بخشی است که در آن تخصیص و تعریف حافظه به صورت تصادفی (Random) اتفاق می‌افتد. این به آن معناست که اگر شما سه بلاک از حافظه را به طور متوالی تخصیص دهید، می توانید بعد از هر دستور آنرا از بین ببرید. مدیر Heap جزئیات را برای شما نگهداری می کند. بنابراین شما به سادگی می توانید یک حافظه جدید را با GetMem و یا بوسیله صدا زدن constructor هنگام ساختن یک شی درخواست کنید و دلفی به شما یک بلاک جدید را برخواهد گرداند .Heap یکی از سه فضای موجود در برنامه کاربردی را استفاده کرده و دوتای دیگر به صورت فضای یکپارچه (Global) و پشته قرار می گیرند .

حافظه Stack

Stackشامل قسمتی از یک بخش از حافظه موجود یک برنامه است که دینامیکی است اما برای تخصیص و آزادسازی فرامین مخصوص دارد. تخصیص Stack به صورت LIFO می باشد. این بدان معناست که آخرین حافظه شیء شما تخصیص داه

خواهد شد و سپس حذف می شود. حافظه پشته در روتین‌های نوعی استفاده می‌شود. وقتی شما یک روتین را صدا میزنید، پارامترهایش و روتین نوع آن در پشته ریخته می شود. همچنین پارامترهایی که در یک روتین تعریف میشوند، در پشته ذخیره میشوند و وقتی روتین خاتمه پیدا می کند تمام

آنها به طور خودکار از بین می رود.

دلفی 2006

شرکت بورلند در سال ۲۰۰۶ نرم‌افزار جدید خود را با ویژگیهای جدید به بازار ارائه کرد . این برنامه جدید امکان برنامه نویسی با دلفی و سی پلاس پلاس و همچنین سی شارپ را بطور هم زمان ارائه می دهد. بدین ترتیب برنامه نویسانی که با ابزارهای مختلفی کار می کنند براحتی م ی‌توانند در این محیط جدید برنامه نویسی کنند . ویژگی مهم این نگارش نسبت به نگارش ۲۰۰۵ بحث مدیریت حافظه است. در نگارش ۲۰۰۵ ضعفهائی در این زمینه وجود داشت که در این نسخه حل شده است. شرکت بورلند افتخار دارد که به عنوان اولین شرکت تولید کننده IDE زبانهای برنامه نویسی مانند دلفی و سی شارپ بیلدر و جی بیلدر(مخصوص زبان جاوا) و .... از تکنولوژی دات نت در محصولات خود استفاده کرده است.

دلفی ۲۰۰۷

در اواخر سال ۲۰۰۶ شرکت بورلند یک شرکت تابع با نام CodeGear را تاسیس کرد تا این شرکت بتواند تمام انرژی خود را صرف محیط های برنامه‌نویسی مشهور خود یعنی دلفی و C++ بیلدر و... کند. بن اسمیت نام اولین مدیر CodeGear بود. شرکت بورلند نیز فعالیتهای خود را در زمینه Application Lifecycle Management ادامه می‌دهد. اولین محصول این شرکت، CodeGear Delphi 2007 هست که بسیاری از نقائص موجود در دلفی ۲۰۰۶ از جمله سرعت پائین آن در این محصول برطرف شده است و بعد از دلفی ۷ می توان از آن به عنوان محصولی مطمئن و قابل استفاده نام برد، هر چند که دلفی ۲۰۰۶ هم محصولی کارامد هست. از دیگر محصولات CodeGear می‌توان به Delphi 2007 For PHP اشاره کرد که تحول شگرفی در رابطه با استفاده از تکنولوژی Ajax در دلفی است.

۲. زبان Fortran

معرفی و تاریخچه فرترن

معرفی زبان های برنامه نویسی

ه مانطور که در اغلب کتابهای مکانیک دیده اید، در اغلب برنامه های مهندسی از این زبان استفاده می گردد . زبان برنامه نویسی فرترن با وجود سادگی از قدرت و سرعت بالایی برخوردار است، بطوریکه می توان از آن برای نوشتن برنامه های CFD استفاده کرد. فرترن اغلب با قابلیت فو ق العاده در مورد اعداد معرفی می گردد و همانطور که از اسمش (FORmula TRANslation) پیداست، برای انجام محاسبات ریاضی در کارهای علمی خلق شده است . در گذشته این زبان دارای محدودیتهایی بوده که در نسخه های جدید این مشکلات حل شده است و این زبان به یک زبان بسیار قوی تبدیل شده است. اغلب کتابخانه های لازم برای انجام انواع محاسبات در این زبان وجود دارد . این زبان در استاندارد فرترن ۹۰ از قابلیت نوشتن برنامه های محاسبات موازی برای کامپیوتر های چند پردازنده ای پشتیبانی می کند که آینده روشنی را برای محاسبات سنگین CFD نشان می دهد. این زبان جزء اولین زبانهای خلق شده برای کامپیوتر می باشد که در IBM نوشته شده است. قابلیت محاسبات سریع با سادگی در برنامه نویسی باعث محبوبیت آن بین مهندسان و دانشمندان شده است . زبان فرترن در حال حاضر دارای قابلیت برنامه نویسی شیء گرا شده است.معمولاً از فرترن ۷۷ و ۹۰ استفاده می گردد که نسخه ۹۰ آن محبوبیت عمومی تری دارد. در این متن از استاندارد های برنامه نویسی فرترن ۹۰ استفاده می کنیم و در موارد لازم در مورد فرترن ۷۷ هم بحث خواهیم کرد.

چرا فرترن؟

زبان اصلی برنامه نویسی در کاربردهای علمی فرترن است . اغلب کدهای قدیمی با این زبان نوشته شده است . بنابراین لازم است که یک دانشجوی مهندسی با این زبان آشنایی داشته باشد. سالها پیش به نظر می رسید که با پیشرفت و محبوبیت عمومی زبانهایی مانند C زبان فرترن منسوخ گر دد، اما با گذشت سالها این زبان همچنان استفاده فراوان دارد . این ماندگاری مدیون کتابخانه ها و کدهای ۴۰ ساله این زبان است. در هر حال استانداردهای جدید این زبان قدرت زیادی به این زبان داده است . این زبان همچنان بعنوان مهمترین زبان برای مهندسان و دانشمندان بشمار می آید. فرترن برای انجام محاسبات ریاضی با سرعت و قابلیت بالا طراحی شده است . البته زبان فرترن هنوز در ایجاد محیط گرافیکی کمبود دارد و اگر بخواهید برای فرترن یک رابط گرافیکی کاربر(Graphical User Interface) بنویسید، باید خود را درگیر فراخوانی توابع ویندوز (API ها!) کنید. انتخاب راحت تر استفاده از یک زبان ساده مانند Visual Basic یا Delphi برای ایجاد رابط گرافیکی کاربر است. در این روش GUI را در این زبانها می سازیم و موتور اصلی برنامه برای کار با معادلات و انجام محاسبات را در FORTRAN می نویسیم و بعد آنرا با فرمت DLL (Dynamic Link Library) در اختیار برنامه قرار می دهیم . برنامه Ansys که بین مهندسان مکانیک محبوبیت دارد با زبان فرترن نوشته شده است.

فرترن ۹۰

معرفی زبان های برنامه نویسی

فرترن ۹۰ فراتر از یک ویرایش جدید از استاندارد فرترن می باشد و برنامه نویسی را بسیار آسانتر کرده است . این ویرایش کدهای فرترن ۷۷ را نیز پشتیبانی می کند. فرترن ۹۰ یک زبان برنامه نویسی انعطاف پذیر و قوی است, این زبان امکانات برنامه نویسی شیء گرا را داراست. امکانات دسترسی به امکانات سیستمی مانند تخصیص حافظه, استفاده از اشاره گرها و بسیاری امکانات دیگر به آن اضافه شده است؛ بعبارت دیگر فرترن ۹۰ بیشتر شبیه به C++ است تا فرترن ۷۷! ارتقاء قابلیتهای عددی, استفاده از دستورات حالت موازی (چند پردازنده ای) که یک پیشرفت شایان ذکر در برنامه نویسی علمی و نوشتن کدهای موازی می باشد . این روند در فرترن ۹۵ دنبال شده است و انتظار می رود در ویرایش های جدیدتر این زبان امکانات گسترده تری برای برنامه نویسی علمی به این زبان اضافه گردد

استانداردهای کد نویسی

برای سازگاری و خوانایی بیشتر در برنامه نویسی بهتر است که قوانین زیر را رعایت کنید.

۱- همهٔ کلمات کلیدی این زبان را با حروف بزرگ و همه چیز دیگر را با حروف کوچک بنویسید (این زبان بین حروف کوچک و بزرگ فرقی نمی داند!). البته این ساختار آنچنان مناسب نمی باشد و می تواند خوانایی برنامه را کاهش دهد و دلیل استفاده از آن قدمت این روش است. در هر حال استفاده از این روش در برنامه نویسی مدرن کار درستی نیست.

۲- از دندانه دار کردن در متن اصلی برنامه و همچنین بلوکهای دیگر استفاده کنید.

۳- اسم برنامه ها، زیر برنامه ها و توابع را در انتهای آنها ذکر کنید.

ساختار برنامه فرترن

ساختار برنامه در فرترن ۹۰ به ساختار زبانهای برنامه نویسی دیگر شباهت زیادی دارد و به شکل زیر است.

```
PROGRAM program_name
! Comment and program information
Declaration of variables and/or external functions
Program body
END PROGRAM program_name
```

معرفی زبان های برنامه نویسی

Declaration and body of user-made functions

در فرترن ۷۷، کل صفحه به ستونهایی تقسیم می گردد و هر بخش از برنامه محل مشخصی دارد. برنامه فرترن ۷۷ باید از قوانین زیر پیروی کند:

۱. تمام دستورات فرترن باید بین ستون ۷ تا ۷۲ از فایل قرار داشته باشند.
۲. فاصله خالی برای مترجم برنامه معنایی ندارد ولی برای خوانایی برنامه باید از آنها استفاده کرد!
۳. دستوراتی که در ستون ۱ تا ۵ آنها خالی است اما در ستون ۶ کاراکتری غیر از صفر قرار دارد به عنوان ادامه دستورات خط قبل به حساب می آیند (حداکثر تا ۱۹ خط مجاز است!).
۴. شماره خط دستورات باید بین ستون ۱ تا ۵ نوشته شوند و حداکثر می تواند یک عدد ۵ رقمی باشد (فرترن به شماره خط نیازی ندارد و فقط برای ارجاع از دستوراتی مانند GOTO استفاده می گردد.
۵. خوب است که حداکثر با دو دندانه در برنامه بخشهای مختلف را مشخص کنید.
۶. توضیحات برنامه در ستون اول خود C دارند.

با توجه به توضیحات بالا ساختار برنامه فرترن ۷۷ به شکل زیر است.

PROGRAM program_name

C Comment and program information

Declare of variables and/or external functions

Body of program

END PROGRAM program_name

Declaration an body of user-made functions

متغیرها

پیشنهاد می گردد که همهٔ متغیرهای برنامه در ابتدای برنامه تعریف (تعیین نوع) گردند. برای اینکه هیچ متغیری از قلم نیفتد، از دستور IMPLICIT NONE در ابتدای برنامه استفاده کنید . این دستور به مترجم برنامه خواهد گفت که اگر در برنامه به متغیر تعریف نشده ای برخورد، اعلام خطا کند. البته فرترن به تعیین نوع نیازی ندارد ! اما این کار خطاهای برنامه نویسی شما را کاهش خواند داد. زبانهای برنامه نویسی زیادی در اولین استفاده از متغیر برای آن متغیر نوعی در نظر می گیرند و نیازی نیست که برنامه نویس نوع متغیرها را مشخص کند زبان فرتر ن نیز چنین است . تعریف متغیر توسط برنامه نویس یا مترجم برنامه در برنامه نویسی حرفه ای برای مدت زیادی مورد بحث بود تا اینکه ناسا یکی از سنسورهای سفینه فضایی ونوس را بعلت یک تعریف متغیر اشتباه

معرفی زبان های برنامه نویسی

توسط مترجم برنامه (مترجم نوع دیگری را برای متغیر در نظر گرفته بود) از دست داد. بنابراین تعریف متغیر توسط برنامه نویس بعنوان یک روش مناسب انتخاب شد . خوشبختانه در بسیاری از زبانهای برنامه نویسی مدرن تعریف متغیر توسط برنامه نویس اجباری است. فرترن برای داشتن سازگاری از تعریف داخلی متغیر توسط مترجم پشتیبانی می کند

قوانین نامگذاری متغیرها :

۱. حداکثر طول نام متغیرها ۳۴ کاراکتر است.
۲. فقط امکان استفاده از حروف کوچک و بزرگ انگلیسی (A...Z, a...z)، اعداد(0,1,..,9) و کاراکتر زیر-خط (_) وجود دارد.
۳. توجه داشته باشید که بزرگی یا کوچکی حروف برای فرترن فرقی ندارد.
۴. اولین کاراکتر یک اسم باید حرف باشد.
۵. از کلمات کلیدی در نامگذاری استفاده نکنید.

انواع داده ها

شما در برنامه متغیرها را برای انواع مختلفی از داده ها بکار می برید. انواعی از داده ها که در فرترن ۷۷ پیشتیبانی می گردند به شرح زیر است:

۱. نوع INTEGER برای اعداد صحیح
۲. نوع REAL برای اعداد اعشاری (تقریباً ۸ رقم)
۳. نوع DOUBLE برای اعداد اعشاری با دقت بیشتر (تقریباً ۱۶ رقم معنی دار) [این اعداد را مضاعف می نامیم]
۴. نوع CHARACTER برای یک کاراکتر یا رشته ای از کاراکترها
۵. نوع LOGICAL برای مقادیر منطقی
۶. نوع COMPLEX برای اعداد مختلط به عنوان یک جفت با دقت REAL، اغلب توابع فرترن ۷۷ بر روی اعداد مختلط قابل استفاده هستند

ثابت های هم به همین شکل بکار می روند مثلاً 1234 یک عدد ثابت صحیح است، 1234.0 یا 1.234E3 یک ثابت اعشار و 1.234D3 عدد اعشار با دقت مضاعف است . اعداد مختلط به شکل (3.14,-1E5) نمایش داده می شوند و کاراکترها بین دو کوتیشن قرار می گیرند 'AbBa' یا 'S'. ثابتهای منطقی فقط می توانند دو مقدار .TRUE. و .FALSE. را داشته باشند (به نقاط ابتدایی و انتهایی هر یک توجه کنید).

معرفی زبان های برنامه نویسی

اگر اعداد بسیار کوچک یا بسیار بزرگ باشند، ممکن است سیستم آنها را صفر در نظر گرفته و مشکل تقسیم بر صفر در محاسبات پیش آید یا اینکه سیستم دچار سر ریز گردد. این خطاها بسیار رایج هستند و اشکال گزارش شده به سیستم بستگی خواهد داشت

تعریف متغیرها

برای تخصیص حافظه لازم به متغیرها، مترجم برنامه (Compiler) باید نا م، نوع و اندازه هر متغیر را بداند . اگر از دستور IMPILICIT NONE استفاده گردد، لازم است که تمام متغیرها تعریف گردند . در غیر اینصورت نوع متغیر با حرف اول آن مشخص می گردد.

a..h و o..z برای متغیرهای اعشار

i,j,k,l,m,n برای متغیرهای صحیح

عبارات و عملیات محاسباتی

عملگرهایی مانند + ، – ، / (همان ÷ است) و * (همان × است) را می شناسید. عملگر توان در فرترن به شکل ** است. اولویت محاسبه در عبارات

پرانتز – اگر در عبارات، پرانتز وجود داشته باشد. اول داخلی ترین پرانتز محاسبه خواهد شد و به همین ترتیب عبارات داخل پر انتز اولویت اول را دارند.

توان

ضرب و تقسیم

جمع و منها

توابع رشته ای

در توابع داخلی فرترن، توابع رشته ای نیز وجود دارد . برای مثال تابع LEN اندازه رشته می دهد، توابع CHAR و ICHAR بترتیب برای تبدیل عدد صحیح به کاراکتر و تبدیل کاراکتر به عدد صحیح به کار می روند. INDEX برای یافتن یک رشته در رشته دیگر کاربرد دارد. توابع مقایسه رشته ها مانند LGE، LGT، LLE و LLT و بسیارس از توابع دیگر وجود دارند که در صورت نیاز امکان مطالعه آنها را خواه ید داشت

ورودی و خروجی

در اغلب برنامه ها نیاز داریم که اطلاعات ورودی را از صفحه کلید یا فایل بخوانیم و اطلاعات خروجی را در صفحه نمایش نشان دهیم یا آنها را در فایل خروجی ذخیره کنیم.

ورودی و خروجی فایل

کار کردن با فایلها در فرترن بسیار ساده است . ورودی وخروجی فایل هم مانند ورودی از صفحه کلید یا خروجی به نمایشگر با دستور READ و WRITE صورت می گیرد. در این حالت بخش UNIT در این دستورات مشخص کننده نوع ورودی و خروجی است. برای ورودی از صفحه کلید و خروجی به صفحه نمایش، بخش UNIT در این دستورات برابر * قرار می گیرد . برای ورودی و خروجی از فایل، ابتدا فایل را باز می کنیم، این کار یک عدد (UNIT) به فایل اختصاص می دهد.

۳.زبان C++

آشنایی کلی با C++:

زبان برنامه‌نویسی C++ (تلفظ می شود: سی پلاس پلاس) یک زبان برنامه‌نویسی کامپیوتری عمومی با قابلیت‌های سطح بالا و سطح پایین می باشد. این زبان دارای قابلیت های کنترل نوع ایستا، نوشتار آزاد، چندمدلی، معمولا زبان ترجمه شده با پشتیبانی از برنامه‌نویسی ساخت‌یافته، برنامه‌نویسی شی‌گرا، برنامه نویسی جنریک است.

زبان c++ یک زبان سطح میانی در نظر گرفته می‌شود. این زبان دارای قابلیت زبان‌های سطح بالا و پایین بصورت همزمان است.

معرفی زبان های برنامه نویسی

زبان C++ توسط بی‌یارنه استراس‌تروپ دانمارکی در سال ۱۹۷۹ در آزمایشگاه های بل (Bell Labs) و بر مبنای زبان C ساخته شد و آن را "C با کلاس" نام‌گزاری نمودند. در سال ۱۹۸۳ به ++c تغییر نام داد. توسعه با اضافه نمودن کلاس ها و ویژگی های دیگری مانند توابع مجازی، سربارگزاری عملگرها، وراثت چندگانه، قالب توابع، و پردازش استثنا انجام شد . این زبان برنامه‌نویسی در سال ۱۹۹۸ تحت نام ISO/IEC 14882:1998 استاندارد شد. نسخه فعلی استاندارد این زبان ISO/IEC 14882:2003 است. نسخه جدیدی از استاندارد (که به صورت غیررسمی C++0x نامیده می‌شود) در دست تهیه است.

تاریخچه زبان:

استراس‌تروپ کار بر روی زبان «c با کلاس» را در سال ۱۹۷۹ آغاز کرد.ایده ساخت این زبان جدید در زمان کار بر روی تز دکترای خود به ذهن استراس‌تروپ خطور نمود. او متوجه شد که سیمولا دارای ویژگی‌های مناسب برای ساخت برنامه‌های بسیار بزرگ است اما برای استفاده عملی بسیار کند است اما BCPL با وجود سرعت بسیار زیاد برای ساخت برنامه‌های بزرگ بسیار سطح پایین است. زمانی که استراس‌تروپ کار خود را در آزمایشگاه های بل (Bell Labs) آغاز نمود با مشکل تحلیل هسته unix با توجه به محاسبات توزیع شده روبرو شده بود. با الهام‌آوری تجربیات خود در دوران دکترا، او زبان C را با استفاده از ویژگی‌های سیمولا گسترش داد. C به این دلیل انتخاب شد که C یک زبان عمومی، سریع، قابل حمل، و بصورت گسترده در حال استفاده بود. علاوه بر C و سیمولا زبان‌های دیگری مانند ALGOL 68، ADA، CLU، ML نیز بر ساختار این زبان جدید اثر گذاشت . در ابتدا ویژگی‌های کلاس، کلاس های مشتق شده، کنترل نوع قوی، توابع درون خطی، و آرگومان پیش‌فرض از طریق Cfront به C اضافه شد. اولین نسخه تجاری در سال ۱۹۸۵ ارائه شد.در سال ۱۹۸۳ نام زبان از «C با کلاس» به C++ تغییر یافت. ویژگی‌های دیگر شامل توابع مجازی، سربارگزاری عملگر و نام تابع، ارجاعات، ثوابت، کنترل حافظه توسط کاربر بصورت آزاد، کنترل نوع بهتر، و توضیحات یک خطی به صورت BCPL با استفاده از «//» نیز به آن اضافه شد . در سال ۱۹۸۵ اولین نسخه زبان برنامه‌نویسی C++ انتشار یافت و مرجع مهمی برای این زبان فراهم شد در حالی که هیچ استاندارد رسمی وجود نداشت. در سال ۱۹۸۹ ویرایش ۲.۰ از زبان C++ ارائه شد. ویژگی‌های جدیدی مانند ارث‌بری چندگانه، کلاس های انتزاعی، اعضای ایستای توابع، اعضای ثابت تابع، و اعضای حفاظت شده به آن اضافه شد . در سال ۱۹۹۰ «راهنمای مرجع C++ » منتشر شد. این کار بر روی استانداردهای بعدی شد. آخرین ویژگی‌های اضافه شده شامل موارد زیر بودند: قالب توابع، استثناها، فضاهای نام، تبدیلات جدید، و یک نوع داده منطقی.در حین تکامل C++ کتابخانه استاندارد نیز بوجود آمد. اولین نسخه کتاب استاندارد شامل کتابخانه جریانات I/O بود که جایگزین printf و scanf شد. در ادامه مهم‌ترین ویژگی اضافه شده Standard Template Library بوده است.

استاندارد زبان:

بعد از سال‌ها کار کمیته مشترک ISO–ANSI در سال ۱۹۹۸ C++ را استاندارد نمودند (ISO/IEC 14882:1998). به مدت چند سال پس از انتشار استاندارد این کمیته گزارشات معایب را مورد بررسی قرار داده نسخه اصلاح شده استاندارد C++ منتشر شد. در سال ۲۰۰۵ گزارشی

معرفی زبان های برنامه نویسی

فری بنام «گزارش فنی کتابخانه ۱» (که معمولا بصورت اختصار TR1 خوانده می‌شود) انتشار یافت. با وجود این که گزارش بخشی رسمی از استاندارد نیست ولی بخش‌هایی را به آن اضافه نموده که انتظار م ی‌رود در نسخه های بعدی استاندارد در نظر گرفته شود . پشتیبانی از این گزارش در حال افزایش بین تمام کامپا یلرهای فعلی است.در حالی که ++C به ه یچ موسسه‌ای وابسته نیست این مستندات بصورت آزادانه در دسترس نیستند.

نام ++C:

این نام منسوب به ریک ماسکیتی (اواسط ۱۹۸۳) است و برای اولین بار در دسامبر سال ۱۹۸۳ به کار برده شد. در طول مدت تحقیق این زبان بنام «C جدید» و بعدها «C با کلاس» خوانده شد. در علوم کامپیوتر هنوز هم ++C به عنوان ابرساختار C شناخته می‌شود. آخرین نام از عملگر ++ در زبان C (که برای افزایش مقدار متغیر به اندازه یک واحد بکار می‌رود) و یک عرف معمول برای نشان دادن افزایش قابلیت‌ها توسط + ناشی گشته است. با توجه به نقل قولی از استراس تروپ: «این نام وی‌ژگی‌ها تکاملی زبان در C را نشان می‌دهد.» +C نام زبانی غیرمرتبط به این زبان است.استراس‌تروپ مبدا این نام را در فصل اول کتاب خود «زبان برنامه‌نویسی ++C» اشاره می‌نماید که معنی دیگر ++C را می‌توان در ضمائم کتاب جرج ارول بنام ۱۹۸۴ یافت. در سه قسمت از زبان تخ یلی Newspeak «کلمات C» برای اشاره به لغات فنی و حرفه‌ای بکار می‌رود. «دو علامت +» برای ایجاد صفات عالی از صفات Newspeak به کار می‌رفت بنابراین ++C به معنای زبانی با بیشترین شباهت به C است.وقتی که به صورت خصوصی از ریک ماسکیتی در مورد این اسم سوال شد او در جواب گفت که ا ین اسم بصورت خودمانی در بین آنها به کار می‌رفته است و تصور نمی‌کردند که این نام بصورت نام رسمی این زبان درآید.

توسعه آینده

++C همچنان در حال تکامل است تا نیازهای آینده را پاسخگو باشد. نسخه جدید استاندارد ++C در حال بررسی است و تحت عنوان C++0x است که انتظار م ی‌رود در سال ۲۰۱۰ منتشر گردد. تغییرات کنونی نشان می‌دهد که که همچنان به صورت چندمدلی ++C تاکید می‌گردد. توسعه‌های مهم پشتیبانی از چندرشته‌ای و مفاهیمی برای راحت نمودن کار با قالب‌هاست. اضافه نمودن وی‌ژگی جمع‌آوری زباله به آن به شدت مورد بحث است. Boost.org گروه‌ی برای بیشترین استفاده از وی‌ژگی‌های فعلی ++C می‌باشد. آنها وی‌ژگی‌های تابعی و فرابرنامه‌نویسی آن را گسترش می‌دهند و در مورد ++C به کمیته استاندارد نصیحت‌هایی نموده است که کدام وی‌ژگی‌ها خوب عمل نمی‌کنند و کدام‌ها نیاز به توسعه دارند.

فلسفه

در کتاب «طراحی و تکامل ++C» استراستروپ قوانین مورد استفاده در طراح ی ++C را بیان می‌نماید. دانستن این قوانین به فهمیدن نحوه عملکرد ++C و چرایی آن کمک می‌کند. جزئیات بیشتر در کتاب قابل دسترسی است:++C طراحی شده است تا یک زبان عمومی با کنترل

معرفی زبان های برنامه نویسی

نوع ایستا و همانند C قابل حمل و پربازده باشد .++C طراحی شده است تا مستقیما و بصورت جامع از چندین شیوه برنامه‌نویسی (برنامه‌نویسی ساخت‌یافته، برنامه‌نویسی شی‌گرا، انتزاع داده، و برنامه‌نویسی جنریک)

++C طراحی شده است تا به برنامه‌نویس امکان انتخاب دهد حتی اگر این انتخاب اشتباه باشد.

++C طراحی شده است تا حداکثر تطابق با C وجود داشته باشد و یک انتقال راحت از C را ممکن سازد.

++C از بکاربردن ویژگی‌های خاص که مانع از عمومی شدن است خودداری می‌نماید.

++C از ویژگی‌هایی که بکار برده نمی‌شوند استفاده نمی‌کند.

++C طراحی شده است تا بدون یک محیط پیچیده عمل نماید.

کتابخانه استاندارد

در سال ۱۹۹۸ استاندارد ++C شامل دو بخش هسته زبان و کتابخانه استاندارد ++C است. این کتابخانه شامل بیشتر بخش‌های STL و کتابخانه استاندارد C است. بیشتر کتابخانه‌های ++C در استاندارد وجود ندارند و با استفاده از تعریف قابلیت پیوند کتابخانه‌ها را می‌توان در زبان‌هایی مانند فرترن، C، پاسکال، بیسیک نوشته شوند. البته با توجه به ویژگی‌های کامپایلر مشخص خواهد شد که کدام زبان را می‌توان استفاده نمود.

کتابخانه استاندارد ++C شامل کتابخانه استاندارد C با یک سری تغییرات برای بهبود عملکرد است . بخش بزرگ بعد ی این کتابخانه STL است. STL شامل ابزار بسیار قدرتمندی مانند نگه‌دارنده‌ها (مانند vector و list)، تکرارکننده‌ها (اشاره‌گرهای عمومی شده) برای شیءسازی دسترسی مانند آرایه الگوریتم‌هایی برای جستجو و مرتب سازی در آنها وجود دارند . نقشه‌ها (نقشه‌های چندگانه) (آرایه شرکت‌پذیر) و مجموعه‌ها (مجموعه‌های چندگانه) واسط‌های عمومی فراهم می‌سازند. در نتیجه با استفاده از قالب تابع، الگوریتم‌های جنریک با هر نگه‌دارنده و دارای تکرارکننده عمل نماید. همانند C ویژگی‌های کتابخانه را می‌توان با استفاده از شبه دستور #include شامل یک سرآیند استاندارد اضافه نمود . c دارای ۶۹ کتابخانه استاندارد است که ۱۹ تا از آنها نامناسب تشخیص داده شده‌اند.استفاده از کتابخانه استاندارد – مانند std::vector یا std::string به جای آرایه‌های C– موجب ایجاد برنامه‌های مطمئن‌تر شده است.STL در آغاز محصولی جداگانه از HP و سپس SGL پیش از ادغام در کتابخانه استاندارد ++C بوده است. استاندارد عبارت STL را بکار نمی‌برد بلکه آن را بخشی از کتابخانه می‌داند اما مردم هنوز هم آن را برا ی جداسازی بخش‌های مختلف کتابخانه با ا ین نام بکار م ی‌برند.(جریان‌های ورودی/خروجی، جهانی‌سازی، تشخیص، زی‌مجموعه کتابخانه C)

بیشتر کامپایلرها کتابخانه استاندارد و STL را پیاده سازی می‌نمایند. پیاده‌سازی‌های مستقلی نیز همانند STLport نیز وجود دارند. پروژه‌های دیگر نیز پیاده‌سازی‌های خود را از STL با توجه به اهداف خود بوجود می‌آورند.

ویژگی‌های معرفی شده در ++C

معرفی زبان های برنامه نویسی

در مقایسه با C زبان ++C ویژگی‌های جدیدی را معرفی نموده است مانند تعریف متغیری به عنوان عبارت، تعریف نوع‌های همانند تابع، new/delete، نوع داده bool، توابع درون خطی، آرگومان پیش‌فرض، گرانبارسازی عملگر و تابع، فضای نام و عملگر تعریف حوزه ::، کلاس‌ها (شامل تمام ویژگی‌های مربوط به کلاس ها همانند وراثت، اعضای تابع، توابع مجاز ی، کلاس‌های انتزاعی، و سازنده ها)، قالب ها، پردازش استثنا، کنترل نوع زمان اجرا، عملگرهای سربار شده ورودی (<<) و خروجی (>>).

برخلاف باور عموم ++C نوع داده ثابت را معرفی ننموده است. کلمه const کمی پیش از استفاده از این کلمه در ++C توسط زبان C بصورت رسمی بکار گرفته شد.در بعضی حالات ++C تعداد کنترل نوع بیشتری نسبت به زبان C انجام می‌دهد. (برای اطلاعات بیشتر بخش «ناهماهنگی با C» را در پایین بخوانید)توضیحات با استفاده از // قبل از زبان C در زبان BCPL معرفی شده بود که مجددا در زبان ++C به کار گرفته شد.

بعضی ویژگی‌های ++C بعدا توسط C به کار گرفته شد مانند نحوه تعریف for، توضیحات به شکل ++C (با استفاده از //)، و کلمه inline با وجود اینکه تعریف این کلمه در C99 با تعریف آن در زبان ++C هماهنگی ندارد. همچنین در C99 ویژگی‌هایی معرفی شده است که در ++C وجود ندارند مانند ماکروهای قابل تغییر و استفاده بهتر از آرایه‌ها به عنوان آرگومان. بعضی کامپایلرها این ویژگی‌ها را پیاده نموده‌اند اما در بقیه این ویژگی‌ها موجب ناهماهنگی می گردد.

پیش پردازنده

++C بطور عمومی در سه فاز ترجمه می‌گردد: پیش پردازنده، ترجمه به کد object، پیوند (که دو مرحله آخر به عنوان عمل کامپایل شناخته می‌شود.) در اولین مرحله در پیش پردازنده، شبه‌دستورات پیش پردازنده تغییرات لغوی بر روی کد منبع ایجاد می‌نمایند و آن را به به مراحل دیگر تحویل می‌دهند.

شبه دستورات پیش پردازنده با استفاده از کاراکتر # قبل از هر گونه فضای خالی آغاز گشته و رشته هایی را در کد منبع با فایل یا رشته‌های دیگر با توجه به قوانین تعریف گشته توسط برنامه نویس جایگزین می‌نمایند. این دستورات معمولا اعمال زیر را انجام می‌دهند: جایگزینی ماکروها، شمول فایل های دیگر (برخلاف ویژگی سطح بالاتر مانند شمول ماجول ها/پکیج ها/پروژه ها/کامپونت ها)، کامپایل شرطی و/یا شمول شرطی. به عنوان مثال:

```
#include <iostream>
```

که این دستور تمام سمبل‌ها در فایل سرایند کتابخانه استاندارد iostream را در فایل منبع وارد می‌سازد.

کاربرد معمول دیگر به عنوان ماکرو خوانده می‌شود:

```
#define MY_ASSERT(x) assert(x)
```

که کد MY_ASSERT(x) را با assert(x) در فایل منبع جایگزین می‌نمایند. که این جایگزینی امکان کنترل استفاده از این تابع را در اختیار برنامه‌نویس قرار می‌دهد.استفاده از ماکروها در عمل چندان توصیه نمی‌گردد چرا که امکان کنترل نوع آرگومان ها را از بین برده در نتیجه

ممکن است اشتباهاتی را وارد کد منبع نمایند. طریقه دیگر برای انجام این کار استفاده از توابع درون خطی است.علاوه بر شبه‌دستورات معمول تعدادی شبه دستور برای کنترل جریان کامپایل وجود دارد که امکان شمول یا عدم‌شمول قطعه‌ای کد یا سایر ویژگی‌های کامپایل را در اختیار ما قرار می‌دهد.دستورات پیش پردازنده برای کاربردهای عددی نیز به کار می‌رود که هم‌اکنون استفاده از const به جای #define ترجیح داده می‌شود. این کار علاوه بر ایجاد کنترل نوع قوی مانع از گمراه شدن در فضاهای نام می‌گردد.هدف کمیته استانداردسازی از بین بردن پیش پردازنده است اما با توجه به خصوصیت مدولار ++C بعید به نظر می‌آید که این حذف امکان‌پذیر باشد.

قالب‌ها

قالب‌ها متفاوت با ماکروها هستند . در حالی که هر دوی این ویژگی‌های زمان کامپایل برای ایجاد کامپایل شرطی استفاده می‌شوند قالب‌ها محدود هستند به تعریف لغوی و متنی نیستند. قالب‌ها با آگاهی از معنا و سیستم نوع در زبان استفاده شده و سایر ویژگی‌های زمان کامپایل می‌توانند از عملیات سطح بالا برای کنترل ترتیب اجرا براساس نوع پارامترها استفاده نمایند. ماکروها کنترل خود را بر کامپایل از طریق ویژگی‌های از پیش تعیین شده انجام می‌دهند ولی قادر به ایجاد انواع جدید و کنترل نوع نیستند و فقط محدود به تعریف متنی پیش از کامپایل هستند. به زبان دیگر ماکروها کنترل خود را با استفاده از نشانه های از پیش تعیین شده انجام می‌دهند اما همانند قالب ها نمی‌توانند نشانه‌ها را خود ایجاد نمایند. قالب‌ها ابزاری برای چندریختی ایستا و برنامه‌نویسی جنریک است. علاوه بر این قالب‌ها یک ویژگی تورینگ-کامل هستند که به این معناست که هر برنامه قابل محاسبه توسط کامپیوتر را می‌توان با استفاده از فرابرنامه نویسی قالب‌ها نوشت.بطور خلاصه استفاده از قالب‌ها به معنای نوشتن هر تابع یا کلاس باشتفاده از تمامی انواع ممکن است که در قالب آن را پیش از کامپایل معین نمی‌کنیم.

اشیاء

سی++ چندین ویژگی شی‌گرا را زبان سی معرفی نمود معرفی کلاس چهار ویژگی که در زبان های شی‌گرا و بعضا غیر شی‌گرا حضور دارد یعنی انتزاع، بسته بندی، وراثت، و چندریختی را فراهم کرد . اشیاء نمونه‌های ساخته شده از کلاس در زمان اجرا هستند . می‌توان کلاس را نمونه‌ای از قالب ها دانست که چندین مورد از آنها بوجود می‌آید.

بسته‌بندی

بسته‌بندی به معنای جمع‌آوری عملیات و داده در یک محل می‌باشد. سی++ بسته‌بندی را با ایجاد امکان تعریف هر کلاس به صورت public، private، protected پیچیده‌سازی نموده است. اعضای private فقط توسط اعضای کلاس و یا کلاس‌ها دقیقا پیش شده (friend) قابل دسترسی هستند. اعضای protected توسط کلاس های ارث برده شده و اعضای کلاس و کلاس های friend قابل دسترسی هستند.در تعاریف شی‌گرا باید تنها توابعی بسته‌بندی گردند که باید از نحوه پیچیده سازی این نوع بخصوص اطلاع داشته باشد . سی++ این ویژگی را با استفاده از توابع عضو و توابع دوست فراهم نموده اما قطعی نکرده است. در سی++ این امکان وجود دارد که تمام نوع را عمومی تعریف نمایند اما در صورتی که نیاز باشد فقط بخشی از آن عمومی گردد در نتیجه این زبان نه تنها شی‌گرا است از مدل های ضعیف تر همانند برنامه‌نویسی

مدولار پشتیبانی می‌نماید.عموما توصیه بر این است که تمام اعضا به صورت خصوصی یا حفاظت شده تبدیل گردند و فقط توابعی که باید توسط دیگر کلاس‌ها به عنوان واسط استفاده شوند عمومی باقی بمانند.

وراثت

وراثت این امکان را ایجاد می‌کند که یک نوع ویژگی دیگر انواع را داشته باشد . وراثت از یک کلاس پایه می‌تواند عمومی، خصوصی یا حفاظت شده باشد. این تعیین سطح دسترسی مشخص می‌سازد آیا کلاس‌های نامربوط و یا مشتق شده می‌توانند به اعضای عمومی یا حفاظت شده کلاس پایه دسترسی داشته باشند. تنها وراثت عمومی به معنای وراثت به کار رفته بصورت عموم است . دو نوع دیگر وراثت به ندرت مورد استفاده قرار می‌گیرند. اگر تعیین کننده سطح دسترسی حذف شود سطح دسترسی برای کلاس خصوصی و برای ساختمان به صورت عمومی تعریف می‌گردد. کلاس‌های پایه ممکن است بصورت مجازی تعریف شوند که به آن وراثت مجازی گویند. وراثت مجازی تضمین می‌کند که فقط یک نمونه از کلاس پایه وجود داشته باشد و مشکلاتی همانند مشکلات وراثت چندگانه بوجود نیاید.وراثت چندگانه یکی از ویژگی‌های مورد بحث در سی++ است. وراثت چندگانه امکان اشتقاق از چند کلاس پایه را فراهم می‌نماید که موجب بوجود آمدن گراف رابطه وراثت بسیار پیچیده است. به عنوان مثال «گربه پرنده» می‌تواند از کلاس «گربه» و کلاس «پستانداران پرنده» ارث برد. در زبان‌های دیگر مانند سی‌شارپ و جاوا به صورت دیگری ویژگی مشابه را پیاده‌سازی می‌نماید هر کلاس می‌تواند از چندین واسط اشتقاق یابد اما فقط یک کلاس پایه برای اشتقاق وجود دارد(واسط‌ها برخلاف کلاس پایه فقط تعریف هستند و هیچ‌گونه پیاده‌سازی را شامل نمی‌گردند).

چندریختی

امکان استفاده از یک واسط برای چندین پیاده‌سازی فراهم می‌نماید و اشیاء در شرایط مختلف رفتار مختلفی از خود نشان می‌دهند.سی++ دو نوع چندریختی در اختیار برنامه‌نویس قرار می‌دهد: چندریختی زمان کامپایل و چندریختی زمان اجرا . چندریختی زمان کامپایل امکان تصمیم‌گیری‌های زمان اجرا را فراهم نمی‌سازد و چندریختی زمان اجرا اغلب موجب پایین آمدن بازده‌ی می‌گردد.

چندریختی ایستا

چندریختی ایستا شامل گرانبارسازی تابع، گرانبارسازی عملگر، آرگومان پیش‌فرض، و قالب کلاس‌ها و تابع است.

گرانبارسازی تابع

گرانبارسازی تابع امکان تعریف چندین تابع با نام یکسان اما با تعداد آرگومان‌های متفاوت را فراهم می‌سازد. این توابع از طریق تعداد پارامترها یا نوع رسمی آنها شناسایی می‌گردند. در نتیجه یک تابع ممکن است با توجه به موقعیت استفاده معنای مختلفی داشته باشد. نوع داده برگشتی برای تشخیص توابع از یکدیگر مورد استفاده قرار نمی‌گیرد.

معرفی زبان های برنامه نویسی

گرانبارسازی عملگر

بطور مشابه گرانبارسازی عملگر امکان استفاده از یک عملگر مشخص می‌شود که عملکرد متفاوتی با توجه به عملوندها ی خود دارد. این عملگرهای گرانبار شده موجب فراخوانی تابع مشخصی متناسب با آن موقعیت می گردند. گرانبارسازی عملگر ترتیب اجرا یا تعداد عملوندهای یک عملگر را تغییر نمی‌دهد. عملگرهای . :: .* ? نمی‌توانند گرانبار شوند.

ساختار برنامه ها

ساختار برنامه ها در این زبان بدین صورت است که همانند زبان سی، هر برنامه با بدنه ی یک تابع اصلی (main) به عنوان بدنه برنامه داشته باشد. هر برنامه معمولا از تعداد زیادی فایل تشکیل می شود که به هم الحاق می گردند (با دستور include) و به این فایل های الحاقی سرآیند (Header) می‌گوییم. فایل های الحاقی حاوی کد های نسخه های اجرایی کلاس ها (مجموعه متغیر ها و توابع) می باشند که در بدنه اصلی برنامه از آنها استفاده می شود. معمولاً هر کلاس (که تعریف یک نوع داده ای با متد های مربوط به آن است) را در یک سرآیند می‌نویسند. هر سرآیند که معمولا تنها تعاریف (معرفی) کلاس را در خود دارد به همراه فایل های پیاده سازی به زبان C++ یا پیاده سازی های کامپایل شده (به صورت فایل اشیاء مانند dll یا so یا...) می تواند به کار برده شود . به مجموعه های یکپارچه ای از کلاس های پیاده سازی شده (به صورت فایل های سرآیند با پیاده سازی های کد یا اشیاء زبان ماشین) که برای برنامه نویسی به کار می روند، یک کتابخانه C++ گفته می شود و قدرت اصلی این زبان در امکان به کارگیری کتابخانه های آماده می باشد. کتابخانه های بزرگ C++ مانند STL، MFC، QT و ... مجموعه قدرتمندی برای تولید برنامه در این زبان ایجاد کرده اند.

محیط های برنامه نویسی

یک برنامه به زبان C++ می تواند در محیط های Turbo C++ , Borland C++ و Dev C++ نوشته شود. این محیط های برنامه‌نویسی، همراه با یک کمپایلر عرضه می‌شوند که کار تبدیل برنامه به فایل اجرایی را راحت می‌کند.

معرفی زبان های برنامه نویسی

۴.زبان Basic

Basic فرزند دردانه بیل گیتس!

به جرات می توان ادعا کرد که در دنیای امروز کمتر کسی را می توان یافت که نام و آوازه «بیل گیتس» رئیس و بنیانگذار مایکروسافت و ثروتمندترین مرد جهان را نشنیده باشد. اما دانستن این موضوع جالب است که پیش از مایکروسافت نام بیل گیتس با «بیسیک» (Basic) عجین بوده است و این ماجرا به زمان دانشجویی گیتس و دوستش «پل آلن» در ۱۹۶۴ بازمی گردد و در واقع همکاری مشترک آنها در توسعه بیسیک بود که چند سال بعد منجر به تاسیس مایکروسافت شد . این دو با تولد اولین بیسیک در سال ۱۹۶۴ در کالج دارتموث به تلاش در توسعه آن همت گماشتند و در این راه با زیرکی و دوراندیشی مثال زدنی با طراحی انواع مفسرها و مترجم های بیسیک توانستند آن را به عنوان یکی از فراگیرترین و کاربردی ترین زبان های کامپیوتری تا امروز مطرح سازند. بیسیک اکنون ۴۰ ساله است و هنوز هم گیتس در مایکروسافت آن را تر و خشک می کند و مانند فرزندی دردانه به رشد و ترقی آن اهمیت می دهد .ظهور میکروکامپیوترها در سال ۱۹۷۵ از یک سو و ارائه بیسیک پیشرفته توسط مایکروسافت از سوی دیگر خیلی سریع سبب شهرت بیسیک به مثابه یک زبان کاربردی که به صورت رایگان بر روی همه کامپیوترها قابل نصب بود، گردید. در واقع بیسیک به علت ساختار مطلوبش پیش نیاز یادگیری همه زبان های کامپیوتری شد . مقاله ای که پیش رو دارید، بیل گیتس در جشن تولد ۲۵ سالگی فرزندش بیسیک (یعنی ۱۵ سال پیش) در مجله بایت منتشر کرد و طی آن برای اولین بار به شرح ماجرای شکل گیری و رشد زبان بیسیک به همراه پل آلن پرداخته است، که مقایسه داده های آن با امروز که بیسیک به سن چهل سالگی رسیده و کامپیوترهای شخصی به اوج پیشرفت و همگانی شدن نایل آمده اند، برای خواننده علاقه مند خالی از لطف نیست.از زمان اجرای اولین برنامه بر روی یک کامپیوتر انگلیسی زبان در سال ۱۹۴۸ تاکنون محاسبات وارد مرحله جدیدی شده است. فقط در طول ۱۵ سال شاهد رشد و ترقی کامپیوترهای ۸ بیتی با ۴ کیلوبایت رم به کامپیوترهای ۳۲ بیتی با ۴ مگابایت رم چه در زمینه صنعت و چه در زمینه علوم بوده ایم .با توجه به رشد بسیار گسترده در زمینه سخت افزار، زبان بیسیک نیز بیست و پنجمین سال تولد خود را پشت سر می گذارد و مفسر آن ۱۵ سال است که میکروکامپیوترهای میلیون ها نفر را قابل دسترسی کرده است.متاسفانه در ابتدا بیسیک خوب شناخته نشد، در همین حال بیشتر از هر زبان دیگری قابل دسترسی بوده و هست چرا که مجاناً بر روی هر کامپیوتری نصب می شود . قدرت و توانایی، سادگی استفاده از یک مفسر، مدیریت قوی، گستردگی و همه منظوره بودن، نوع انگلیسی لغات کلیدی و ترکیبات و آزادی بیسیک، همگی باعث تجربه بیشتر برنامه نویسان می شود و می تواند به

معرفی زبان های برنامه نویسی

ع‌نوان یک زبان ایده آل به مبتدیان در شناخت بهتر کامپیوترشان کمک کند. با توجه به سرگذشت بیسیک طی سال های طولانی و با وجود تکنولوژی قوی سیستم عاملی مانند /2OS و ویژگی برنامه نویسی موضوعی می توانید به خوبی شاهد چگونگی حضور و رشد بیسیک تا به امروز باشید.بیسیک بدون هیچ نیازی، مستقلاً به عنوان وسیله ای برای تبادل اطلاعات بین مردم و کامپیوتر ایجاد شد و در سال ۱۹۶۲ یکی از ریاضیدانان کالج دارتموث به نام پروفسور «توماس کورتس» طرحی را تسلیم «جان کمنی» رئیس کالج کرد. در این طرح تمامی دانشجویان این کالج موظف به آموختن کام پیوتر در دوره ۴ ساله تحصیلشان بودند.کامپیوترهای دسته گرای آن زمان چنین انتظاری را غیرممکن می کردند چرا که اگر برنامه ساده ای به طور صحیح ترجمه می شد کامپیوتر برای درک آن گاهی اوقات به روزها وقت نیاز داشت در نتیجه چنین برنامه ای تنها می توانست نتیجه یک محاس به را به عنوان جواب برگرداند و شخص برنامه نویس هرگز نمی توانست اجرای برنامه را ببیند . کمنی و کورتس برای گسترش تحقیقات خود به آزمایشگاه های «هیت» و «بل» رفته و در آنجا سیستم عامل چند کاربره ای را برای کامپیوتر جدیدی که قرار بود به زودی تحویل کالج شود ساختند. در همین زمان آنها به دانشجویان پیشنهاد کردند که زمان استفاده از کامپیوتر را بین خودشان تقسیم کنند اما به دلیل عدم وجود یک زبان ساده برای صحبت با ماشین دانشجویان به ندرت از کامپیوتر کالج استفاده می کردند. متاسفانه «فورترن» و «آلگول» نمی توانستند امکانات م ناسب و آسانی را برای دانشجویان فراهم کنند لذا کد و دستورالعمل نمادی همه منظوره مخصوص افراد مبتدی (بیسیک) به عنوان یک ترکیب ساده و بهتر از فورترن و آلگول توسط دانشجویان کالج دارتموث ساخته شد.در اول ماه می سال ۱۹۶۴ دانشجویان کالج دارتموث با اعلان معروف READY> در ترمینال های راه دورشان آشنا شدند . بدین ترتیب آنها می توانستند برنامه های ساده ای نوشته و آنها را برای ترجمه و اجرا انتقال دهند. کمنی و کورتس به تولد بیسیک در کتابشان به نام «برگشت به بیسیک» اشاره کرده اند. (انتشارات «ادیسون وزلی»، ۱۹۸۵)

• ظهور میکروکامپیوترها:

اولین میکروکامپیوتر با حافظه ای بسیار کوچک که بیشتر جنبه نمایشی داشت در سال ۱۹۷۵ به صحنه آمد. این کامپیوتر فقط زبان ماشین را می فهمید. در این هنگام من به اتفاق یکی از دوستانم به نام «پل آلن» فرصت را غنیمت شمرده و برگردانی از بیسیک را به منظور اجر ا در آن فضای بسیار کوچک حافظه نوشتیم . دانشجویان با استفاده از اولین بیسیک، که برای دستگاه های MITS ساخته شده بود می توانستند برنامه هایشان را بر روی این گونه کامپیوترها با حافظه ۴ کیلوبایت اجرا کنند. در آن زمان برای ما حافظه بسیار ارزشمند بود به طوری که مج بور شدیم برای اشغال فضای کمتری از حافظه، اعلان READY> - که چهار کارکتر فضا می گرفت - را به OK> - که دو کارکتر فضا می گیرد _ تبدیل کنیم. فشار محدودیت حافظه تا اندازه ای بود که ما را بر آن داشت که بیسیک را به عنوان یک مفسر پیاده سازی کنیم . (مفسر بیسیک، برنامه ای است که فایل های بیسیک را به زبان ماشین ترجمه می کند.) البته عامل دیگری که ما را به سمت مفسرها هدایت می کرد، متعادل بودن و آ سانی استفاده از مفسرها بود که به برنامه نویسی با بیسیک هنر و زیبایی خاصی می بخشید . یک برنامه نویس با استفاده از مفسر می تواند به کامپیوتر وظیفه ای را

محول کند، کامپیوتر نیز متقابلاً و بلافاصله به او جواب خواهد داد که این جواب می تواند شامل گزارشی از خطاهای احتمالی نیز باشد. این تعامل مفسر به این خاطر است که وجود آن به عنوان بخشی از زبان در نظر گرفته شده است نه به عنوان برنامه ای کاملاً مجزا مانند یک مترجم.با استفاده از تجربیاتی که از نوشتن یک مفسر بیسیک برای کامپیوتر PdP-8 در دوران دبیرستان به دست آورده بودم به اتفاق پل آلن بیسیک کامپیوتر اصلی خودمان را یک مفسر تک _ نمایش ساختیم. به این ترتیب برای ذخیره بیشتر کد مبدا به فرم یک متن مجبور شدیم آن را به طور فشرده تری ترجمه کنیم چرا که با فشار و محدودیت حافظه روبه رو بودیم . به این ترتیب ما به مقصود خود رسیدیم و ترتیبی دادیم که برنامه نویس بتواند بلافاصله برنامه اش را دیده و هنگام اجرای برنامه با سرعتی قابل قبول مراحل مختلف آن را تشخیص د هد. در مفسری که ما ساختیم از تصاویری پایین تر از یک بایت تا تصاویری بیشتر از آنچه که کدهای اسکی (ASCII) نیاز دارد برای نشانگذاری کلمات کلیدی بیسیک استفاده شده بود. همچنین برای اولین بار فرمان های کوتاه TRON _TROFF را به منظور فعال و غیرفعال کردن ابزار اشکا ل زدایی توکار بیسیک مانند توانایی ردیابی را درون آن مفسر قرار دادیم.قرار دادن کلمات رزرو شده بیسیک، پیغام های خطا و کتابخانه اعداد با ممیز شناور به منظور اجرای برنامه در یک ماشین ۴ کیلوبایتی از کارهای بسیار سختی بود که به کمی تیزهوشی و زیرکی نیاز داشت . کده‌ای استفاده شده در آن زمان که از ظرافت و انعطاف خاصی برخوردار بود هنوز برای من به یاد ماندنی و جالب است.

۶. زبان Visual Basic

visual basic چیست؟

معرفی زبان های برنامه نویسی

visual basic یک زبان برنامه نویسی تحت windows است.برنامه های به زبان visual basic در محیط برنامه نویسی IDA پیاده سازی میشوند.محیط IDA تسهیلات لازم جهت پیاده سازی وخطا یابی برنامه های visual basic رادر اختیار برنامه نویس قرار میدهد. محیط IDE منحصر به visual basic نیست و امکان توسعه برنامه در کلیه محیطهای visual را میدهد. محیط IDE امکان پیاده سازی برنامه ها را در حد اقل زمان فراهم کرده است ، ضمن انکه چنان تسهیلاتی ایجاد کرده است که برنامه های تحت windows بدون نیاز به برنامه نویس متخصص قابل پیاده سازی باشد . Visual basic زبانی است به وضوح متفاوت ازسایر زبانها که درعین سادگی امکان استفاده ازترکیبات قدرتمند مانند ACTIVEX، OOP،GUI WIN 32 API ، برنامه نویسی ساخت یافته ، کنترل رویدادها ، کنترل خطاها و بسیاری از ترکیبات قدرتمند دیگررا برای برنامه نویس فراهم کرده است.Visual basicیک زبان تفسیری است . در ویرایشهای حرفه ای و تخصصی این امکان که بتوان کدهای Visual basic را به کد های زبان ماشین تبدیل کرد وجود دارد.

تاریخچه visual basic

Visual basic از زبان basic برگرفته شده است . basic ازاواسط دهه ۱۹۶۰ توسط پرفسور "جان-کمنی" و پرفسور"توماس-کورتس" از دانشکده "دارت موس"ساخته شد و به عنوان یک زبان برنامه نویسی برای پیاد ه سازی برنامه های ساده توسعه یافت ضمن انکه هدف نهایی از ایجاد وتوسعه زبان basic (آموزش برنامه نویسی) بود.استفده متداول از basic و به کار گیری ان در هر مکان وبا هر نوع کامپیوتر ،انگیزه و علتی برای توسعه و پیشرفت این زبان بود . در اواخر دهه ۱۹۸۰« رابط گرافیکی کاربر» یا به اختصار GUI-در محیط windows ، توسعه یافت. نتیجه سازگاری basic با GUI در محیط windows بود که در سال ۱۹۹۱ میلادی توسط شرکت ماکروسافت ایجاد و به بازار عرضه شد. تا قبل از ظهور visual basic ،برنامه نویسی کار پر زحمت وطاقت فرسایی بود . visual basic ،برنامه نویسی در محیط windows رابسیار ساده کرد . از سال ۱۹۹۱ تاکنون،شش نسخه از این محصول به بازار عرضه شده است که اخرین نسخه ان۶ visual basic در سپتامبر ۱۹۹۸ انتشار یافت.

برنامه نویسی ساخت یافته

دهه ۱۹۶۰ بود که برخی از نرم افزارهای بزرگ با مشکلات سرویس دهی وخدمات مواجه شدند . به عنوان مثال پیاده سازی نرم افزارها بیش از زمانبندی پیش بینی شده زمان می برد وهزینه تولید نرم افزارها بسیار افزون بر بودجه ای می گردید که در ابتدای عمل بر اورد نشده بود . همچنین محصولات تکمیل شده قابل اعتماد نبودند . این موضوع باعث تقویت این فکر شد که تولید و توسعه نرم افزار بسیار پیچیده تراز چیزی میباشد که تصور شده است . فعالیت تحقیقاتی که در سال ۱۹۶۰ انجام شد نشان داد که برنامه نویسی ساخت یافته میتواند گره گشای مشکلات باشد .برنامه نویسی ساخت یافته یعنی بلوک بندی وایجاد نظم در نوشتن برنامه ها که باعث واضح تر شدن خطا یابی وهمچنین ساده تر شدن تغییر واصلاح برنامه ها میشود

معرفی زبان های برنامه نویسی

یکی از نتایج تحقیقات فوق ایجاد و توسعه زبان برنامه نویسی pascal توسط (نیک لاوس- ورت) در سال ۱۹۷۱ میلادی بود زبان پاسکال برگرفته شده ازنام (بلیاس- پاسکال) ریاضیدان وفیلسوف بزرگ قرن هفتهم بود که پس از ان در اکثر دانشگاه ها به عنوان زبان برنامه نویسی اصلی رحجان داده شد.

۶. زبان C#

معرفی و تاریخچه:

سی‌شارپ همچون زبان برنامه‌نویسی جاوا زبانی‌ست شئ‌گرا و سطح بالا (high level). محصول شرکت Microsoft و بر پایه .NET از آنجایی که شئ‌گرایی و سطح بالا بودن از ابزارها ی مدیریّت مؤثّر و کارآمد پیچیدگی در فضای پیچیدهٔ اینترنت مدرن می باشند، در واقع می‌شود جاوا و سی‌شارپ را از جمله زبان های اصلی برای ایجاد و انجام برنامه‌های کاربردی تحت وب (web

applications) و خدمات وب دانست.بر اساس ادّعای شرکت مایکروسافت، این زبان در سال ۲۰۰۰ توسط تیمی به سرکردگی آندرس هلزبرگ و زیر سکات ویلتاموث ساخته شد. سی‌شارپ که فقط برای دات نت است در مجموعه .NET Platform SDK ارائه گردید که در محیط های برنامه نویسی استودیوی بصری دات نت (Visual Studio .NET)، در نسخه‌های ۲۰۰۳ و ۲۰۰۵ آن موجود است. دستورات زبان سی‌شارپ مانند جاوا سطح بالا تر از C و C++ است و از VB ساده تر. این زبان همانند پیدایش زبان C انقلابی را در امر برنامه نویسی موجب شد، چراکه به طور همزمان می‌تواند امکانات سطح پایین و سطح بالا را به بهترین شکل پشتیبانی کند. در طراحی این زبان برای جلوگیری از پیدایش Bug در زمان اجرا بسیار تلاش شده و اجازه هر کاری (مانند ساخت اشاره گر) را نمی‌دهد (مگر اینکه اصرار بر انجام این کارها داشته باشید)

سکّوی دات نت:

در ماه ژوئن سال ۲۰۰۰ میلادی بود که شرکت ما یکروسافت ابداع و ایجاد سکّوی (platform) جدید برنامه نویسی خود را موسوم به دات نت (Microsoft .NET) اعلام نمود. در جهت بهبود سکّوهای پیشین مایکروسافت،دات نت مدل تازه‌ای را برای ایجاد نرم افزار ها ی کاربردی ارائه می‌دهد که در آنجا زبان ها ی گوناگون می توانند در کنار هم قرار گیرند و با یکدیگر به همکاری بپردازند. این درست مدل کوچک تری ست از تمامی فضای اینترنت بدانگونه که عوامل گوناگون می‌توانند در سطوحی گسترده به همکاری اقدام کنند.

انعطاف پذیری سی شارپ:

سی شارپ زبانی است کاملا شی گرا و بر پایه .NET Framework. این زبان مانند بسیاری از زبانهای شی گرای دیگر از فایل های کتابخانه ای Net. استفاده می‌کند و همچنین فقط بر روی سیستم عامل Microsoft Windows XP SP2(یا بعد از آن) که .NET Framework بر روی آن نصب باشد اجرا می‌شود سی شارپ به کلی برای دات نت طراحی شده و علاوه بر تولید پروژه های Console Programs در تولید Windows Application هم بسیار قوی است سی شارپ بر پایه نظری اولیه سی طراحی شده و همچنین بسیاری از صفات خود را از C++ و Java و Delphi به ارث برده است ولی به گفته مایکروسافت بر پایه C برای رقابت با Java و به سادگی VB طراحی شده است. در کل سی شارپ یک زبان مدرنیزه شی گرا می‌باشد که در محیط Visual Studio بسیار قدرت یافته است .

مثالی از سی شارپ

تمام کدهای سی شارپ بر پایه کلاس ها تعریف میشوند یک برنامه ساده سی شارپ این چنین است :

{ { public class MyClass { public MyClass() { // implementation}}

برنامه ساده ای که متنی را در خروجی چاپ میکند چنین در سی شارپ نوشته میشود

hello.cs: public class Program { public void Main() { System.Console.WirteLine("Hello , "World);}}

منبع اطلاعات در سی شارپ:

سی شارپ میتواند به خوبی با Database ها ارتباط برقرار کرده و آنها را ویرایش و یا از آنها اطلاعات بگیرد این ویژگی که در Visual Studio .NET به خوبی پشتیبانی شده کار را برای کاربر بسیار راحت کرده تا جایی که فقط با چند دستور ساده میتوان با منبع اطلاعات ارتباط برقرار کرد. در این زبان از بانک اطلاعات به صورت Connectionless استفاده میشود. به این شکل که کل بانک به داخل یک Dataset بر روی حافظه اصلی کپی میشود و سپس استفاده میشود.این روش به منظور کاهش ترافیک شبکه طراحی شده است.

فایلهای تولیدی در سی شارپ:

فایلهایی که بیشتر با کاربر سر و کار دارند عبارتند از : فایلهای با پسوند .Sln : این فایل سطح بالای Solution است که برای هر برنامه یک فایل از این نوع موجود است . هر فایل Solution یک یا چند فایل پروژه را در خود دارد. فایلهای با پسوند .Csproj: این یک فایل پروژه C# است .هر فایل پروژه یک یا چند فایل سورس دارد . فایل های سورس در یک پروژه باید به یک زبان برنامه نویسی نوشته شود . فایلهای با پسوند Cs: این یک فایل سورس است و کد برنامه در این فایل نوشته می شود و این فایل شامل کدی است که ویژوال استودیو به صورت خود کار ایجاد می کند . فایل Assemblyinfo.cs: این یک فایل سورس دیگر است با این تفاوت که می توان از این فایل برای اضافه کردن مواردی مثل اسم نویسنده و تاریخ نوشتن برنامه و امثال آن به برنامه استفاده می شود. فایل .Ico: این آیکون برنامه است .آیکونی که در زمان اجرای برنامه در نوار وظیفه قرار می گیرد.

ابزارهای دیگر در سی شارپ:

سی شارپ نه فقط مخصوص ساخت یک برنامه DOS یا Windows Form است بلکه میتوان با آن نرم افزارهای کاربردی تری مانند Movie Collection ها را به صورت سفارشی کمپایل کرد حتی قدرت بسیار بالای این زبان در ساخت Screen Saver

ها هم بسیار زبان زد است که این خود نوعی مزیت نسبت به C++ یا دیگر نرم افزارهای برنامه نویسی به شمار می‌رود ولی این زبان برای این کارهای ساده طراحی نشده است . از جمله کارها یی که این زبان می‌تواند انجام دهد طراحی نرم افزار برای SmartPhone,Pocket PC ,Windows CE, Linux است. در ضمن این زبان برای طراحی وب سایت به‌زبان ASP.Net نیز به کار برده می‌شود. همچنین برای کار با Registry و فایلها مناسبترین زبان است. در زمینه کار با بانکهای اطلاعاتی و APIها نیز بسیار قوی است.

۷.زبان PhP

PHPچیست؟

هر روز وب سایت ها گسترش پیدا میکنند و مخاطبان بیشتری را در بر میگیرند .تا به حال بیش از ۱۲۵.۰۰۰.۰۰۰ دامنه در اینترنت ثبت شده است که نشان دهنده گسترش سریع اینترنت می باشد با بزرگ شدن و زیاد شدن مخاطبان وب سایت ها

معرفی زبان های برنامه نویسی

دیگر HTML پاسخگوی نیاز مدیران وب سایت ها نبود، از این رو به زبانی نیاز بود تا بتواند مانند نرم افزارهای Dektop عمل کند و به راحتی قابل گسترش باشد. زبان هایی مانند PHP, Asp, Java و ... به وجود آمدند تا به وب سایت ها خدمت کنند. در این مقاله شما را تا حدودی با زبان برنامه نویسی PHP آشنا میکنیم و نگاه کوتاهی به تاریخچه و عمل کرد PHPخواهیم انداخت .

PHPیک زبان قدرتمند برای ساخت وب سایت های پویا است. این زبان اسکریپتی میتواند با HTML ادغام شود php . یک زبان در سمت سرور است, بدین معنا که کدهای php روی سرور تفسیر میشوند و خروجی html و یا خروجی های دیگری تولید میکند که توسط کاربر قابل مشاهده است .

phpدر سال ۱۹۹۴ توسط Rasmus Lerdorf ارایه شد. از آن زمان تا به حال تغییرات زیادی در این زبان اسکریپتی تحت لینوکس ایجاد شده است و در حال حاضر نسخه ۵ آن توسط تیم گسترش دهنده php عرضه شده است. تا به امروز حدودا بیش از ۳۵.۰۰۰.۰۰۰ وب سایت از این زبان برای ساخت برنامه های تحت وب خود استفاده کرده اند و به عنوان یکی از پرطرفدارترین زبان های اسکریپتی به حساب می آید .

برخی از رقبای php عبارتند از Perl, JSP, ColdFusion :و ASP.NET اما زبان php برتریهای بسیاری در مقایسه با رقیبان خود دارد مانند :

کارایی بالا، واسطه های مختلف برای سیستم پایگاه های اطلاعاتی مختلف، کتابخانه داخلی برای انجام امور متداول، هزینه پایین، امنیت بالا و ... یکی از نکات مهم زبان php مستقل از محیط کار بودن آن است، بدین صورت که در تمامی کامپیوترها و سیستم عامل ها قابل اجرا است php .روی لینوکس، ویندوز و ...به سادگی اجرا میشود .

برنامه های php را به وسیله مرورگر وب اجرا میکنیم. به وب سایتی که برنامه php در آن قرار دارد میرویم و با در خواست ما سرور کدهای php را تجزیه و تحلیل میکند و پاسخ آن را به صورت html به ما نشان میدهد. با استفاده از زبان php میتوان وب سایت هایی را ساخت که :

داده ها را از منابع مختلفی مانند بانک های اطلاعاتی و یا فایل ها جمع آوری کنند .

عناصری مثل جستجو، فروم، عضویت، ورود و خروج کاربران، گالری عکس و ... ایجاد کرد .

سیستم پست الکترونیکی ایجاد کرد، فروشگاه آنلاین ساخت و به کاربران اجاز خرید داد و ...

به طور کلی میتوان گفت php برای وب سایت هایی مناسب است که با کاربران زیادی سر و کار داردند و به صورت مرتب به روز میشوند. شاید این نکته برای شما جالب باشد که بدانید Yahoo به عنوان یکی از هامیان php است و بسیاری از قسمت های خود را با زبان php ساخته است. از وب سایت های معروفی دیگری که از php استفاده میکنند میتوان به

Google, flickrو Facebook اشاره کرد .

برنامه نویسی php معمولا به این صورت است که برنامه نویسان در یک کامپیوتر (بدون اتصال به اینترنت) برنامه خود را مینویسند، آزمایشات مقدماتی خود را روی آن انجام میدهند سپس آن را روی سرور منتقل میکنند. در بیشتر موارد برنامه نویس اطلاعات وب سایت را در داخل یک بانک اطلاعاتی قرار میدهید، اطلاعاتی مانند شناسه کاربران و کلمات عبورشان، اخبار و نوشته ها ....سپس با استفاده از php به بانک اطلاعاتی متصل میشوند و با اطلاعات آن کار میکنند. یکی از متداولترین بانک های اطلاعاتی که برای php مورد استفاده قرار میگیرد MySQL است MySQL .چندی پیش توسط شرکت Sun خریداری شد و در حال حاضر نسخه ۵ آن آخرین نسخه ارایه داده شده می باشد .

همانطور که در بالا هم گفته شد در زمان نوشتن این مقاله php 5 آخرین نسخه ارایه شده توسط تیم php است (چندی پیش تیم php گسترش و پشتیبانی php ۴ را به صورت رسمی کنار گذاشت). اگر بخواهیم نگاه کوتاهی به php5 بیندازیم میتوانیم به نکات زیر اشاره کنیم :

پشتیبانی کاملا مناسب از مدل شیء گرا

مدیریت آسان داده های XML

مدیریت استثنایی خطاها ...

با آمدن php 5 تحول جدیدی در php ایجاد شد به صورتی که Microsoft سازنده زبان Asp بار دیگر از رقیب خود یعنی php عقب افتاد.

در آخر میتوان گفت php زبانی است که وب سایت را به همان صورت که تمایل دارید میسازد و رویاهای برنامه نویسان را به حقیقت تبدیل میکند.

۸.زبان Java

جاوا:

(Java): یک زبان برنامه‌نویسی شئ‌گراست که برای اولین بار توسط جیمز گوسلینگ در شرکت سان مایکروسیستمز ایجاد شد و در سال ۱۹۹۵ به عنوان بخشی از سکوی جاوا منتشر شد. زبان جاوا شبیه به C++است اما مدل شیءگرایی آسان‌تری دارد و از قابلیت‌های سطح پایین کمتری پشتیبانی می‌کند. یکی از قابلیت‌های اصلی جاوا این است که مدیریت حافظه را بطور خودکار

معرفی زبان های برنامه نویسی

انجام می‌دهد. ضریب اطمینان عملکرد برنامه‌های نوشته‌شده به این زبان بالا است و وابسته به سیستم‌عامل خاصی نیست، به عبارت دیگر می‌توان آن را روی هر رایانه با هر نوع سیستم‌عاملی اجرا کرد. برنامه‌های جاوا به صورت کدهای بیتی همگردانی(کامپایل) می‌شوند. که مانند کد ماشین هستند و به ویژه وابسته به سیستم‌عامل خاصی نیستند.

تاریخچه:

در مقایسه با زبان‌های دیگر، همچون C++یا بیسیک یا فورترن، جاوا زبان نسبتاً جدیدتری است. شرکت sun) (Microsystemsسان مایکروسیستمز در سال ۱۹۹۱ یک پروژه تحقیقاتی به نام گرین (Green) را آغاز کرد. هدف این پروژه ایجاد زبانی جدید شبیه به C++بود که نویسنده اصلی آن، جیمز گاسلینگ، آن را *طوط* (Oak)نامید. اما بعدها به دلیل برخی مشکلات حقوقی از میان لیستی از کلمات تصادفی نام آن به جاوا تغییر کرد.

پروژه گرین به دلیل مشکلات بازاریابی در شرف لغو شدن بود تا اینکه گسترش وب در سال۱۹۹۳ باعث نمایش توانایی‌های وافر جاوا در این عرصه گشت .اینگونه بود که شرکت سان مایکروسیستمز در مه ۱۹۹۵ جاوا را رسماً به بازار عرضه کرد.

جاوا یک زبان برنامه‌نویسی است که در آغاز توسط شرکت سان مایکروسیستمز ایجاد شده‌است و در سال ۱۹۹۵ به عنوان بخش اصلی سکوی جاوا منتشر شد. این زبان قسمت‌های بسیاری از گرامر خود را از Cو C++گرفته اما دارای مدل شی‌گرایی ساده‌ای است و امکانات سطح پایین کمی دارد. کاربرد جاوا در کامپایل به صورت بایت کد است که قابلیت اجرا روی تمامی ماشین‌های شبیه‌سازی جاوا را داشته باشد صرف نظر از معماری و خصوصیات آن کامپیوتر. اجرای اصلی کامپایلرهای جاوا، ماشین‌های پیاده‌سازی و کتابخانه‌های آن توسط این شرکت از سال ۱۹۹۵ منتشر شد. در ۱۹۹۷ may این شرکت، نرم‌افزار رایگان این زبان را فراهم کرد. دیگران هم کاربردهای دیگری از این زبان را منتشر کردند مثل کامپایلر GNU برای جاوا.

مرورگرهای اصلی وب، به هم پیوستند تا به طور مطمئن جاوا اپلت را بدون صفحات وب اجرا کنند و به این صورت جاوا خیلی زود معروف و محبوب شد. با پیدایشjava2 ، نسخه جدید توانست ترکیب‌های جدیدی را برای نوع‌های مختلف پلت فرم‌ها ایجاد کند. به عنوان مثالJ2EE ، باهدف کاربرد برای تشکیلات اقتصادی، و نسخه سکوی جاوا، نسخه میکرو برای موبایل منتشر شد. در سال ۱۹۹۶ با هدف بازاریابی، این شرکت نسخه جدید J2 را با نام‌های سکوی جاوا، نسخه سازمانی، سکوی جاوا، نسخه میکرو و سکوی جاوا، نسخه استاندارد منتشر کرد. در سال ۱۹۹۷ شرکت سان میکروسیستمز، ISO/IEC JTC1 standards bodyو Ecma International را به فرمول جاوا تغییر داد. شرکت sun بسیاری از کاربردهای جاوایش را بدون هیچ هزینه‌ای فراهم آورد. شرکت sun با فروش مجوز برای بعضی از کاربردهای خاصش مثل Java Enterprise System درآمدی را بدست آورد. اولین تمایزی که بین SDKو سکوی جاوا داد شامل فقدان کامپایلر برای JREو سرفایل‌ها بود. در ۱۳ نوامبر ۱۹۹۶ شرکت sun نرم‌افزار جاوا را به صورت رایگان و با مجوز عمومی برای همه منتشر کرد. جاوا یک پلتفرم نرم‌افزاری است.

معرفی زبان های برنامه نویسی

اهداف اولیه

۱. این زبان باید ساده، شی‌گرا و مشهور باشد.
۲. مطمئن و بدون خطا باشد.
۳. وابسته به معماری کامپیوتر نبوده و قابل انتقال باشد.
۴. باید با کارایی بالا اجرا شود.
۵. باید به صورت پویا و نخ‌کشی‌شده باشد.

برنامه‌های جاوا و اپلت‌ها

جاوا برای نوشتن انواع برنامه‌های کاربردی مناسب است. با جاوا می‌توان انواع برنامه‌های زیر را نوشت:

- برنامه‌های تحت وب
- برنامه‌نویسی سیستم‌های کوچک مانند موبایل، پاکت پی‌سی و...
- برنامه‌های کاربردی بزرگ(Enterprise)
- برنامه‌های رومیزی(Desktop)
- و غیره.

قابلیت خاصی در جاوا وجود دارد بنام اپلت .اپلت‌ها امکانات فراوانی برای نوشتن برنامه‌های تحت وب در اختیار برنامه‌نویسان قرار می‌دهند که دیگر زبان‌های برنامه‌نویسی فاقد آن هستند. البته وجود ماشین مجازی جاوا برای اجرای اپلت لازم است. اپلت‌ها نظیر فناوری اکتیوایکس شرکت مایکروسافت هستند که برنامه‌نویسان را قادر می‌سازد تا امکاناتی را به مرورگر کاربر بیفزایند. البته تفاوت این دو در امنیت می‌باشد به گونه‌ای که اپلت‌ها بدلیل اینکه در محیطی به نام جعبه شنی اجرا می‌شوند امن هستند ولی Activexها فاقد چنین امنیتی هستند.

۱. **سیستم‌عامل** :هر چقدر زبانهای net.قوی باشند تنها بر روی پلت فرم ویندوز اجرا می‌شوند و برخی ویندوز را سیستم‌عامل غیر قابل اعتمادی در برنامه نویسی Enterprise می‌دانند. ولی جاوا از این نظر انتخابی خوب است.
۲. **قابلیت حمل** :جاوا بر روی پلتفرم‌های گوناگونی قابل اجرا است، از ATM و ماشین رختشویی گرفته تا سرورهای سولاریس با قابلیت پشتیبانی از ۱۰۲۴ cpu برای پردازش.
۳. **جاوا بیشتر از یک زبان است** :جاوا فقط یک زبان نیست و انجمن‌هایی متشکل از بزرگان صنایع و برنامه‌نویسان زیادی مشغول به توسعه و ایجاد استانداردهای جدید و به روز هستند.

خط مشی جاوا

یکی از ویژگی‌های جاوا قابل حمل بودن آن است. یعنی برنامه نوشته شده به زبان جاوا باید به طور مشابهی در کامپیوترهای مختلف با سخت‌افزارهای متفاوت اجرا شود. و باید این توانایی را داشته باشد که برنامه یک بار نوشته شود، یک بار کامپایل شود و در همه کامپیوترها اجرا گردد. به این صورت که کد کامپایل شده جاوا را ذخیره می‌کند، اما نه به‌صورت کد ماشین بلکه به‌صورت بایت کد جاوا. دستورالعمل‌ها شبیه کد ماشین هستند، اما با ماشین‌های مجازی که به طور خاص برای سخت‌افزارهای مختلف نوشته شده‌اند، اجرا می‌شوند. در نهایت کاربر از سکوی جاوا نصب شده روی ماشین خود یا مرورگر وب استفاده می‌کند. کتابخانه‌های استاندارد یک راه عمومی برای دسترسی به ویژگی‌های خاص فراهم می‌کنند. مانند گرافیک، نخ‌کشی و شبکه. در بعضی از نسخه‌های ماشین مجازی جاوا بایت کدها می‌توانند قبل و در زمان اجرای برنامه به کدهای محلی کامپایل شوند. فایده اصلی استفاده از بایت کد، قسمت کردن است. اما ترجمه کلی یعنی برنامه‌های ترجمه شده تقریباً همیشه کندتر از برنامه‌های کامپایل شده محلی اجرا می‌شوند. این شکاف می‌تواند با چند تکنیک خوش‌بینانه که در کاربردهای JVM قبلی معرفی شد، کم شود. یکی از این تکنیک‌ها JIT است که بایت کد جاوا را به کد محلی ترجمه کرده و سپس آن را پنهان می‌کند. در نتیجه برنامه خیلی سریع‌تر نسبت به کدهای ترجمه شده خالص شروع و اجرا می‌شود. بیشترVM های پیشرفته، به‌صورت کامپایل مجدد پویا، در آنالیزVM ، رفتار برنامه اجرا شده و کامپایل مجدد انتخاب شده و بهینه‌سازی قسمت‌های برنامه، استفاده می‌شوند. کامپایل مجدد پویا می‌تواند کامپایل ایستا را بهینه‌سازی کند .زیرا می‌تواند قسمت hot spot برنامه و گاهی حلقه‌های داخلی که ممکن است زمان اجرای برنامه را افزایش دهند را تشخیص دهد. کامپایل JITو کامپایل مجدد پویا به برنامه‌های جاوا اجازه می‌دهد که سرعت اجرای کدهای محلی بدون از دست دادن قابلیت انتقال افزایش پیدا کند.

تکنیک بعدی به عنوان کامپایل ایستا شناخته شده‌است. که کامپایل مستقیم به کدهای محلی است مانند بسیاری از کامپایلرهای قدیمی. کامپایلر ایستای جاوا، بایت کدها را به کدهای شی محلی ترجمه می‌کند.

کارایی جاوا نسبت به نسخه‌های اولیه بیشتر شد. در تعدادی از تست‌ها نشان داده شد که کارایی کامپایلر JITکاملاً مشابه کامپایلر محلی شد. عملکرد کامپایلرها لزوماً کارایی کدهای کامپایل شده را نشان نمی‌دهند. یکی از پیشرفت‌های بی نظیر در در زمان اجرای ماشین این بود که خطاها ماشین را دچار اشکال نمی‌کردند. علاوه بر این در زمان اجرای ماشینی مانند جاوا وسایلی وجود دارد که به زمان اجرای ماشین متصل شده و هر زمانی که یک استثنا رخ می‌دهد، اطلاعات اشکال زدایی که در حافظه وجود دارد، ثبت می‌کنند.

پیاده سازی

شرکت سان میکروسیستم مجوز رسمی برای پلت فرم استاندارد جاوا را به مایکروسافت ویندوز ,لینوکس، و .Solarisداده‌است. همچنین محیط‌های دیگری برای دیگر پلت فرم‌ها فراهم آورده‌است .علامت تجاری مجوز شرکت سان میکروسیستم طوری بود که با همه پیاده سازی‌ها سازگار باشد. به علت اختلاف قانونی که با ماکروسافت پیدا کرد، زمانی که شرکت سان ادعا کرد که پیاده سازی ماکروسافت از RMI یا JNI پشتیبانی نکرده و ویژگی‌های خاصی را برای خودش اضافه کرده‌است. شرکت سان در سال ۱۹۹۷ پیگیری قانونی کرد و در سال ۲۰۰۱ در توافقی ۲۰ میلیون دلاری برنده شد. در نتیجه کمی بعدماکروسافت جاوا را به

ویندوز فرستاد. در نسخه اخیر ویندوز، مرورگر اینترنت نمی‌تواند از جاوا پلت فرم پشتیبانی کند. شرکت سان و دیگران یک سیستم اجرای جاوای رایگان برای آنها و نسخه‌های دیگر ویندوز فراهم آوردند.

اداره خودکار حافظه

جاوا از حافظه بازیافتی خودکار برای اداره حافظه در چرخه زندگی یک شی استفاده می‌کند. برنامه‌نویس زمانی که اشیا به وجود می‌آیند، این حافظه را تعیین می‌کند. و در زمان اجرا نیز، زمانی که این اشیا در استفاده زیاد طولانی نباشند، برنامه نویس مسئول بازگرداندن این حافظه‌است. زمانی که مرجعی برای شی‌های باقیمانده نیست، شی‌های غیر قابل دسترس برای آزاد شدن به صورت خودکار توسط بازیافت حافظه، انتخاب می‌شوند. اگر برنامه‌نویس مقداری از حافظه را برای شی‌هایی که زیاد طولانی نیستند، نگه دارد، چیزهایی شبیه سوراخ حافظه اتفاق می‌افتند.

یکی از عقایدی که پشت سر مدل اداره حافظه خودکار جاوا وجود دارد، این است که برنامه‌نویس هزینه اجرای اداره دستی حافظه را نادیده می‌گیرد. در بعضی از زبان‌ها حافظه لازم برای ایجاد یک شی، به صورت ضمنی و بدون شرط، به پشته تخصیص داده می‌شود. و یا به‌طور صریح اختصاص داده شده و از heap بازگردانده می‌شود. در هر کدام از این راه‌ها، مسئولیت اداره اقامت حافظه با برنامه‌نویس است. اگر برنامه شی را برنگرداند، سوراخ حافظه اتفاق می‌افتد. اگر برنامه تلاش کند به حافظه‌ای را که هم‌اکنون بازگردانده شده، دستیابی پیدا کند یا برگرداند، نتیجه تعریف شده نیست و ممکن است برنامه بی‌ثبات شده و یا تخریب شود. این ممکن است با استفاده از اشاره‌گر مدتی باقی بماند، اما سرباری و پیچیدگی برنامه زیاد می‌شود. بازیافت حافظه اجازه دارد در هر زمانی اتفاق بیفتد. به‌طوری که این زمانی اتفاق می‌افتد که برنامه بی‌کار باشد. اگر حافظه خالی کافی برای تخصیص شی جدید در heapوجود نداشته باشد، ممکن است برنامه برای چند دقیقه متوقف شود. در جایی که زمان پاسخ یا اجرا مهم باشد، اداره حافظه و منابع اشیا استفاده می‌شوند.

جاوا از نوع اشاره‌گر ریاضی C و C++ پشتیبانی نمی‌کند. در جایی که آدرس اشیا و اعداد صحیح می‌توانند به جای هم استفاده شوند. همانند C++ و بعضی زبان‌های شی‌گرای دیگر، متغیرهای نوع‌های اولیه جاوا شی‌گرا نبودند. مقدار نوع‌های اولیه، مستقیماً در فیلدها ذخیره می‌شوند. در فیلدها (برای اشیا) و در پشته (برای توابع)، بیشتر از heapاستفاده می‌شود. این یک تصمیم هوشیارانه توسط طراح جاوا برای اجرا است. به همین دلیل جاوا یک زبان شی‌گرای خالص به حساب نمی‌آید.

گرامر

گرامر جاوا خیلی بزرگتر از C++ است. مثل C++ که ترکیب ساختارها و برنامه‌های شی‌گرا می‌باشد، نیست. بلکه زبان جاوا یک زبان شی‌گرای خالص است. همه کدهایی که داخل کلاس نوشته می‌شود و همه چیزهایی که داخل شی است، با استثنائات نوع داده اصلی، که به صورت کلاس نیستند، برای اجرا .جاوا بسیاری از ویژگی‌ها را پشتیبانی می‌کند. از کلاس‌ها برای ساده‌تر کردن زبان و جلوگیری از رخداد خطا.

نمونه‌هایی از برنامه‌های جاوا

در زیر نمونه‌ای از برنامه‌ای که در جاوا نوشته شده‌است آورده شده‌است. البته برای کامپایل کردن این برنامه بایستی کیت توسعه جاوا بر روی سیستم مورد نظر نصب شده باشد.

```
public class Test{
public static void main(String[] args) {
   System.out.println("HelloWorld!");
                    }
                    }
```

برای اجرای برنامه بالا، ابتدا باید یک فایل به نام Test.java ساخته شود و سپس کامپایل شود:

```
$ javac Test.java
```

سپس یک فایل خروجی به نام Test.class دریافت می‌شود. بعد با استفاده از دستور زیر برنامه قابل اجرا است:

```
$ java Test
```

مثال‌ها

برنامه Hello world به این صورت در زبان جاوا می‌تواند نوشته شود:

```
// HelloWorld.java
public class HelloWorld {
public static void main(String[] args) {
  System.out.println("Hello, World!");
                    }
                    }
```

بر طبق قرارداد فایل هل بعد از کلاس‌های عمومی نام گذاری می‌شوند. سپس باید پسوند java را به این صورت اضافه‌کرد : Hello world.java. این فایل اول باید با استفاده از کامپایلر جاوا به بایت کد کامپایل شود. در نتیجه فایل Hello world.class ایجاد می‌شود. این فایل قابل اجرا است. فایل جاوا ممکن است فقط یک کلاس عمومی داشته باشد. اما می‌تواند شامل چندین کلاس با دستیابی عمومی کمتر باشد.

کلاسی که به صورت خصوصی تعریف می‌شود ممکن است در فایل java.ذخیره شود .کامپایلر برای هر کلاسی که در فایل اصلی تعریف می‌شود یک کلاس فایل تولید می‌کند. که نام این کلاس فایل همنام کلاس است با پسوندclass.

کلمه کلیدی) public عمومی) برای قسمت‌هایی که می توانند از کدهای کلاس‌های دیگر صدا زده بشوند، به کار برده می‌شود. کلمه کلیدی) static ایستا) در جلوی یک تابع، یک تابع ایستا را که فقط وابسته به کلاس است و نه قابل استفاده برای نمونه‌هایی

از کلاس، نشان می‌دهد. فقط تابع‌های ایستا می‌توانند توسط اشیا بدون مرجع صدا زده شوند. داده‌های ایستا به متغیرهایی که ایستا نیستند، نمی‌توانند دسترسی داشته باشند.

کلمه کلیدی) void تهی) نشان می‌دهد که تابع main هیچ مقداری را بر نمی‌گرداند. اگر برنامه جاوا بخواهد با خطا از برنامه خارج شود، باید system.exit()صدا زده شود. کلمه main یک کلمه کلیدی در زبان جاوا نیست .این نام واقعی تابعی است که جاوا برای فرستادن کنترل به برنامه، صدا می‌زند. برنامه جاوا ممکن است شامل چندین کلاس باشد که هر کدام دارای تابع mainهستند.

تابع main باید آرایه‌ای از اشیا رشته‌ای را بپذیرد. تابع main می‌تواند از آرگومان‌های متغیر به شکل public static void main(string…args)استفاده کند که به تابع main اجازه می‌دهد اعدادی دلخواه از اشیا رشته‌ای را فراخوانی کند. پارامتر string[]argsآرایه‌ای از اشیا رشته ایست که شامل تمام آرگومان‌هایی که به کلاس فرستاده می‌شود، است.

چاپ کردن، قسمتی از کتابخانه استاندارد جاوا است. کلاس سیستم یک فیلد استاتیک عمومی به نام out تعریف کرده‌است. شی outیک نمونه از کلاس printstreamاست و شامل تعداد زیادی تابع برای چاپ کردن اطلاعات در خروجی استاندارد است. همچنین شامل println(string) برای اضافه کردن یک خط جدید برای رشته فرستاده شده اضافه می‌کند.

توزیع‌های جاوا

منظور از توزیع جاوا پیاده‌سازی‌های مختلفی است که برای کامپایلر جاوا و همچنین مجموعه کتابخانه‌های استاندارد زبان جاوا (JDK)وجود دارد. در حال حاضر چهار توزیع‌کننده عمده جاوا وجود دارند:

- **سان میکروسیستمز :**توزیع کننده اصلی جاوا و مبدع آن می‌باشد. در اکثر موارد هنگامی که گفته می‌شود جاوا منظور توزیع سان می‌باشد.
- **GNU Classpath:**این توزیع از سوی موسسه نرم‌افزارهای آزاد منتشر شده و تقریبا تمامی کتابخانه استاندارد زبان جاوا در آن بدون بهره‌گیری از توزیع شرکت سان از اول پیاده‌سازی شده‌است. یک کامپایلر به نام GNU Compiler for Javaنیز برای کامپایل کردن کدهای جاوا توسط این موسسه ایجاد شده‌است. فلسفه انتخاب نام Classpathبرای این پروژه رها کردن تکنولوژی جاوا از وابستگی به علامت تجاری جاوا است طوریکه هیچ وابستگی یا محدودیتی برای استفاده آن از لحاظ قوانین حقوقی ایجاد نشود و از طرفی به خاطر وجود متغیر محیطی classpathدر تمامی محیط‌های احرایی برنامه‌های جاوا، این نام به نوعی تکنولوژی جاوا را برای خواننده القا می‌کند. کامپایلر GNU توانایی ایجاد کد اجرایی (در مقابل بایت کد توزیع سان) را داراست. لازم به ذکر است که در حال حاضر شرکت سان تقریبا تمامی کدهای JDK را تحت مجوز نرم‌افزارهای آزاد به صورت متن باز منتشر کرده‌است و

قول انتشار قسمت بسیار کوچکی از این مجموعه را که به‌دلیل استفاده از کدهای شرکت‌های ثانویه نتوانسته به صورت متن باز منتشر نماید در آینده نزدیک با بازنویسی این کدها داده‌است.

- **مایکروسافت :J#** این در حقیقت یک توزیع جاوا نیست. بلکه زبانی مشابه می‌باشد که توسط مایکروسافت و در چارچوب net.ارائه شده‌است. انتظار اینکه در سیستم‌عاملی غیر از ویندوز هم اجرا شود را نداشته باشید.

- **AspectJ:**این نیز یک زبان مجزا نیست. بلکه یک برنامه الحاقی می‌باشد که امکان برنامه نویسی Aspect Orientedرا به جاوا می‌افزاید. این برنامه توسط بنیاد برنامه‌نویسی جلوه‌گرا و به صورت کدباز ارائه شده‌است.

کلاس‌های خاص

برنامه‌های کاربردی کوچک

اپلت جاواها برنامه‌هایی هستند که برای کاربردهایی نظیر نمایش در صفحات وب، ایجاد شده‌اند. واژه import باعث می‌شود کامپایلر جاوا کلاس‌های javaapplet.Appletو java.awt.Graphicsرا به کامپایل برنامه اضافه کند .کلاس Hello کلاس Applet را توسعه می‌دهد. کلاس اپلت چارچوبی برای کاربردهای گروهی برای نمایش و کنترل چرخه زندگی اپلت، درست می‌کند. کلاس اپلت یک تابع پنجره‌ای مجرد است که برنامه‌های کوچکی با قابلیت نشان دادن واسط گرافیکی برای کاربر را فراهم می‌کند. کلاس Hello تابع موروثی print(Graphics)را از سوپر کلاس container باطل می‌کند، برای اینکه کدی که اپلت را نمایش می‌دهد، فراهم کند. تابع paint شی‌های گرافیکی را که شامل زمینه‌های گرافیکی هستند را می‌فرستد تا برای نمایش اپلت‌ها استفاده شوند .تابع paint برای نمایش "Hello world!" تابع drawstring(string,int,int) را صدا می‌زند.

جاوا سرولت

تکنولوژی servlet جاوا گسترس وب را به آسانی فراهم می‌کند. و شامل مکانیزم‌هایی برای توسعه تابعی سرور وب و برایدسترسی به سیستم‌های تجاری موجود است servlet.قسمتی از javaEE است که به درخواست‌های مشتری پاسخ می‌دهد.

```
// Hello.java
import java.io. *;
import javax.servlet. *;

public class Hello extends GenericServlet {
public void service(ServletRequest request, ServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
final PrintWriter pw = response.getWriter();
pw.println("Hello, world!");
pw.close();
}
```

```
}
```

واژه import کامپایلر جاوا را هدایت می‌کند که تمام کلاس‌های عمومی و واسطها را از بسته‌های java.io و java.servlet را در کامپایل وارد کند.

کلاس Hello کلاس Genericservlet را توسعه می‌دهد. کلاس Genericservlet واسطی برای سرور فراهم می‌کند تا درخواست را به servlet بفرستد و چرخه زندگی servlet را کنترل کند.

JSP

صفحه سرور جاوا قسمتی از سرور javaEE است که پاسخ تولید می‌کند. نوعاً صفحات HTML به درخواست‌های HTTP از مشتریJSP.ها کد جاوا در صفحه HTML را با استفاده از حائل <%and%>اضافه می‌کنند JSP.به javaservlet کامپایل می‌شود.

سوینگ

Swingکتابخانه واسط گرافیکی کاربر است برای پلت فرم .javaSE ابزاری مشابه پنجره، GTKو motif توسط شرکت sunفراهم شده‌اند. این مثال کاربرد swingیک پنجره واحد همراه با Hello world را ایجاد می‌کند.

```
// Hello.java (Java SE 5)
import java.awt.BorderLayout;
import javax.swing. *;

public class Hello extends JFrame {
public Hello() {
super("hello");
setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
setLayout(new BorderLayout());
add(new JLabel("Hello, world!"));
pack();
}

public static void main(String[] args) {
new Hello().setVisible(true);
}
}
```

اولین جمله import کامپایلر جاوا را هدایت می‌کندتا کلاس Borderlayout را از بسته java.awt در جاوا به کامپایل اضافه کند. و import دوم همه کلاس‌های عمومی و واسط آن‌ها را از بسته javax.swing اضافه می‌کند. کلاس Helloکلاس Jframeرا توسعه می‌دهد. کلاس Jframe یک پنجره با میله عنوان و کنترل بستن است.

زمانی که برنامه آغاز می‌شود، تابع main با JVM صدا زده می‌شود. این یک نمونه جدید از کلاس Hello را ایجاد کرده و با صدا زدن تابع setvisible(boolean)با مقدار true نمایش داده می‌شود.

Generics

قبل از کلاس‌های عمومی برای هر متغیر باید یک نوع خاص تعریف می‌کردیم .به عنوان مثال برای کلاس‌های ظرف این امر مشکل بود زیرا را آسانی برای ایجاد یک container وجود نداشت که نوع‌های خاصی از اشیا را بپذیرد .کلاس‌های عمومی اجازه می‌دهند نوع زمان کامپایل، بدون نیاز به ایجاد تعداد زیادی ازcontainer ، چک شود. همه آنها کدهای مشابهی دارند.

کتابخانه‌های کلاس

کتابخانه‌های جاوا که به صورت بایت کد از کد اصلی کامپایل شده‌اند، برای پشتیبانی از بعضی از کاربردهای جاوا، توسط JRE منتشر شده‌است. مثال‌هایی از این کتابخانه‌ها عبارتند از:

- کتابخانه‌های مرکزی که شامل:
- کتابخانه‌هایی که برای ساختار داده کاربرد دارند. مثل لیست‌ها، درخت‌ها، مجموعه‌ها، مترجم‌ها.
- (کتابخانه پرداز) XML تجزیه، تغییر شکل، اعتبار
- کتابخانه‌های موضعی و بین‌المللی
- کتابخانه‌های انتگرال گیری که امکان تایپ کردن توسط سیستم‌های بیرونی را می‌دهند.
- JDBCبرای دستیابی به داده‌ها
- JNDIبرای مراجعه و کشف کردن
- CORBA & RMIبرای توسعه کاربرد توزیع کردن
- کتابخانه‌های واسط کاربر
- AWT(توابع پنجره‌ای مجرد)که قسمت‌هایی از GUI را فراهم می‌کنند.
- کتابخانه‌های swing که در AWT ساخته شده‌اند اما کاربردهایی از AWT widgetry را فراهم می‌کنند.
- APLها برای ضبط صدا، پردازش و بازنواختی
- کاربردهای وابسته پلت فرم ماشین‌های مجازی جاوا
- Pluginsکه توانایی اجرا شدن در مرورگرهای وب را به اپلت می‌دهد.
- java web start
- دادن مجوز و مستند سازی

شرکت سان میکروسیستم، ۴ نوع ویرایش از کاربردهای مختلف جاوا را ارائه داده‌است:

۱. Java card for smartcard

۲. JavaME
۳. JavaSE
۴. JavaEE

ایرادات مطرح شده :

مهم‌ترین ایرادی که برنامه نویسان سایر زبان‌ها به زبان جاوا می‌گیرند سرعت اجرایی پایین جاوا در مقایسه با زبان‌ها سطح پایین‌تر مانند C++ و اسمبلی است. یک برنامه جاوا به صورت بایت کد می‌باشد و باید در ماشین مجازی جاوا اجرا گردد. به همین دلیل سرعت اجرای پایینی را در مقابل زبان‌هایی همچون C++ دارد. به صورت دیگر یک برنامه C به طور متوسط تا ۱۰ برابر سریعتر از برنامه مشابه جاوا اجرا می‌گردد.

جاوا علی رغم شیءگرا بودن در بخشی از قسمت‌ها برخی از اصول شی‌گرایی را نادیده گرفته‌است .[1]از جمله این قسمت‌ها قابلیت بازتابش Reflection می‌باشد. هدف اصلی بازتابش بررسی (مشاهده) و ایجاد تغییر در برنامه در حال اجرا است ولی این مهم با زیر پا گذاشتن بعضی اصول ممکن شده‌است. برای نمونه با استفاده از بازتابش (و در صورت داشتن مجوز لازم ضمن اجرای برنامه) می‌توان به متدهای خصوصی دیگر کلاس‌ها دسترسی داشت.

زبان جاوا در مقابل زبانی مثل++C ساده تر و یادگیری آن آسانتر است. این آسانتر بودن با حذف بسیاری از موارد که باعث قدرتمندتر بودن زبان++C بوده‌اند ایجاد شده‌است. مهم‌ترین این موارد اشاره گرها و وراثت چندگانه بوده‌اند که در زبان جاوا یافت نمی‌شوند.

از آنجایی که جاوا زبانی با عدم وابستگی به بستر می‌باشد پس استفاده از توابع سیستم‌عامل در برنامه را به طور مستقیم نمی‌پذیرد. به همین صورت نمی‌توان به طور مستقیم از واسط‌های برنامه نویسی غیر از جاوا در آن استفاده نمود.

پاسخ به ایرادات:

سرعت پایین برنامه‌های جاوا در محیطی که اجرا می‌شوند ملاک کارایی نبوده زیرا در محیط وب مسئله‌ای که سرعت را کند می‌سازد، شبکه بوده و ابتدا باید سربار شبکه را از روی برنامه‌ها برداشت. از طرف دیگر در برنامه‌های رومیزی هم در JDK 5.0 و ۶،۰ بهینه سازی بسیاری بوجود آمده که این مسئله باعث شده که در آخرین تست کارایی که انجام شده یک برنامه جاوا در محدوده ۰،۸ تا ۱،۳ همان برنامه در C++ کارایی داشته باشد که ۱،۳ آن مربوط به بخش واسط کاربری و سرعت ۰،۸ آن مربوط به بسته تخلیه حافظه می‌شده که هیچ الگوریتمی نتوانست از الگوریتم Garbage Collector جاوا پیشی بگیرد. همچنین سال ۱۹۹۹ در مقاله‌ای آقای Lutz Prechelt به این مسئله را ثابت کردند که تجربه برنامه‌نویسی که برنامه‌ای را می‌نویسد از انتخاب زبانی که برنامه برروی آن نوشته می‌شود در کارایی تأثیر بیشتری دارد و این بدان معناست که کارایی یک برنامه را برنامه‌نویس مشخص می‌کند و نه زبان برنامه‌نویسی(ایشان در همان مقاله از زبان جاوا استفاده نمودند تا ذهنیت بد را از بین ببرند.)

همچنین در صنعت نرم‌افزار هزینه اصلی مربوط به ساخت نرم‌افزار است و نه تهیه سخت‌افزار برای دستیابی به سرعت بیشتر.

حذف اشاره‌گرها در جاوا به دلیل مشکلاتی بوده که آنها در طول تاریخشان بوجود آورده‌اند، اگرچه این موارد در برنامه‌های سیستمی لازم به‌نظر می‌رسد ولی در محیط‌های تحت‌وب که بستر اصلی جاوا هستند می‌توانند اثراتی به مراتب شدیدتر نسبت به آنچه در برنامه‌های سیستمی دارند داشته باشند و باعث می‌شود که توجه برنامه‌نویسان از مسائلی چون کارایی، قابلیت اطمینان و مقیاس‌پذیری برنامه به تنظیم اشاره‌گرها معطوف گردد.

وجود وراثت چندگانه در زبانی مانندC++ ، باعث ایجاد مشکلات اساسی‌ای می‌گردید که اکثر برنامه‌نویسان C++ از آن دوری می‌کرده و هنوز هم می‌کنند .ولی قابلیت چندریخته شدن یک کلاس از لحاظ شی گرایی بسیار مهم بوده و بنابراین توجیهی برای وجود وراثت چندگانه را فراهم می‌نمود. در جاوا با وارد شدن مفهومی به نام واسط برنامه‌سازی(Interface) ، دیگر نیازی به وجود وراثت چندگانه احساس نشد و بنابراین از زبان جاوا حذف گردید در حال حاضر اکثر طراحان برنامه‌ها حتی به این نتیجه رسیده‌اند که وراثت تکی هم باعث ایجاد مشکل بوده و تا آنجایی که می‌شود باید از Composition استفاده نمود و در تمامی کتاب‌های طراحی که از سال ۲۰۰۰ به این طرف چاپ شده به آن اشاره نموده‌اند.

از ابتدای بوجود آمدن جاوا، کتابخانه JNI - Java Native Interface در آن وجود داشته که قابلیت فراخوانی و دستکاری برنامه‌هایی در C++ و... را می‌داده که از نمونه‌های آن می‌توان به Jtwain که یک بسته‌ایست که از کتابخانه‌های ویندوز برای اسکن عکس استفاده می‌کند، یا SWT که یک بسته نرم‌افزاریست که از کتابخانه‌های ویندوز و لینوکس (برحسب سیستم‌عامل) برای ساخت واسط کاربری (UI) استفاده می‌کند، نام برد.

بسیاری از موارد یاد شده به عنوان ایرادات به جاوا به عنوان ایرادات به طراحی زبان‌های سطح بالا هستند و نه جاوا.

یک اشتباه متداول:

برخی مردم به علت شباهت اسمی، جاوا و جاوااسکریپت را با هم اشتباه می‌گیرند. در حالیکه این دو زبان گرچه در ظاهر و کلمات شبیهند ولی بطور ساختاری با یکدیگر متفاوتند. جاوا اسکریپت محصول شرکت نت اسکیپ است. جاوا برای اجرا باید به زبان ماشین مجازی ترجمه شود اما جاوااسکریپت زبانی است که معمولا در صفحات وب نوشته می‌شود و توسط مرورگر تفسیر می‌گردد. در جاوا متغیرها همگی بر اساس نوعشان معرفی می‌شوند اما در جاواسکریپت نوع متغیرها به صورت ضمنی مشخص می‌شود.