

به نام خدا

طراحی زبان‌های برنامه‌سازی

مرجان عظیمی
Azimi.marjan@gmail.com

فصل نهم

کنترل زیر برنامه

کنترل ترتیب زیر برنامه



- زیربرنامه ساده فراخوانی - برگشت: هر برنامه متشکل از یک برنامه اصلی است که در حین اجرا می تواند زیربرنامه هایی را فراخوانی کند و هر زیربرنامه زیربرنامه های دیگر را.
- دستور فراخوانی زیربرنامه مثل این است که قبل از اجرا یک کپی از زیربرنامه در نقطه ای که فراخوانی صورت می گیرد قرار داده شود.

کنترل ترتیب زیر برنامه (ادامه)



زیربرنامه های فراخوانی - برگشت

- ☐ پیاده سازی: نیاز به چیزهای دیگر:
- ☐ بین تعریف زیربرنامه و سابقه فعالیت آن تفاوت وجود دارد.
- ☐ سابقه فعالیت دو بخش دارد : سگمنت کد و رکورد فعالیت
- ☐ سگمنت کد در حین اجرا تغییر نمی کند.
- ☐ رکورد فعالیت در هر بار اجرای زیربرنامه ایجاد می شود و با خاتمه زیربرنامه از بین می رود.

کنترل ترتیب زیر برنامه (ادامه)



زیربرنامه های فراخوانی - برگشت (ادامه)

پیاده سازی پشته ای

- ☐ ساده ترین تکنیک مدیریت حافظه زمان اجرا پشته است.
- ☐ برای کنترل مدیریت حافظه نیاز به اشاره گر پشته است.
- ☐ در پاسکال یک پشته مرکزی و یک ناحیه حافظه به طور ایستا تخصیص می یابد.
- ☐ پشته در لیسپ به صورت فراخوانیهای زیربرنامه به صورت تو در تو می باشد.

کنترل ترتیب زیر برنامه (ادامه)



زیربرنامه های بازگشتی

- ☐ مشخصات: اگر فراخوانی بازگشتی زیربرنامه امکانپذیر باشد A می تواند هر زیربرنامه ای از جمله خودش را فراخوانی کند.
- ☐ پیاده سازی: در هنگام فراخوانی هر زیربرنامه رکورد فعالیت جدیدی ایجاد می شود و با دستور برگشت از بین می رود.

صفات کنترل داده ها



اسامی و محیطهای ارجاع

☐ اشیای داده به دو روش به عنوان عملوند یک عملیات مورد استفاده قرار می گیرند:

☐ انتقال مستقیم

☐ مراجعه از طریق شی داده ای که دارای نام است.

صفات کنترل داده ها (ادامه)



اسامی و محیط‌های ارجاع (ادامه)

☐ عناصری از برنامه که دارای نام هستند (عناصر مشترک):

- ☐ اسامی متغیرها
- ☐ اسامی پارامترهای مجازی
- ☐ اسامی زیربرنامه ها
- ☐ اسامی انواع تعریف شده
- ☐ اسامی ثوابت تعریف شده
- ☐ برچسب دستورات
- ☐ اسامی استثناها
- ☐ اسامی عملیات اولیه مثل + و * و sort
- ☐ اسامی ثوابت لیترال مثل ۲۵/۳ و ۱۷

صفات کنترل داده ها (ادامه)



اسامی و محیطهای ارجاع (ادامه)

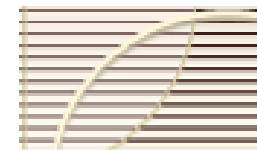
وابستگیها و محیطهای ارجاع

□ محیط ارجاع: مجموعه ای از وابستگی ها که در طول اجرای یک زیر برنامه وجود دارند. و برای آن قابل دسترس می باشند.

□ کنترل داده ها به انقیاد شناسه ها به اشیای داده زیر برنامه ها مربوط می شود.

□ این انقیاد را وابستگی می نامند و ممکن است به صورت جفتی از شناسه و شی داده یا زیر برنامه مربوط به آن نمایش داد.

صفات کنترل داده ها (ادامه)



اسامی و محیطهای ارجاع (ادامه)
وابستگیها و محیطهای ارجاع (ادامه)

- ☐ در حین اجرای برنامه در اغلب زبانها مشاهده می شود که:
- ☐ در آغاز اجرای برنامه اصلی وابستگی شناسه ها ، نام هر متغیر تعریف شده در برنامه را به شی داده مقید می سازند و نام هر زیر برنامه فراخوانده شده را به یک تعریف زیر برنامه مقید می سازند.
- ☐ وقتی برنامه اصلی اجرا می شود عملیات ارجاعی را فراخوانی می کند .
- ☐ هر وقت زیربرنامه جدید فراخوانی می شود وابستگیهای دیگری برای زیر برنامه ایجاد می شود.
- ☐ وقتی زیربرنامه کنترل را به برنامه اصلی برمی گرداند وابستگیهای آن از بین میروند.
- ☐ وقتی کنترل به برنامه اصلی برمیگردد اجرا مشابه قبل ادامه می یابد.

صفات کنترل داده ها (ادامه)



اسامی و محیطهای ارجاع (ادامه)
وابستگیها و محیطهای ارجاع (ادامه)

- ☐ مفاهیم اصلی کنترل داده:
- ☐ محیطهای ارجاع
 - ☐ محیط ارجاع محلی (یا محیط محلی)
 - ☐ محیط ارجاع غیرمحلی
 - ☐ محیط ارجاع عمومی
 - ☐ محیط ارجاع از پیش تعریف شده
- ☐ قابلیت مشاهده
- ☐ حوزه پویا

صفات کنترل داده ها (ادامه)



اسامی و محیطهای ارجاع (ادامه)

نام مستعار برای اشیای داده

- ❑ یک شی داده در طول عمرش ممکن است بیش از یک نام داشته باشد یعنی ممکن است چندین وابستگی در محیطهای ارجاع متفاوت هر کدام نام متفاوتی را برای شی داده‌ای ایجاد کنند.
- ❑ به دلیل مشکلاتی که نام مستعار ایجاد می کند طراحی زبان جدید سعی در حذف یا محدود کردن نام مستعار دارد.

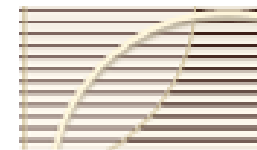
صفات کنترل داده ها(ادامه)



حوزه ایستا و پویا

- ☐ حوزه پویای وابستگی مربوط به یک شناسه مجموعه ای از سابقه های فعالیت زیربرنامه است که وابستگی در حین اجرا قابل مشاهده است.
- ☐ قاعده حوزه پویا : حوزه پویای هر وابستگی را برحسب حالت پویای اجرای برنامه تعریف می کند.
- ☐ اهمیت حوزه ایستا: اغلب فرآیندها یکبار در زمان ترجمه انجام شوند.

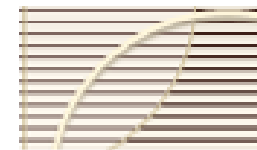
صفات کنترل داده ها(ادامه)



ساختار بلوکی

- ☐ مفهوم ساختار بلوک در زبانهای ساخت یافته بلوکی مثل پاسکال پیدا شد.
- ☐ هر زیربرنامه یا برنامه به صورت مجموعه ای از بلوکهای تودرتو سازماندهی می شود.
- ☐ ویژگی مهم بلوک : محیط ارجاع جدیدی را معرفی میکند.
- ☐ با مجموعه ای از اعلان ها برای اسامی شروع می شود و سپس مجموعه ای از دستورات قرار می گیرد که به آن اسامی مراجعه می کنند.

صفات کنترل داده ها(ادامه)



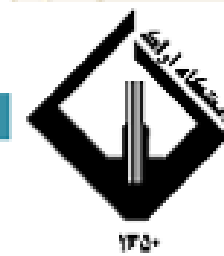
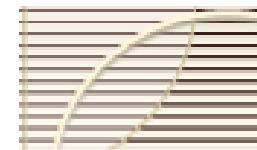
داده های محلی و محیطهای ارجاع محلی

دو پیاده سازی متفاوت برای محیط ارجاع محلی:

❑ نگهداری: وابستگی در هنگام پایان یافتن یک زیر برنامه نگهداشته شده و از بین نمی روند.

❑ حذف: وابستگی های محیط ارجاع محلی در هنگام پایان یافتن یک زیر برنامه از بین رفته و حذف می شوند و در هر ورود مجدداً ایجاد می شوند.

پارامترها و انتقال پارامترها



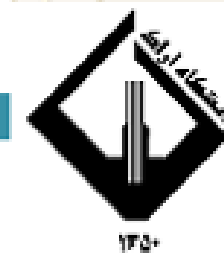
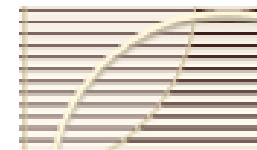
تبادل اطلاعات بین زیربرنامه‌ها:

محیط‌های مشترک صریح: داده‌های اشتراکی باید صراحتاً مشخص گردند و زیربرنامه‌هایی که از آن‌ها استفاده می‌کنند نیز مشخص شوند. محیط‌های اشتراکی ضمنی: داده‌ها بین زیربرنامه‌ها مشترکند ولی تعریف صریحی از آن‌ها انجام نمی‌شود.

حوزه پویا

حوزه ایستا

پارامترها و انتقال پارامترها



پارامتر: مکانیزمی برای تبادل اطلاعات صریح بین فقط دو زیربرنامه است.

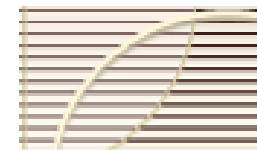
پارامترهای مجازی و واقعی

اصطلاحات آرگومان و نتیجه به داده هایی اطلاق می شود که با مکانیزمهای مختلفی به زیربرنامه ارسال و از آن دریافت می شود.

پارامترهای مجازی: در تعریف برنامه ذکر می شود.

پارامترهای واقعی: مقداری که به آن در هنگام فراخوانی نسبت داده می شود.

پارامترها و انتقال پارامترها



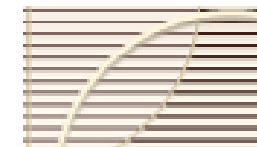
پارامترهای مجازی و واقعی (ادامه)

تناظر بین پارامترهای مجازی و واقعی

تناظر موقعیتی

تناظر براساس نام

پارامترها و انتقال پارامترها



روشهای انتقال پارامترها

فراخوانی با نام

فراخوانی با ارجاع

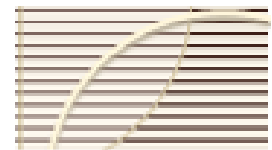
فراخوانی با مقدار

فراخوانی با مقدار و نتیجه

فراخوانی با مقدار ثابت

فراخوانی با نتیجه

پارامترها و انتقال پارامترها



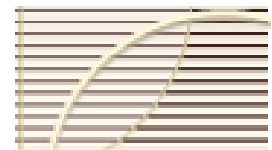
روشهای انتقال پارامترها

فراخوانی با نام: وقتی پارامترها از طریق نام ارسال می شوند، پارامترهای واقعی از نظر متنی، به جای پارامتر مجازی در داخل زیربرنامه قرار می گیرد.

فراخوانی با ارجاع: در این روش، مسیر دستیابی (آدرس) به زیربرنامه ارسال می شود.

فراخوانی با مقدار: در آن مقادیر واقعی کپی می شوند.

پارامترها و انتقال پارامترها



روشهای انتقال پارامترها

فراخوانی با مقدار و نتیجه: مقدار پارامتر واقعی در پارامتر مجازی کپی می شود. هنگام خاتمه زیربرنامه مقدار محتویات نهایی پارامتر مجازی در پارامتر واقعی کپی می شود.

فراخوانی با مقدار ثابت: هیچ تغییری در پارامتر مجازی هنگام اجرای برنامه مجاز نمی باشد.

فراخوانی با نتیجه: مقدار اولیه پارامتر واقعی تفاوتی نمی کند و نمی تواند توسط زیر برنامه استفاده گردد. هنگام خاتمه زیر برنامه مقدار نهایی پارامتر مجازی به پارامتر رسمی ارسال می شود.

پارامترها و انتقال پارامترها



پیاده سازی انتقال پارامتر

چون هر سابقه فعالیت زیربرنامه مجموعه متفاوتی از پارامترها را دریافت می کند. حافظه پارامترهای مجازی زیربرنامه به عنوان بخشی از رکورد فعالیت زیربرنامه تخصیص می یابد هر پارامتر مجازی یک شی داده محلی در زیربرنامه است.

پارامترها و انتقال پارامترها



پیاده سازی انتقال پارامتر

مثالهایی از انتقال پارامترها

متغیرهای ساده و ثوابت

ساختمان داده ها

عناصر ساختمان داده ها

عناصر آرایه با اندیسهای محاسبه شده

اشاره گرها

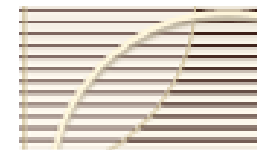
نتایج عبارات

نام مستعار و پارامترها

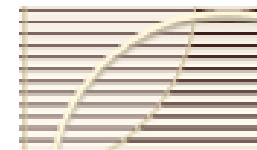
محیط‌های مشترک صریح

مشخصات: محیط مشترک معادل محیطی برای یک زیربرنامه است با این تفاوت که بخشی از یک زیربرنامه خاص نیست. به مجموعه‌ای از اشیا که باید بین مجموعه‌ای از زیربرنامه‌ها به اشتراک گذاشته شوند، حافظه‌ای در بلوک‌های مجزای دارای نام تخصیص داده می‌شود.

پیاده سازی: در فرترن و C هر زیربرنامه ای که از محیط مشترک استفاده می کند اعلان‌هایی برای متغیرهای مشترک دارد .



محیط‌های مشترک صریح

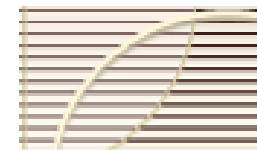


اشتراک صریح متغیرها

به جای اینکه گروهی از متغیرها در محیط مشترک و جدا از زیربرنامه ها باشند هر متغیر دارای یک مالک است و آن زیربرنامه است که در آنجا اعلان می شود. به منظور قابل رویت نمودن متغیر محلی در خارج از زیر برنامه تعریف صریح برای ارسال آن باید انجام شود.

پیاده سازی: اثر اشتراک صریح متغیر مشابه استفاده از یک متغیر در محیط مشترک است .

محیط‌های مشترک صریح



حوزه پویا

قاعده تازه ترین وابستگی: در زنجیره پویایی از فراخوانی زیربرنامه ها که از P شروع شد از تازه ترین وابستگی ایجاد شده برای X استفاده می کنیم .

پیاده سازی: پیاده سازی آن با توجه به پیاده سازی پشته مرکزی برای ذخیره رکوردهای فعالیت زیربرنامه ساده است.

محیط‌های مشترک صریح



حوزه ایستا و ساختار بلوکی

قوانین محدوده ایستا برای پیاده‌سازی توسط کامپایلر مشخص هستند.

برای پیاده‌سازی کامل لازم است ساختار بلوک ایستا در حین اجرا طوری نمایش داده شود که بتواند ارجاع غیرمحلی را کنترل کند.

محیط‌های مشترک صریح



حوزه ایستا و ساختار بلوکی (ادامه)

پیاده سازی زنجیر ایستا

اشاره گر زنجیر ایستا همیشه حاوی آدرس پایه جدول محلی دیگری است که در محل پایینتر جدول قرار دارد.

اشاره گرهای زنجیر ایستا مبنایی برای الگوی ارجاع است.