

اجزای یک برنامه

۱- کلمات ذخیره شده

۲- شناسه ها

۳- ثابتها

۴- علائم

نحوه چیدن اجزای یک برنامه را نحو (SYNTAX) نام دارد.

کلمات ذخیره شده کلماتی هستند که منحصراً در اختیار زبان برنامه نویسی هستند و به منظور خاص در نظر گرفته شده و نمی توان به منظور دیگری از آن استفاده کرد مانند **Begin** ، **For** و ...

شناسه ها : شامل دو دسته استاندارد و غیر استاندارد می باشد.

شناسه های استاندارد شامل شناسه هایی هستند که از سوی پاسکال تعرف شده اند مانند (**Read**) پردازنده ها و توابع کتابخانه ای و نوع داده هایی که رزو شده نیستند و شناسه های غیر استاندارد شامل شناسه هایی هستند که از سوی برنامه نویس تعریف می شوند. مانند نام ثابت نام متغییر نام زیر برنامه پارامتر ...

ساختار برنامه

۱- عنوان Heading

۲- تعاریف و اعلانات Declaration

۳- بلاک Block

بخش عنوان با **Program** شروع می شود و در توربر پاسکال اختیاری است

بخش تعاریف و اعلانات : شامل تعریف برچسب ، ثابت، نوع و متغیر می باشد.

تعریف موارد فوق در پاسکال استاندارد الزامی است ولی در توربوپاسکال الزامی نیست.

بخش بلاک : این قسمت بین **Begin, end** قرار می گیرد.

بلاک اصلی شامل **Begin, end** می باشد. ابتدا اجرا می شود و در انتهای برنامه قرار دارد

بلاک فرعی شامل **Begin, end** می باشد.

برچسب : برای علامتگذاری در برنامه و تعیین مقصد برای پرش مورد استفاده قرار می گیرد

ثابتها : شناسه هایی که معرف مقدار ثابتی هستند از ثابتها معمولاً برای بالا بردن خوانایی و سادگی در ویرایش استفاده میشود

انواع ثابتها

حقیقی (ساده) : طبق تعریف ثابت در طول برنامه قابل تغییر نیست.

مجازی (نوعدار) : برای آن نوع تعیین می شود و می توان مقادیر آنرا در طول برنامه تغییر داد پس می

توان گفت نوعی متغیر است که مقدر اولیه دارد.

تعریف داده : هر نوع داده بیانگر مجموعه ای مشخص و محدود از مقادیر است بعضی از آنها استاندارد تعریف شده اند مانند **integer** ولی بعضی از آنها غیر استاندارد هستند که کاربر آنها را تعریف می کند.

متغیرها : مکانهایی از حافظه هستند که اطلاعات را به طور موقت نگهداری می کنند

انواع داده های عددی در پاسکال

صحیح **byte – long int- word – integer - shortint**

اعشاری **real**

ssingel – double- extended – comp ۸۰۸۷

در بین داده های **۸۰۸۷ come** عدد صحیح بسیار بزرگ ذخیره می کند و بقیه اعشاری برای نوشتن توضیحات در برنامه پاسکال می توان توضیحات را بین علامات فوق قرار دهید

{ ... } (الف)

(* ... *) (ب)

عملگرها :

عملگرهای یکتایی عملگرهایی هستند که یک **عملوند** دارند .

به تعداد عملوند های یک عملگر درجه عملگر می گویند.

عبارت ساده : عبارتی که تعدادی عملگرهای یک نوع در آن استفاده شده باشد

عبارت مرکب : از تعدادی عملگرمختلف تشکیل شده است

انواع دستورات :

ساده : دستوراتی که داخل آنها از دستورات دیگر استفاده نشده است مانند

• انتساب = :

• فراخوانی پرداز **Read**

• پرش بدون شرط **goto**

ساختار یافته : دستوراتی هستند که در آنها از دستورات دیگر استفاده شده است و دارای ساختار الگوریتمی است.

مانند:

دستورات تکرار (حلقه ها)

دستورات شرطی

دستور **with**

دستور مرکب : شامل تعدادی دستور که بین Begin, end قرار می گیرد.

برای جداسازی دستورات از علامت ؛ استفاده می شود پس می تواند در آخر دستور یا در یک خط جداگانه یا اول دستور بعد قرار گیرد.

تذکر : کلمات Begin ، end دستور العمل نیستند پس بعد از Begin و قبل از end نیاز به علامت ؛ نیست
یونیت مجموعه ای از تعاریف شامل نوع داده، ثابت، متغیر و زیر برنامه هاست :

- دارای پسوند tpu می باشد
- uses را نمی توان بیش از یکبار تکرار کرد
- در یونیت System تعاریف پر کاربرد مورد استفاده است مانند Read و ...
- یونیت System بطور خودکار توسط توربوپاسکال مورد استفاده قرار گیرد و نیازی به uses نیست .
- یونیت های معمول Dos , graph , crt می باشد.
- در یونیت CRT دستوراتی مانند Delay- gotoxy – clrscr - textcolr قرار دارند.
- در دستور read یا Readln اگر داده های داده شده کمتر از تعداد متغیرها باشد پاسکال منتظر وارد کردن بقیه می ماند.
- در Readln اگر تعداد داده ای ورودی بیشتر باشد داده های اضافی در نظر گرفته نمی شود
- در Read اگر تعداد داده ای ورودی بیشتر باشد داده های اضافی در حافظه می ماند و برای دستورات بعدی مورد استفاده قرار می گیرد.
- نوع داده وارد شده باید با نوع متغیر متناسب باشد.

دستورات شرطی : IF و CASE

عبارت مورد استفاده در CASE باید قابل شمارش و ترتیبی (اسکالر) و حداکثر دو بایتی باشد. داده اسکالر بعداً بررسی می شوند.

دستورات تکرار : FOR – WHILE – REPEAT

- FOR از این حلقه زمانی استفاده می شود که تعداد دفعات تکرار معین باشد
- متغیر شمارنده باید از نوع داده اسکالر باشد
- قرار دادن علامت ; بعد از Do سبب می شود که پایان حلقه بعد از Do باشد .
- برای While نیز چنین است.
- پس از پایان حلقه محتوای متغیر شمارنده برابر با مقداری نهایی شمارنده For می باشد.
- While : وقتی استفاده می شود که تکرار دستورات وابسته به شرطی باشد به عبارتی این حلقه تا زمانی تکرار می شود که عبارت جلو While مقدار True را نتیجه بدهد.
- Repeat در حلقه فوق نیازی به Begin , end نیست (برای چند دستور) بر خلاف حلقه های بالا .
- در هر شرایطی حلقه یکبار اجرا می شود زیرا شرط ادامه حلقه در آخر آن بررسی می شود
- حلقه در صورتی تکرار می شود که عبارت جلو Until غلط باشد
- در حلقه های متداخل تعداد تکرار دستورات داخل حلقه داخلی برابر با حاصلضرب دفعات تکرار هر کدام است البته به شرطی که حلقه ها وابسته نباشند.

زیر برنامه ها :

تابع	Function
پردازه	Procedure

مزایای زیر برنامه ها

- ۱- عیب یابی ساده تر است
- ۲- اعمال تغییرات و توسعه راحت تر است
- ۳- تقسیم مسئله به مسائل کوچکتر
- ۴- آزمایش مستقل هر قسمت برنامه
- ۵- تکرار دستورات کم می شود.

پردازه Procedure

- محل تعریف در قسمت تعاریف برنامه قبل از بلوک اصلی
- عنوان هر پردازه حداقل شامل نام پردازنده و کلمه Procedure می باشد.
- فراخوانی پردازه مانند دستورات (فقط نام و تعیین پارامترها)
- تعداد پارامترهای تعریف شده در پردازه - باید مساوی تعداد آنها در فراخوانی باشد.
- نوع اطلاعات تعریف شده در عنوان و فراخوانی پردازه باید متناسب (تطابق) باشد.
- می تواند چند ورودی و چند خروجی داشته باشد

تابع Function

- محل تعریف مانند پرداز ه.
 - عنوان تابع حداقل شامل کلمه **Function** و نام و نوع خروجی می باشد.
 - برای فرا خوانی تابع می توان آنرا مانند یک متغیر در یک عبارت و یا دستورات انتساب و یا یک دستور خروجی بکار برد.
 - تعداد پارامترها هنگام فراخوانی با تعریف یکسان باشد
 - نوع اطلاعات هنگام فراخوانی و تعریف باید متناسب باشد
 - چند ورودی و حداقل یک خروجی دارند.
 - معمولاً در انتهای تابع داده خروجی به نام تابع واگذار می شود پس تابع حداقل یک دستور دارد.
- تذکر :** تابع را می توان به عنوان ورودی به زیر برنامه ای دیگر ارسال کرد اما پرداز ه نمی توان ارسال کرد.

انواع متغیرها :

محلی Local: متغیرهای که داخل زیر برنامه تعریف می شوند و دامنه شناسایی آنها زیر برنامه و زیر برنامه های داخل آن است و مدت عمر کوتاهی دارد از ورود به پرداز ه (فراخوانی) تا زمانیکه زیر برنامه تمام می شود.

سراسری Global : متغیرهایی که در **var** اصلی برنامه تعریف می شوند و در همه جای برنامه و زیر برنامه ها قابل شناسایی هستند و عمر آنها مدت عمر برنامه است .

پارامترها :

- پل عبور داده ها بین برنامه اصلی و زیر برنامه میباشد
- عمر پارامترها مانند یک متغیر محلی است و فقط در داخل زیربرنامه شناخته می شوند.

انواع پارامتر

صوری (مقداری) **Value** : این پارامترها فقط ورودی هستند مانند راه یک طرفه و اگر در داخل زیر برنامه تغییر کنند در پارامتر نظیر برنامه اصلی تغییر ایجاد نمی شود

مرجع (متغیر) **Variable** : این پارامترها ورودی خروجی هستند و قبل از آن کلمه **var** قرار می گیرد با تغییر در زیر برنامه در پارامتر متناظر برنامه اصلی تغییر ایجاد می شود. در پارامتر مرجع هنگام فرا خوانی، پارامتر نظیر باید متغیر باشد. (ثابت – عبارت و ... نمی تواند باشد).

تذکر مهم : هنگام فراخوانی زیر برنامه پاسکال برای پارامترهای مقداری یک مکان جداگانه از حافظه را در نظر می گیرد در صورتیکه برای پارامترهای مرجعی به همان مکان پارامتر برنامه اصلی اشاره می کند بطور خلاصه اطلاعات فقط در یک مکان است

پارامتر واقعی (Actual) پارامترهای فراخوانی در برنامه اصلی و رسمی (Formal) در تعریف زیر برنامه گفته می شوند

رهنمودهای : \$N برای استفاده از داده های ۸۰۸۷ و \$E برای شبیه سازی کمک پردازنده ۸۰۸۷ می باشد.
خط جدید LF ۱۰ # - سر خط CR ۱۳ # - فضای خالی ۳۲ □ - بوق ۷ #

انواع داده های توربوپاسکال

long, word, integer, shortint, byte صحیح char کاراکتر Boolean (بولی) منطقی زیر قلمرو شمارشی	اسکالر	ساده
اعشاری		
ساختاری		
رشته ای		
اشاره گر		
پردازه ای		

داده های اسکالر داده هایی هستند که دارای ترتیب و شمارش پذیر باشند هر عنصر قبلی و بعدی دارد بجز اولین و آخرین عنصر

زیر قلمرو :

زیر مجموعه ای از مقادیر نوع داده اسکالر (میزبان) می باشد و متغیرهای زیر قلمرو تمام خصوصیات متغیر نوع میزبان را دارند و به اندازه میزبان فضا اشغال می کنند.

در تعریف زیر قلمرو اولین عنصر و آخرین باید ثابت بوده و اسکالر باشند و همچنین اولین باید کوچکتر یا مساوی آخرین باشد.

در زمان اجرای برنامه هیچ محدودیتی برای مقادیری که در زیر قلمرو قرار می گیرد (متغیر) وجود ندارد و می تواند مقادیر نوع میزبان خود را نگه دارد مگر اینکه از $\{ \$R+ \}$ استفاده شود که Range چک می شود و سبب کاهش سرعت اجرا می شود.

مزایای استفاده از زیر قلمرو:

- ۱- اگر بخواهید اشتباهاً خارج از محدوده مقداری را ذخیره کنید با پیام خطا کشف می شود.
- ۲- چون متغیر میزبان از سوی پاسکال انتخاب می شود پس در مصرف حافظه صرفه جویی می شود.
- ۳- خوانایی برنامه بالاتر می رود.

داده‌های شمارشی :

- به مجموعه ای مرتب از اسامی که هر یک هویت مستقل دارند ولی تحت یک عنوان قرار می گیرند :
- کاربرد آن فقط برای افزایش خوانایی برنامه است
 - حداکثر ۲۵۶ عضو می پذیرد با شماره های ۰-۲۵۵
 - هر تعداد عضو داشته باشند یک بایت اشغال می کند.
 - قابل خواندن و نوشتن مستقیم نیستند.

آرایه :

- تعدادی متغیر از یک نوع داده و تحت یک نام
- آرایه از نظر فیزیکی مکانهای متوالی از حافظه است و از نظر منطقی سطر یا ستون
 - عناصر آرایه از نظر تعداد ثابت است
 - آرایه نمی تواند خروجی تابع باشد.

پارامتر ثابت : برای جلوگیری از اشتباه و تغییر نکردن مقادیر آرایه قبل از پارامتر کلمه Const نوشته می شود و سبب می شود در زیر برنامه مقادیر آرایه تغییر نکند.

پارامتر باز : گاهی اوقات مشخص نیست که چه تعداد عناصر به زیر برنامه فرستاده میشوند به عبارتی آرایه چند عنصری به زیر برنامه فرستاده می شود پس با ذکر نکردن تعداد عناصر آرایه در تعریف زیر برنامه آنرا باز در نظر می گیرند و این تنها حالتی است که آرایه مستقیماً در عنوان تعریف میشود:

Function AM(A: Array of Real) : Real

رکورد

یک نوع داده ساختاری است که شامل مجموع های از عناصر (فیلد) غیر هم نوع است (البته همگی هم نوع می توانند باشند).

با کلمه Record (رزرو شده) شروع و با End خاتمه می پذیرد (نیازی به Begin ندارد)

انواع رکورد

طول ثابت : فضای اشغالی برابر با مجموع فضای فیلدها

طول متغیر : بزرگتری قسمت متغیر + نوع گونه + قسمت ثابت = فضای اشغالی

- رکورد نمی تواند خروجی یک تابع باشد
- رکورد می تواند ورودی زیر برنامه ها باشد.

لیست : به مجموعه ای از داده ها که در خاصیتی مشترک هستند

انواع جستجو :

- ترتیبی (خطی)
- دودویی (باینری)

در جستجوی خطی عنصر (کلید) یا داده ای که دنبال آن می گردیم از بالای لیست با تک تک عناصر مقایسه می شود یا عنصر پیدا میشود و یا اینکه به آخر لیست می رسیم و عنصر پیدا نمی شود.
 پس : نیاز نیست لیست مرتب باشد و در بدترین حالت n مقایسه انجام می شود پس اگر عنصر در وسط لیست باشد $\frac{n}{2}$ مقایسه صورت می گیرد.

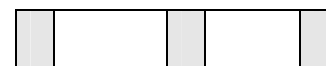
در روش دودویی :

- در این روش لیست باید به صورت نزولی یا صعودی مرتب باشد .
 - عمل مقایسه بجای اولین عنصر از عنصر وسط صورت می گیرد .
 - سرعت عمل این جستجو نسبت به ترتیبی بسیار بالاتر است .
 - در بدترین حالت $[\log_2^n] + 1$ مقایسه صورت می گیرد.
 - در این روش لیست به دو قسمت تقسیم شده و در صورت پیدا نشدن عنصر یکی از قسمتها حذف شده و لیست باقیمانده دوباره تقسیم می شود.
- اگر L ابتدا و R انتها و M وسط باشد

Key = A[M] عنصر پیدا شده

Key > A [M] L = M + ۱ حذف چپ

Key < A[M] R = M - ۱ حذف



L M R

جستجو تا زمانی ادامه پیدا می کند که یا عنصر پیدا شده باشد و یا اینکه $L > R$ در این صورت عمل جستجو قطع می شود.

(۱) در مرتب سازی حبابی هر عنصر با بعدی مقایسه می شود اگر جای آن مناسب نبود جابجا می شود

مرتب سازی حبابی

(۲) در هر گذر لیست کوچکتر می شود

(۳) تعداد مقایسه $\frac{n(n-1)}{2}$ می باشد

(۱) کوچکترین عدد را پیدا کرده و با ابتدای لیست جابجا می کند.

(۲) هر بار لیست کوچکتر می شود

(۳) تعداد مقایسه $\frac{n(n-1)}{2}$ می باشد

مرتب سازی انتخابی

(۴) این روش سریعتر است زیرا تعداد جابجایی ها کمتر می باشد

فایل : به مجموعه ای از اطلاعات مرتبط که روی حافظه جانبی ذخیره می شوند فایل گویند.

امکان ایجاد فایل های زیر در توربو پاسکال است (متنی - نوع عددی - بدون نوع)

- در فایل متنی پایان هر خط کارکترهای LF , CR قرار می گیرند.
- عمل انتساب شناسه و نام فایل کافی است یکبار صورت گیرد.
- اگر فایل موجود نباشد و RESET کنیم با پیام خطا مواجه می شویم.
- عددی که بوسیله تابع Ioresult برگردانده میشود نتیجه وضعیت و دلیل باز شدن و یا نشدن فایل می باشد در باز شدن فایل صفر باز می گرداند.
- با عمل بستن فایل Close عمل Assign دوباره باید صورت گیرد (ارتباط قطع می شود).

گرافیک :

هر نقطه در صفحه گرافیک که امکان دستیابی به آن است پیکسل می گویند.

هر چه تعداد پیکسلها بیشتر باشد اندازه آنها کوچکتر است در نتیجه کیفیت بیشتری دارد.

پردازه `initgraph` عمل مقدار دهی و تنظیم کارت در وضعیت های مختلف گرافیکی را فراهم می کند.

درایورهای گرافیکی در فایل هایی با پسوند `BGI` و در شاخه `BGI` هستند .

فرمان های گرافیکی ثابتها و ... در یونیت `graph.tup` قرار دارد.

توربو پاسکال در حالت معمول حداکثر `VGA` و `۱۶ * ۴۸۰ * ۶۴۰` را پشتیبانی می کند در صفحه نمایش

گرافیکی رنگ چشمک زن وجود ندارد.

پس از ورود به صفحه نمایش گرافیکی مختصات جاری (۰ و ۰) است .

تعریف شکل ظاهری هر فونت در فایل جداگانه ای با پسوند `chr` است در شاخه `bgi`

کلیدها و منوهای مورد استفاده

کلید `F10` و یا حرف اول `ALT +` منوی مورد نظر را انتخاب می کند.

کلید `F2 = File - save` برای ذخیره کردن

کلید `F3 = File - open` باز کردن

کلید `F5 = windows - zoom` پنجره را به اندازه تمام محیط باز می کند

`ALT + F9 = Compile - compile` برای کامپایل کردن برنامه (اجرا نمی شود) فقط ترجمه

`CTRL + F5 = Run - Run` در آخر برنامه برای دیدن خروجی برنامه

`ALT + F3 = Windows - close` بستن غیر جاری

`File = Exit = ALT+X` خروج از پاسکال

`Edit - copy = Ctrl + ins` کپی برداری

`Edit - cut = Ctrl + Dele` عمل Cut

- Past عمل Edit – Paste = Shift + ins
- Windows – next = F۶ پنجره بعدی
- Windows – Previus = Shift + F۶ پنجره قبلی
- Rum – program reset = F۷ قطع برنامه ای که در حلقه افتاده
- Run – Trace into=F۷ اجرای خط به خط برنامه داخل پردازنده و تابع می رود
- Run – Stepover = F۸ اجرای قدم به قدم برنامه داخل پردازنده و تابع نمی رود
- Run – Goto cursor = F۴ برنامه تا خطی که کرسر در آن قرار دارد اجرا می شود
- Debug – Add watch = Ctrl+F۷ برای دیدن محتویات متغیرها
- Windowd – size – move = Ctrl + F۵ تغییر اندازه و حرکت پنجره
- Ctrl +F۱ راهنمای مربوط به دستور خاص که کرسر زیر آن است
- = F۱ راهنما
- Debug –Call stack = Ctrl + F۳ محتوای پشته را نمایش می دهد

اگر نوع متغیر سمت چپ با نوع سمت راست در دستور واگذاری مطابقت نداشته باشد. پیام خطای

Type Mismatch

اگر در ساختار دستور IF دستور قبل از Else علامت ؛ بگذاریم پاسکال دستور IF را خاتمه یافته حساب می کند پس پیام خطای:

Error in statement

پیام خطای Constant out of range وقتی صادر می شود که بخواهیم خارج از محدوده زیر قلمرو مقدار دهی کنیم البته زمانی که { \$R+ } باشد

زمانی که سعی کنیم اطلاعات شمارش را از صفحه کلید بخواهیم پیام :

cannot read or write variables of this type

اگر فایل موجود نباشد با اجرای دستور Append , Reset پیام خطای

File not found

اگر دستور Read اجرای شود و نشانگر فایل خارج از محدوده باشد پیام

Disk Read Error

This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.