

فصل ۱۶: طراحی در سطح مؤلفه

طراحی در سطح مؤلفه که «طراحی رویه‌ای» نیز نام دارد، پس از تعیین طراحی داده‌ها، معماری و واسط انجام می‌گیرد و هدف تبدیل مدل طراحی به نرم‌افزار عملیاتی می‌باشد. اما سطح انتزاعی مدل طراحی موجود، نسبتاً زیاد بوده و میزان انتزاع برنامه عملیاتی کم است. تبدیل و برگردان، سخت و دشوار بوده و موجب ایجاد خطاهای ظریفی می‌شود که یافتن و تصحیح آنها در مراحل بعدی فرآیند نرم‌افزاری دشوار است.

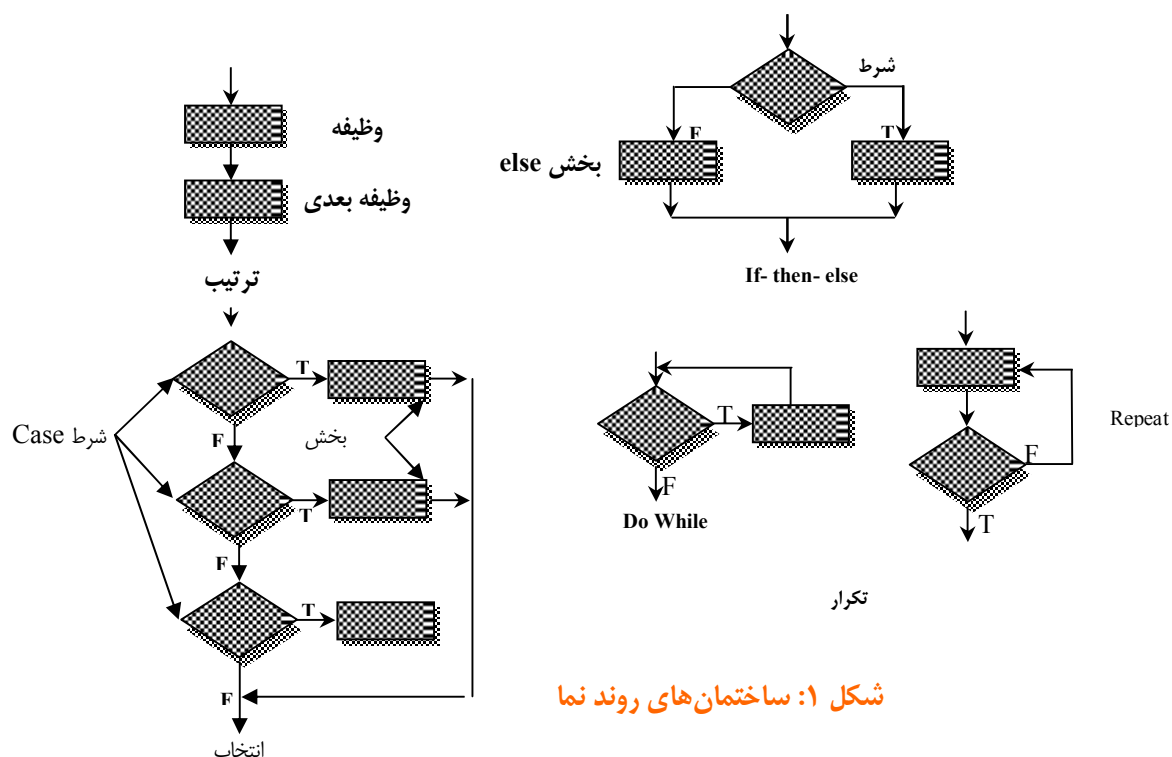
نمایش طراحی لایه‌ای با استفاده از یک زبان برنامه‌سازی امکان‌پذیر است. اساساً برنامه با به کارگیری مدل طراحی به عنوان یک راهنما، ایجاد می‌گردد. روش دیگر نمایش طراحی رویه‌ای با استفاده از نمایش واسطه (مثلاً گرافیکی، جدولی یا متنی است) که به راحتی قابل تبدیل به برنامه منبع می‌باشد. صرف‌نظر از مکانیزم به کار رفته در نمایش طراحی رویه‌ای، ساختمان‌های داده‌ای، واسط‌های و الگوریتم‌های مشخص شده باید با مجموعه‌ای از رهنمودهای طراحی مطابقت داشته باشند تا ضمن پیشرفت و تکمیل طراحی رویه‌ای، مانع خطا گردند.

برنامه سازی ساختیافته

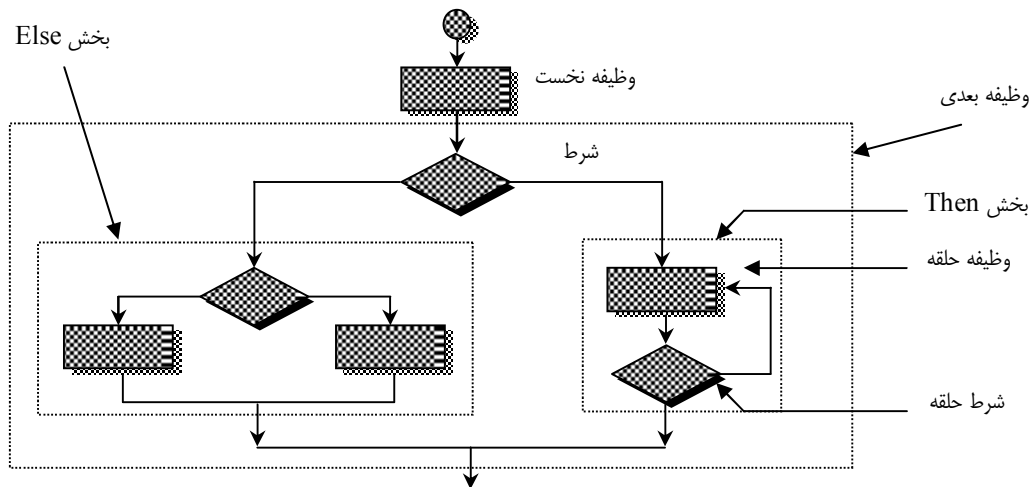
استفاده از ساختارهای منطقی پیشنهاد محققین می‌باشد که هر برنامه ای را با آن‌ها می‌توان نوشت. این ساختارها عبارتند از **توالی**، **شرط و تکرار**، **توالی**، مراحل پردازشی ضروری در تعیین هر الگوریتم را اجرا می‌کند. شرط، امکان پردازش انتخابی بر اساس رخداد منطقی را فراهم می‌کند و **تکرار**، ایجاد حلقه را امکان‌پذیر می‌سازد. این سه سازه در برنامه‌سازی ساختیافته که یک تکنیک مهم طراحی رویه می‌باشد، اساسی و ضروری به شمار می‌روند.

نشانه گذاری طراحی گرافیکی

هیچ تردیدی نیست که ابزارهای گرافیکی مانند روند نما (Flowchart) یا نمودارهای مستطیلی، الگوهای تصویری مفیدی هستند که به سادگی جزئیات رویه‌ای را مصور می‌سازند. هر چند در صورت کاربرد غلط ابزارهای گرافیکی، تصویر اشتباه ممکن است به نرم‌افزاری نادرست منجر گردد.

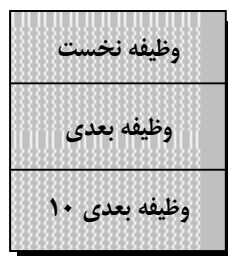


شکل ۱: ساختمان‌های روند نما

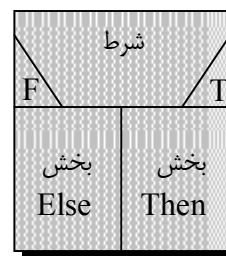


شکل ۲: ساختمانهای تو در تو

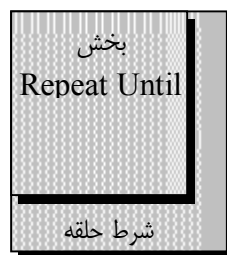
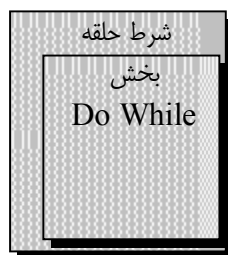
نمایش گرافیکی سازه‌های ساختیافته با استفاده از نمودار مستطیلی (یا جعبه‌ای) در شکل ۳ نشان داده شده است. عنصر اصلی نمودار یک کادر یا جعبه است. برای نمایش توالی، دو مستطیل از پایین به بالا به یکدیگر متصل می‌شوند. برای نمایش if-Then-else، مستطیل به دنبال Condition قرار می‌گیرد. تکرار با یک الگوی محدودکننده که فرآیند قابل تکرار را محصور می‌کند. (Repeat-Until-Part DO-While-Part) به نمایش در می‌آید. در نهایت، انتخاب با به کارگیری فرم گرافیکی نشان داده شده در پایین تصویر، به نمایش در می‌آید. نمودار کادری نیز مانند فلوجارت‌ها به هنگام پالایش عناصر پردازشی یک پیمانه، در چندین صفحه لایه‌بندی می‌شود. یک فراخوانی پیمانه توسط کادری در داخل پیمانه که نام آن درون یک بیضی محصور شده، قابل نمایش است.



ترتیب



If- then - else



تکرار



انتخاب

شکل ۳: ساختمان های نمودار جعبه‌ای

علایم طراحی جدولی

در بسیاری از کاربردهای نرم افزاری، به منظور ارزیابی ترکیب پیچیده شرایط و انتخاب اعمال مناسب مبتنی بر این حالتها، ممکن است به پیمانه ای نیاز باشد. جداول تصمیم گیری، علایمی را به کار می برد که اعمال و شرایط (توصیف شده در شرح پردازش) را به شکل جدولی تبدیل می کند. تعبیر غلط جدول، دشوار است و حتی ممکن است به عنوان ورودی برای یک الگوریتم قابل خواندن ولی برای یک دستگاه قابل خواندن نباشد.

سازمان دهی جدول تصمیم گیری در شکل ۴ به نمایش درآمده است. مطابق شکل، این جدول به چهار بخش تقسیم می شود. یک چهارم سمت چپ بالایی شامل لیستی از تمامی شرایط است. یک چهارم سمت چپ پایینی لیستی از تمامی اعمال و اقدامات ممکن و مبتنی به ترکیب شرایط را در بردارد. یک چهارم های سمت راست، ماتریسی را تشکیل می دهند که نشان دهنده ترکیب شرایط و اعمال و اقدامات متناظر با یک ترکیب خاص هستند. هر ستون ماتریس ممکن است به عنوان یک قانون پردازشی تفسیر شود.

در ایجاد جدول تصمیم گیری، مراحل زیر طی می شود:

- ۱- تمامی اعمال مرتبط با یک رویه (یا پیمانه) خاص را فهرست کنید.
- ۲- تمامی شرایط (یا تصمیمات اتخاذ شده) طی اجرای رویه را لیست نمایید.
- ۳- مجموعه خاص شرایط را با اعمال و اقدامات بخصوصی که ترکیبات غیر ممکن شرایط را برطرف می سازند، مرتبط نمایید یا این که هرگونه جابه جایی ممکن در شرایط ایجاد کنید.
- ۴- با بیان اعمال انجام شده در ارتباط با مجموعه شرایط، قوانین را تعریف کنید.

شرایط	۱	۲	۳	۴				n
شرط شماره ۱	✓			✓	✓			
شرط شماره ۲		✓		✓				
شرط شماره ۳			✓		✓			
شرط شماره ۴								
شرط شماره ۵								
اقدام شماره ۱	✓			✓	✓			
اقدام شماره ۲		✓		✓				
اقدام شماره ۳			✓					
اقدام شماره ۴			✓	✓	✓			
اقدام شماره ۵	✓	✓			✓			

شکل ۴: ساختار جدول تصمیم گیری

نمونه ای دیگر از جدول تصمیم گیری

شرایط	۱	۲	۳	۴	۵
نرخ ثابت حساب	T	T	F	F	F
نرخ متغیر حساب	F	F	T	T	F
مصرف > ۱۰۰ کیلو وات ساعت	T	F	T	F	
مصرف <= ۱۰۰ کیلو وات ساعت	F	T	F	T	
اقدامات					
حداقل هزینه ماهانه					
صورت حساب زمانبندی الف					
صورت حساب زمانبندی ب					
دیگر اقدامات					

شکل ۵: ساختار جدول تصمیم گیری در مورد مصرف برق

زبان طراحی برنامه (PDL)

الف) " زبان طراحی برنامه (PDF) که انگلیسی ساخت یافته یا شبه کد نیز نام دارد " یک زبان آمیخته است به طوری که واژگان یک زبان (یعنی انگلیسی) و نحوه کلی زبانی دیگر (یعنی زبان برنامه سازی ساخت یافته) را به کار می برد. **ب)** یک زبان طراحی باید ویژگیهای زیر را داشته باشد:

- ♦ یک نحو ثابت از کلمات کلیدی که تمام سازه های ساخت یافته، تعاریف داده ها و ویژگیهای پیمانه ای شدن را تهیه کند.
- ♦ یک نحو آزاد از زبان طبیعی که ویژگیهای پردازشی را تشریح کند.
- ♦ تسهیلات تعریف داده ها که باید مشتمل بر ساختارهای داده ای ساده (اسکالر، آرایه) و پیچیده (لیست پیوندی یا درخت) باشد.
- ♦ تعریف زیر برنامه و فنون فراخوانی که حالات مختلف توصیف رابط را پشتیبانی کند.

```

PROCEDURE security.monitor;
INTERFACE RETURNS system.status;
TYPE signal 1S STRUCTURE DEFINED
name 1S STRING LENGTH VAR;
address 1S HEX device location;
bound. Value IS upper bound SCALAR;
message IS STRING LENGTH VAR;
    END signal TYPE;
    TYPE system.status IS BIT (4);
    TYPE alarm. Type DEFINED
    
```

Smoke.alarm IS INSTANCE OF signal;
 Fire.alarm IS INSTANCE OF signal;
 Water.alarm IS INSTANCE OF signal;
 temp.alarm IS INSTANCE OF signal;
 burglar.alarm IS INSTANCE OF signal;
 TYPE phone.Number IS area code + 7-digit number;

ج) نشان گذاری طراحی باید به نمایش رویه‌ای منجر شود که درک و بررسی آن آسان باشد. به علاوه، نشان گذاری باید توانایی "code to" را تقویت کند به طوری که برنامه یا کد در واقع به محصول جانبی و طبیعی طراحی تبدیل شود. و نهایتاً این که نمایش طراحی بایستی به راحتی قابل نگهداری باشد به نحوی که طراحی همواره به طور صحیح نشان دهنده برنامه باشد.

مقایسه نشانه گذاری های طراحی

خصوصیات زیر در مورد نشان گذاری طراحی در زمینه مشخصات کلی فوق الذکر تعیین شده‌اند:
قابلیت پیمانه‌ای: نشان گذاری طراحی باید ایجاد نرم افزار پیمانه‌ای را حمایت کرده و شیوه‌ای برای تعیین رابط را فراهم آورد.

سادگی همه جانبه: یادگیری نشانه گذاری طراحی باید تقریباً آسان بوده و کاربرد آن نیز نسبتاً راحت باشد و به طور کلی از لحاظ خواندن نیز دشوار نباشد.

سهولت ویرایش: ضمن پیشرفت روند نرم افزار، طراحی رویه‌ای ممکن است نیاز به اصلاح و تغییر داشته باشد. سهولت و راحتی در ویرایش نمایش رویه‌ای، موجب تسهیل وظایف مهندسی نرم افزار گردد.

قابلیت خواندن سیستم: نشان گذاری که بتواند مستقیماً ورودی یک سیستم توسعه کامپیوتری باشد، مزایای قابل توجهی را به همراه دارد.

قابلیت نگهداری: نگهداری نرم افزار پر هزینه‌ترین مرحله در دوره زندگی نرم افزاری است. نگهداری پیکربندی نرم افزار تقریباً همواره به معنای نگهداری از نمایش طراحی رویه‌ای می‌باشد.

تقویت ساختار: مزایای یک رهیافت طراحی که از مفاهیم برنامه نویسی ساخت یافته را به کار می‌برد، قبلاً مورد بحث قرار گرفته است. نشان گذاری طراحی که کاربرد صرف سازه‌های ساخت یافته را تقویت می‌کند، یک شیوه خوب طراحی را توسعه می‌دهد.

پردازش خودکار: طراحی رویه‌ای دربردارنده اطلاعاتی است که قابل پردازش بوده و می‌تواند درباره صحت و کیفیت طراحی، بینش و درک بهتر یا جدیدی را در اختیار طراح قرار دهد. این شناخت از طریق گزارش‌های حاصل از ابزارهای طراحی نرم افزار، قابل بهبود و تقویت می‌باشد.

بازنمایی داده‌ها: توانایی نمایش داده‌های محلی و سراسری، عنصر اصلی و ضروری طراحی سطح مؤلفه‌هاست. در وضعیت ایده‌آل، نشان گذاری طراحی باید این داده‌ها را به طور مستقیم نمایش دهد.

تأیید منطق: تأیید خودکار منطق طراحی هدفی مهم در مرحله آزمون نرم افزار می‌باشد. نشان گذاری که باعث تقویت توانایی تأیید منطق شود، کفایت آزمون را به شدت افزایش می‌دهد.

توانایی تبدیل به برنامه: "وظیفه بعدی مهندس نرم افزار پس از طراحی اجزاء، تولید برنامه است. نشان گذاری که به راحتی قابل تبدیل به برنامه منبع باشد، میزان تلاش و خطا را کاهش می‌دهد.

تست‌های فصل شانزده: طراحی در سطح مولفه

- ۱- کدام طراحی از نظر زمانی در آخرین مرحله انجام می‌گیرد؟
 (الف) طراحی داده‌ها
 (ب) طراحی معماری
 (ج) طراحی رویه‌ای
 (د) طراحی واسط
- ۲- سطح انتزاع در کدام طراحی پایین‌تر است؟
 (الف) طراحی داده‌ها
 (ب) طراحی معماری
 (ج) طراحی رویه‌ای
 (د) طراحی واسط
- ۳- کدام یک از موارد زیر جزء ساختارهای منطقی پیشنهادی دیکسترا (Dijkstra) برای نوشتن برنامه نمی‌باشد؟
 (الف) Sequence
 (ب) State
 (ج) Conditional
 (د) Loop
- ۴- پرهزینه‌ترین مرحله از چرخه حیات نرم‌افزار کدام است؟
 (الف) طراحی داده‌ها
 (ب) طراحی معماری
 (ج) طراحی رویه‌ای
 (د) نگهداری آن
- ۵- کدام یک از ویژگی‌های نمایش نمودار مستطیلی در مقایسه با نمایش نمودار گردش (روند نما) وجود ندارد؟
 (الف) سادگی
 (ب) قابل فهم
 (ج) غیر ممکن بودن انتقال اختیاری کنترل
 (د) به سهولت تعیین کردن دامنه داده‌ها
- ۶- کدام یک از موارد زیر از ویژگی‌های PDL نمی‌باشد؟
 (الف) سادگی
 (ب) قابل فهم بودن
 (ج) شبه کد بودن
 (د) کامپایلری بودن
- ۷- کدام یک از مراحل زیر برای توسعه یک جدول تصمیم‌گیری اجرا نمی‌شود؟
 (الف) لیست کردن عملیاتی که به یک رویه مشخص ربط دارد.
 (ب) لیست کردن کلیه شرطها
 (ج) توسعه هر جای گشت ممکن از شرطها
 (د) تعیین قواعد با مشخص کردن این که چه عمل‌هایی برای یک مجموعه از شرایط رخ می‌دهد.
- ۸- کدام یک از موارد زیر پایه‌ای برای تهیه برنامه‌ای ساختیافته نیست؟
 (الف) برگشتی
 (ب) شرطی
 (ج) تکرار
 (د) توالی
- ۹- کدام یک از موارد زیر یک نمایش گرافیکی برای توصیف رویه‌ای است؟
 (الف) نمودار مستطیلی
 (ب) جدول
 (ج) نمودار ER
 (د) ماتریس گرافیک
- ۱۰- به‌طور کلی نمودارهای مستطیلی و روند نماها فلوجارت‌ها باید
 (الف) به جای زبان‌های طراحی برنامه‌نویسی استفاده شوند.
 (ب) برای مستندسازی کامل طراحی استفاده شوند یا اصلاً نشوند.
 (ج) فقط برای مستندسازی و یا ارزیابی طراحی در یک مورد خاص استفاده شوند.
 (د) هیچ یک از موارد فوق
- ۱۱- یک جدول تصمیم در موارد زیر استفاده می‌شود.
 (الف) تمام شرایط مستندسازی می‌شوند.
 (ب) راهنمایی برای تهیه برنامه مدیریت پروژه است.
 (ج) فقط در زمانی تهیه می‌شود که موضوع کار، یک سیستم خبره می‌باشد.
 (د) وقتی که مجموعه پیچیده از شرایط و اعمال در مؤلفه‌ها ظاهر می‌شود.

۱۲- اغلب زبان طراحی برنامه (PDL) عبارت است از

- الف) ترکیبی از ساختارهای برنامه نویسی و توصیف متنی
- ب) زبان برنامه نویسی مجاز در قالب درست خود
- ج) زبان توسعه نرم افزاری که قابلیت خواندن توسط ماشین را دارد.
- د) راه مفیدی برای نمایش معماری نرم افزار است.

۱۳- چون یک زبان طراحی برنامه، زبان برنامه نویسی واقعی نیست، بنابراین طراحان آزاد هستند طراحی رویه‌ای خود را بدون نگرانی از خطاهای معنایی (syntax) بنویسند.

- الف) درست
 - ب) نادرست
- ۱۴- مهندسین نرم افزار مدرن معتقدند که مفیدترین نماد طراحی برای نمایش رویه فقط تولید شبه کد است.
- الف) درست
 - ب) نادرست

۱۵- کدامیک از معیارهای زیر برای ارزیابی موثر بودن یک نماد طراحی رویه خاص مفید است.

- الف) قابلیت نگهداری
- ب) قابلیت پیمانه‌بندی
- ج) سادگی
- د) همه موارد فوق