

فصل ۱۶ طراحی (سطح اجزاء)  
 این کتاب تنها به خاطر حل مشکل دانشجویان پیام نور تبدیل به پی دی اف شد. همین جا از ناشر و نویسنده و تمام کسانی که با افزایش قیمت کتاب ما را مجبور به این کار کردند و یا متحمل ضرر شدند عذرخواهی می‌کنم.  
 گروهی از دانشجویان مهندسی کامپیوتر مرکز تهران

## فصل ۱۶ طراحی تفصیلی (سطح اجزاء)

### مفاهیم کلیدی (مرتب بر حروف الفبا)

برنامه سازی ساختیافته، ترتیب، جدول تصمیم گیری، زبان طراحی برنامه، ساختمان تکرار، ساختمان های ساختیافته، ساختمان شرط، علائم طراحی، علائم گرافیکی، نمودار جعبه ای

### KEY CONCEPTS

box diagram, condition construct, decision table, design notation, Graphical notation, program design language, repetition construct, sequence, structured constructs, structured programming

### نگاه اجمالی

طراحی اجزاء چیست؟ طراحی داده‌ها، معماری و رابط باید به نرم‌افزار عملیاتی تبدیل و ترجمه شود. در انجام این کار، طراحی باید در سطحی از انتزاع که نزدیک به برنامه می‌باشد، ارائه شود. طراحی اجزاء جزئیات الگوریتمی لازم برای دستکاری ساختارهای داده را ایجاد کرده، بر ارتباط بین اجزاء نرم‌افزاری از طریق رابط‌های آنها، تأثیر می‌گذارد و الگوریتم‌های پردازشی مختص به هر جزء را اجرا می‌کند.

چه کسی عهده دار این امر می‌باشد؟ طراحی در سطح اجزاء برعهده مهندس نرم‌افزار می‌باشد.

طراحی از چه جهت اهمیت دارد؟ قبل از ساخت برنامه، باید بتوانید کارکرد یا عدم کارکرد آن را تعیین کنید. طراحی اجزاء نوعی ارائه نرم‌افزاری است که به شما امکان می‌دهد تا جزئیات طراحی را به‌ظهور صحت هماهنگی و سازگاری با نمونه‌های اولیه طراحی (یعنی طراحی داده‌ها، معماری و رابط) بررسی کنید و نیز شیوه‌ای جهت ارزیابی کارایی ساختارهای داده‌ای، رابط‌ها و الگوریتم‌ها را فراهم می‌نماید.

مراحل طراحی کدامند؟ بازنمایی طراحی داده‌ها، معماری و رابط‌ها، اساس طراحی اجزاء را تشکیل

می‌دهند. با به‌کارگیری مجموعه‌ای از سازه‌های برنامه‌سازی ساختیافته، شرح پردازشی هر یک از اجزاء به یک مدل طراحی رویه‌ای تبدیل می‌شود. و علائم گرافیکی، جدولی یا متنی برای نمایش طراحی به‌کار می‌روند.

محصول کاری چیست؟ طراحی رویه‌ای هر یک از اجزاء که با علایم گرافیکی، جدولی یا متنی نشان داده می‌شود، محصول عمده کاری است که طی طراحی اجزاء تولید می‌شود.

چگونه می‌توان درستی انجام طراحی را تضمین کرد؟ بررسی دقیق بازرسی طراحی انجام می‌گیرد. طراحی مورد بررسی قرار می‌گیرد تا معلوم شود که آیا ساختارهای داده‌ای، رابط‌ها، توالی‌های پردازشی و شرایط منطقی، صحیح هستند و داده‌های مناسب را ایجاد خواهند کرد و تبدیل و تغییر اختصاص یافته به هر جزء را طی مراحل اولیه طراحی، کنترل می‌نماید.

طراحی اجزاء که «طراحی رویه‌ای»<sup>۱</sup> نیز نام دارد، پس از تعیین طراحی داده‌ها، معماری و رابط انجام می‌گیرد و هدف تبدیل مدل طراحی به نرم‌افزار عملیاتی می‌باشد. اما سطح انتزاعی مدل طراحی موجود، نسبتاً زیاد بوده و میزان انتزاع برنامه‌عملیاتی کم است. تبدیل و جزگردان، سخت و دشوار بوده و موجب ایجاد خطاهای ظریفی می‌شود که یافتن و تصحیح آنها در مراحل بعدی فرآیند افزاری دشوار است. «ادگار دایکسترا» [DIJ72]<sup>۲</sup> که نظرات مؤثری در درک طراحی ارائه داد، در سخنرانی معروفی اظهار داشت:

به نظر می‌رسد نرم‌افزار با بسیاری از محصولات دیگر متفاوت باشد، چنانچه به عنوان یک اصل، کیفیت بالاتر مستلزم قیمت بیشتر است. آنهایی که خواستار نرم‌افزاری واقعاً مطمئن هستند، درمی‌یابند که برای شروع، باید به دنبال شیوه‌ای برای اجتناب از اکثریت خطاها باشند و به این ترتیب روند برنامه‌سازی راحت‌تر خواهد شد ... برنامه‌سازی کارآمد نباید وقت خود را صرف اشکال‌زدایی کنند و کار آنها نباید توأم با خطاهای پنهان باشد.

گرچه این اظهارات مربوط به سال‌ها قبل است اما هنوز هم صحت دارند. هنگام تبدیل و برگردان مدل طراحی به برنامه منبع بایستی مجموعه‌ای از اصول طراحی را رعایت کرد که نه تنها تبدیل و ترجمه را انجام می‌دهند، بلکه از خطاهای آغازین نیز جلوگیری می‌کنند.

نمایش طراحی اجزاء 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. صرف نظر از مکاتیزم به کار رفته در نمایش طراحی اجزاء، ساختارهای داده‌ای، رابط‌های و الگوریتم‌های مشخص شده باید با مجموعه‌ای از رهنمودهای طراحی مطابقت داشته باشند تا ضمن پیشرفت و تکمیل طراحی رویه‌ای، مانع خطا گردند. در این فصل به بررسی رهنمودهای طراحی می‌پردازیم.

1. procedural design

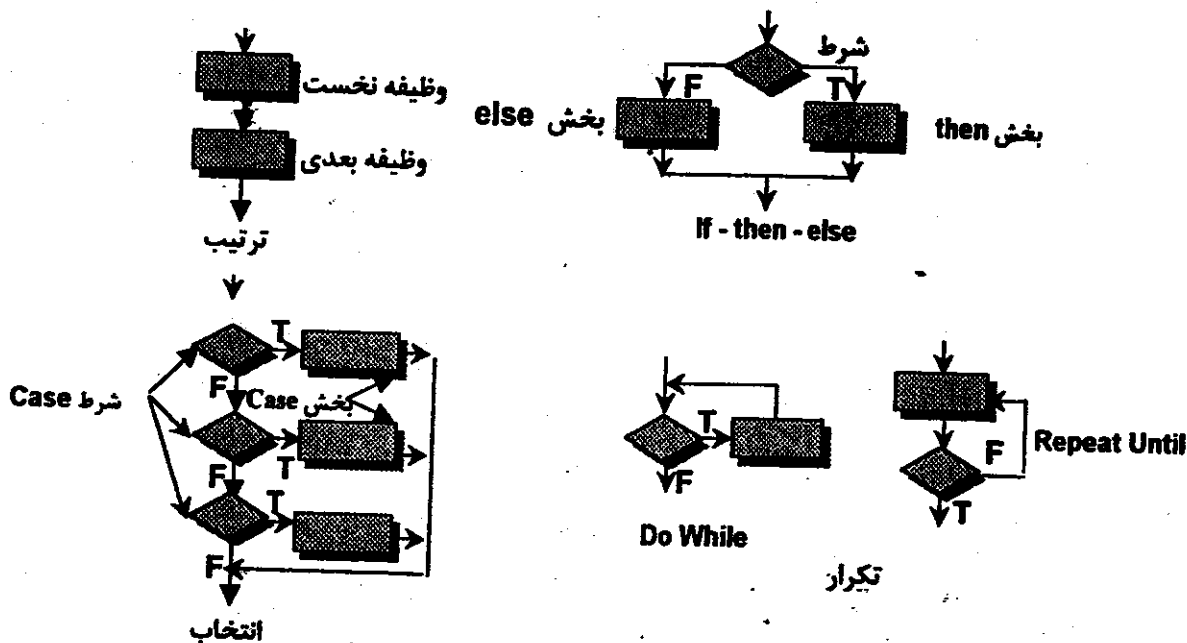
2. Dijkstra, E.

## ۱-۱۶ برنامه سازی ساختیافته

اصول طراحی اجزاء در اوایل دهه ۶۰ پایه‌ریزی شد و با کار «ادگار دایکسترا» و همکارانش [BOH66]<sup>۱</sup> و [DIJ65]<sup>۲</sup> و [DIJ76]<sup>۳</sup> کامل گردید. در اواخر دهه ۱۹۶۰، «دیجک استرا» و سایرین کاربرد مجموعه‌ای از سازهای محدود منطقی را پیشنهاد دارند که از طریق آنها هرگونه برنامه‌ای قابل شکل‌گیری بود. این ساختارها عبارتند از توالی<sup>۴</sup>، وضعیت<sup>۵</sup> و تکرار<sup>۱</sup>. توالی، مراحل پردازشی ضروری در تعیین هر الگوریتم را اجرا می‌کند. وضعیت، امکان پردازش انتخابی بر اساس رخداد منطقی را فراهم می‌کند و تکرار، حلقه‌زنی را امکان‌پذیر می‌سازد. این سه سازه در برنامه‌سازی ساختیافته - که یک تکنیک مهم طراحی اجزاء می‌باشد، اساسی و ضروری به‌شمار می‌روند.

## نقل قول

ایده‌ها و یافته‌های  
فنی و تکنولوژیکی  
موتور محرکه و  
اقتصادی می‌باشد  
وال استریت ژورنال



شکل ۱۶-۱ ساختارهای روند نما

سازهای ساختیافته جهت محدود ساختن طراحی رویه‌ای نرم‌افزار بر تعداد محدودی از عملیات قابل پیش‌بینی، پیشنهاد شدند. متریک‌های پیچیدگی (فصل ۱۹) بیانگر آنند که سازهای ساختیافته، پیچیدگی برنامه را کاهش داده و بدین‌طریق، خوانایی، آزمون و نگهداری را تقویت می‌کند. استفاده از تعداد محدود



نرم‌افزار هم معمول  
است و هم وسیله‌ای  
برای انتقال و حمل  
معمول

1. Bohm, C. and G.
2. Dijkstra, E.
3. Dijkstra, E.
4. Sequence
5. Condition

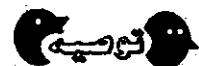
سازه‌های منطقی، همچنین در روند درک انسانی که روانشناسان آن را «چانکینگ»<sup>۲</sup> می‌نامند، نیز مؤثر است. به‌منظور درک این فرآیند، نحوه خواندن این صفحه را در نظر بگیرید. شما حروف جداگانه را نمی‌خوانید بلکه مجموعه حروف که واژه‌ها یا عبارات را تشکیل می‌دهند، مدنظر قرار می‌دهید.

سازه‌های ساختیافته، قطعات منطقی هستند که به خواننده امکان می‌دهند تا عناصر رویه‌ای یک پیمانه را شناسایی کند، نه این‌که طراحی یا برنامه را خط به خط بخواند. در **فهم به‌هنگام** با طرح‌های منطقی که به‌راحتی قابل تشخیص اند، تقویت می‌گردد.

### ۱-۱۶-۱ علائم طراحی گرافیکی

«یک تصویر معادل هزار واژه ارزش دارد»<sup>۳</sup> اما شناخت نوع تصویر و هزار کلمه، اهمیت دارد. هیچ تردیدی نیست که ابزارهای گرافیکی مانند فلوجارت یا نمودارهای کادری، 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.

فلوجارت<sup>۴</sup> از لحاظ تصویری کاملاً ساده است. برای نشان دادن یک مرحله پردازشی از کادر استفاده می‌شود. نماد توری شکل شرط منطقی را نمایش می‌دهد و پیکان‌ها نمایانگر جریان کنترل هستند. تصویر ۱-۱۶ سه سازه ساخت یافته را نشان می‌دهد. توالی به‌صورت دو کادر پردازشی نمایش داده می‌شود که با یک خط (پیکان) کنترل به‌هم متصلند. حالت<sup>۵</sup> که «دستور شرطی»<sup>۶</sup> نیز نام دارد به‌صورت یک نماد تصمیم‌گیری مصور می‌شود که اگر درست باشد موجب پردازش بخش **Then**<sup>۷</sup> می‌گردد و اگر غلط باشد، باعث فراخوانی پردازش بخش **Else**<sup>۸</sup> می‌گردد. «تکرار»<sup>۹</sup> با به‌کارگیری دو فرم کمی متفاوت نشان داده می‌شود: **do - While**<sup>۱۰</sup> شرط را آزمون می‌کند و تا مادامی‌که شرط صدق کند، حلقه‌زنی را به‌طور مکرر اجرا می‌نماید. **Repeat - Until**<sup>۱۱</sup> ابتدا حلقه‌زنی را اجرا می‌کند، سپس شرط را آزمون کرده و تکرار حلقه را تا فقدان شرط ادامه می‌دهد. ساختار گزینشی **Case** (یا مورد انتخاب) که در شکل نشان داده شده، در



ساختمان برنامه سازی  
ساختیافته، طراحی را  
قابل فهم تر می  
سازد اگر در استفاده از  
آنها پیچیدگی های  
غیر ضروری نباشد  
را بکار نگیرید نتیجه  
سالم خواهد بود.

1. structured programming

2. chunking

۳. گویا یک ضرب‌المثل چینی است. مترجم.

۴. در بسیاری متون فلوجارت را به روندنما برگردانده‌اند.

5. Condition

6. if-then-else

7. then-part

8. else-part

9. Repetition

10. do while

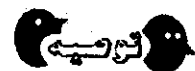
11. repeat until

واقع بسط و گسترش If - Then - Else می‌باشد. آزمون یک پارامتر به‌واسطه تصمیم‌های متوالی تا وقوع شرط منطقی و اجرای مسیر پردازشی بخش Case، انجام می‌شود.

سازه‌های ساخت یافته همان‌طور که در شکل ۱۶-۲ نشان داده شده، ممکن است داخل یکدیگر قرار گیرند. با توجه به شکل، repeat - until، بخش Then یک If - Then - else را شکل می‌دهد. (که با مرز خط تیره بیرونی محصور شده است). If - Then - else دیگر، بخش else شرط بزرگتر را تشکیل می‌دهد. در نهایت خود شرط، بلوک دوم در یک توالی می‌شود. به‌واسطه این شیوه تداخل سازه‌ها، یک طرح پیچیده منطقی ممکن است ایجاد شود. لازم به‌ذکر است که هر یک از بلوک‌های شکل ۱۶-۲ می‌تواند معرف پیمانه ای دیگر باشد و بدین‌ترتیب لایمندی رویه‌ای که لازمه ساختار برنامه است، انجام می‌گیرد.

بمطوری کلی، زمانی که فرار از مجموعه حلقه‌های تو در تو یا شرایط متداخل ضرورت دارد، کاربرد جزمی صرف ساختارهای ساخت یافته می‌تواند موجب بازدهی کم گردد. و مهم‌تر از آن، پیچیدگی بیشتر همه آزمون‌های منطقی در امتداد مسیر قرار می‌تواند موجب دشواری جریان کنترل نرم‌افزار شده، احتمال خطا را افزایش دهد و تأثیری منفی بر قابلیت خواندن و نگهداری داشته باشد. ما چه کار می‌توانیم بکنیم؟

دو امکان در اختیار طراح است. (۱) نمایش رویه‌ای مجدداً طراحی می‌شود به‌طوری‌که "شاخه فرار" در یک محل تو در تو جریان کنترل، ضرورتی ندارد. (۲) سازه‌های ساخت یافته به شیوه‌ای کنترل شده نقض می‌گردند؛ یعنی انشعاب محدودی خارج از جریان تو در تو طراحی می‌شود. ظاهراً انشعاب اول، روش ایده‌آل است اما امکان دوم بدون نقض شیوه برنامه‌سازی ساخت یافته، قابل انطباق است. نمودار کادری که یکی دیگر از ابزارهای طراحی گرافیکی است ناشی از تمایل و گرایش در جهت ایجاد یک نمایش طراحی رویه‌ای است که موجب تخطی از سازه‌های ساخت یافته نمی‌شود. این گونه نمودارها که توسط "ناسی" و "اشنیدرمن" ایجاد شده و توسط "چاپین" توسعه یافتند و (به چارتهای ناسی- اشنیدرمن<sup>۱</sup> یا ان - اس چارت<sup>۲</sup> یا چاپین چارت<sup>۳</sup> نیز معروفند)، دارای مشخصات زیر هستند: [NAS73] و [CHA74] (۱) قلمرو کاری (یعنی حوزه تکرار یا if - Then - else) کاملاً مشخص و به‌عنوان یک نمایش تصویری به وضوح قابل رؤیت است. (۲) انتقال اختیاری کنترل غیرممکن است. (۳) حوزه داده‌های محلی و/یا سراسری به‌راحتی قابل تعیین است. (۴) نمایش بازگشت آسان است.

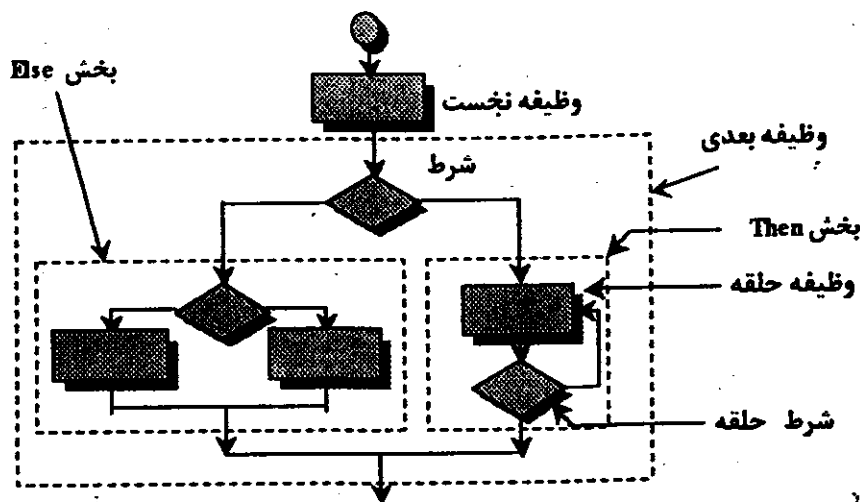


اینک روندنما و نمودارهای جمعی ای کاربرد گسترده پیشین را از دست داده اند. به طور کلی برای مستند سازی یا ارزیابی طراحی در مصادیق مشخص آنها را استفاده کنید، نه برای نمایش کل یک سیستم.

1. Nassi-Shneiderman charts

2. N-S charts

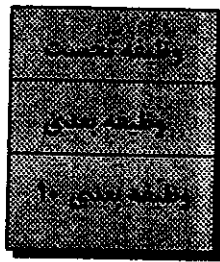
3. Chapin charts



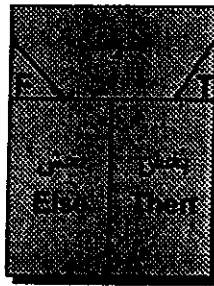
شکل ۱۶-۲ ساختمانهای تودرتو

نمایش گرافیکی سازه‌های ساخت یافته با استفاده از نمودار کادری (یا جعبه‌ای) در شکل ۱۶-۲ نشان داده شده است. عنصر اصلی نمودار یک کادر یا جعبه است. برای نمایش توالی، دو کادر از پایین به بالا به یکدیگر متصل می‌شوند. جهت نمایش if-Then-else، کادر if-Then-Part، Then-Part، else-Part به دنبال Condition قرار می‌گیرد. تکرار با یک الگوی محدودکننده که فرایند قابل تکرار را محصور می‌کند. (Do-While-Part یا Repeat-Until-Part) به نمایش درمی‌آید. در نهایت، انتخاب با به کارگیری فرم گرافیکی نشان داده شده در پایین تصویر، به نمایش درمی‌آید.

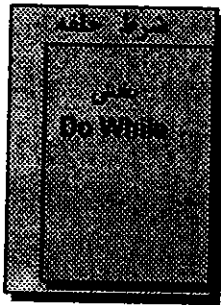
نمودار کادری نیز مانند فلوجارت‌ها به هنگام پالایش عناصر پردازشی یک پیمانه، در چندین صفحه لایه‌بندی می‌شود. یک فراخوانی پیمانه توسط کادری در داخل پیمانه که نام آن درون یک بیضی محصور شده، قابل نمایش است.



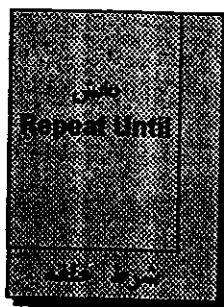
ترتیب



if - then - else



تکرار



انتخاب

شکل ۱۶-۳ ساختمانهای نمودار جعبه ای

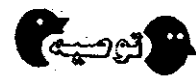
## ۲-۱-۱۶ علائم طراحی جدولی

در بسیاری از کاربردهای نرم‌افزاری، جهت ارزیابی ترکیب پیچیده شرایط و انتخاب اعمال مناسب مبتنی بر این حالت‌ها، ممکن است یک پیمانه ضرورت یابد. جداول تصمیم‌گیری، علایمی را به کار می‌برد که اعمال و شرایط (توصیف شده در شرح پردازشی را به شکل جدولی تبدیل می‌کند. تعبیر غلط جدول، دشوار است و حتی ممکن است به عنوان ورودی قابل خواندن دستگاه به یک الگوریتم جدولی مورد استفاده قرار گیرد.

ند چاپین<sup>۱</sup> در توضیح مفصل این ابزار طراحی چنین می‌گوید: [HUR83]

”برخی ابزارها و فنون قدیمی نرم‌افزاری به خوبی با ابزارها و فنون جدید مهندسی نرم‌افزار سازگاری دارند. جداول تصمیم‌گیری نمونه بسیار خوبی هستند. جداول تصمیم‌گیری حدوداً یک دهه مقدم بر مهندس نرم‌افزارند، اما به خوبی با مهندسی نرم‌افزار سازگار هستند، به طوری که ممکن است بدان منظور طراحی شوند.

سازمان‌دهی جدول تصمیم‌گیری در شکل ۱۶-۴ به نمایش درآمده است. مطابق شکل، این جدول به چهار بخش تقسیم می‌شود. 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 چهارم‌های سمت راست، ماتریسی را تشکیل می‌دهند که نشان‌دهنده ترکیب شرایط و اعمال و اقدامات



یک جدول تصمیم‌گیری را هنگام برخورد با مجموعه پیچیده‌ای از شرایط و اقدامات، در طی جزء به کار ببرد.



چگونه یک جدول تصمیم‌گیری را بسازیم؟

1. Hurley, R.B.

متناظر با یک ترکیب خاص هستند. هر ستون ماتریس ممکن است به عنوان یک قانون<sup>۱</sup> 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 به خصوصی که ترکیبات غیر ممکن شرایط را برطرف می سازند، مرتبط نمایید یا این که هر گونه جابه جایی ممکن در شرایط ایجاد کنید.
- ۴- با بیان اعمال انجام شده در ارتباط با مجموعه شرایط، قوانین را تعریف کنید.

#### قواعد

شرایط	۱	۲	۳	۴	n
شرط شماره ۱	?			?	
شرط شماره ۲		?		?	
شرط شماره ۳			?		
شرط شماره ۴					
شرط شماره ۵					
اقدامات					
اقدام شماره ۱	?			?	
اقدام شماره ۲		?		?	
اقدام شماره ۳			?		
اقدام شماره ۴		?	?	?	
اقدام شماره ۵	?	?		?	

شکل ۱۶-۴ ساختار جدول تصمیم گیری

جهت توضیح کاربرد جدول تصمیم گیری، قطعه زیر از شرح پردازشی برای یک سیستم صدور صورت حساب خدمات عمومی را در نظر بگیرید:

اگر حساب مشتری با استفاده از شیوه نرخ ثابت فاکتور شود، هزینه حداقل ماهانه برای مصرف کمتر از ۱۰۰ KWH (کیلو وات در ساعت) برآورد می گردد. در غیر این صورت، صدور فاکتور کلمپیوتری، ساختار نرخ *Schedule A* را به کار می برد. هر چند، اگر صورت حساب با به کارگیری روش نرخ متغیر فاکتور گردد، ساختار نرخ *Schedule A* (جدول زمانی) در مورد مصرف پایین ۱۰۰ KWH به کار خواهد رفت و مصرف اضافی طبق *Schedule B* (جدول زمانی) فاکتور می گردد.

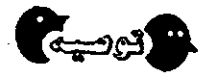


شکل ۱۶-۵ نمایش جدول تصمیم‌گیری شرح قبلی را نشان می‌دهد. هر یک از پنج قانون بیان‌گر آن است که یکی از پنج شرط (یعنی "T" یا True) در صورت حساب نرخ ثابت و متغیر، در متن این رویه مفهومی ندارد. بنابراین، این شرط حذف می‌شود. به‌عنوان یک قانون کلی، جدول تصمیم‌گیری را می‌توان به‌طور مؤثر جهت تکمیل سایر علائم طراحی رویه‌ای مورد استفاده قرار داد.

### ۱۶-۱-۲ زبان طراحی برنامه (PDL)

"زبان طراحی برنامه" (PDF) که انگلیسی ساختاریافته<sup>۱</sup> یا شبه کد<sup>۲</sup> نیز نام دارد، یک زبان آمیخته است به‌طوری‌که واژگان یک زبان (یعنی انگلیسی) و نحوه کلی زبانی دیگر (یعنی زبان برنامه‌سازی ساختاریافته) را به‌کار می‌برد. در این فصل PDL به‌عنوان یک ارجاع کلی برای یک زبان طراحی به‌کار می‌رود. [CAI75]

در نگاه اول، PDL مانند یک زبان برنامه‌نویسی مدرن به‌نظر می‌رسد. تفاوت بین PDL و یک زبان واقعی برنامه‌نویسی مربوط به تفاوت متن روایتی است (مثل انگلیسی) که مستقیماً داخل دستورات PDL جاسازی می‌شود. با فرض کاربرد ترکیبی متن روایتی مستقیماً جا گرفته در یک ساختار نحوی، PDL قابل کاپایل نیست. (حداقل فعلاً این‌گونه است.) هر چند ابزارهای PDL که در حال حاضر وجود دارند PDL را به زبان برنامه‌نویسی "Skeleton" و/یا نمایش گرافیکی (مثل فلوچارت) طراحی برمی‌گردانند. این ابزارها هم‌چنین نقشه‌های تداخل، نمایه شاخص عملکرد طراحی، جداول ارجاع متقابل و مجموعه‌ای از اطلاعات دیگر را به‌وجود می‌آورند.



اینکه از زبان برنامه سازی خودتان برای PDL (زبان طراحی برنامه) استفاده کنید ایده خوبی است. این امر شما را قادر می‌سازد همگام با طراحی اجزاء اسکلت برنامه را نیز (همراه با توضیحات) تولید نمایید.

1. Program Design Language

2. structured English

3. Pseudo Code

4. Caine, S. and K.

## قواعد

شرایط	۱	۲	۳	۴	۵
نرخ ثابت حساب	T	T	F	F	F
نرخ متغیر حساب	F	F	T	T	F
کیلو وات ساعت < ۱۰۰ مصرف	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 ۵-۱۶ جدول تصمیم گیری نتایج

زبان طراحی برنامه ممکن است نسخه ساده زبانی مثل Ada یا C بوده و یا آن که ممکن است محصولی باشد که به طور ویژه جهت طراحی رویه های خریداری می شود. صرف نظر از منشاء تهیه آن، 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.

نحو<sup>۱</sup> اصلی PDL باید در بردارنده سازه هایی برای تعریف زیر برنامه، توصیف رابط، معرفی داده ها، فنون ساخت بلوک، سازه های شرطی، سازه های تکرار و ساختارهای I/O باشد. قالب و معنانشناسی برخی از این سازه های PDL در قسمت بعدی ارائه می گردد.

قابل ذکر است که PDL با در برداشتن کلمات کلیدی برای چند وظیفه ای کردن و/یا پردازش همزمان، وقفه گردانی همروندی بین فرآیندی و بسیاری از خصوصیات دیگر، قابل توسعه و گسترش است.

طرح‌های کاربردی که PDL برای آنها مورد استفاده قرار می‌گیرد، بایستی فرم نهایی زبان طراحی را تعیین کند.

#### ۱۶-۱-۴ مثالی از زبان طراحی برنامه

جهت توضیح کاربرد PDL، نمونه‌ای از طراحی رویه‌ای برای نرم‌افزار سیستم امنیت خانه امن را که در فصل‌های پیشین معرفی شد، ارائه می‌دهیم. سیستم مورد نظر خانه امن اعلان خطر برای آتش، دود، سارق، آب و دما را کنترل می‌کند. (مثلاً وقتی صاحبخانه در زمستان حضور ندارد، لوله‌ای بترکد) زنگ خطر را ایجاد کرده و خدمات ناظر را با تولید صدا سازی شده، فرا می‌خواند. در نمونه ارائه شده PDL برخی سازه‌های مهم ذکر شده در قسمت‌های قبل را توضیح می‌دهیم.

به یاد داشته باشید که PDL یک زبان برنامه‌نویسی نیست. طراح می‌تواند بدون نگرانی از بابت خطاهای نحوی، در صورت لزوم انطباق را ایجاد کند. با این وجود، طراحی نرم‌افزار ناظر باید مورد مرور و تجدیدنظر قرار گیرد (آیا به‌منظر شما مشکلی وجود ندارد؟) و قبل از نوشتن برنامه باید اصلاح گردد. PDL زیر توضیح طراحی رویه‌ای را برای جزء ناظر امنیتی<sup>۱</sup> تعیین می‌کند:

```
PROCEDURE security.monitor;
INTERFACE RETURNS system.status;
TYPE signal IS STRUCTURE DEFINED
name IS STRING LENGTH VAR;
address IS 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;
```

```
initialize all system ports and reset all hardware;
CASE OF control.panel.switches (cps):
  WHEN cps = "test" SELECT
    CALL alarm PROCEDURE WITH "on" for test.time in seconds;
  WHEN cps = "alarm-off" SELECT
    CALL alarm PROCEDURE WITH "off";
  WHEN cps = "new.bound. temp" SELECT
    CALL keypad.input PROCEDURE;
  WHEN cps = "burglar. alarm.off" SELECT deactivate signal
    [burglar.alarm];

  DEFAULT none;
ENDCASE
  REPEAT UNTIL activate. switch is turned off
  reset all signal.values and switches;
  DO FOR alarm. type = smoke, fire, water, temp, burglar;
  READ address [alarm.type] signal. value;
  IF signal.value > bound [alarm.type]
  THEN phone. message = message [alarm.type];
  set alarm.bell to "on" for alarm. timesseconds;
  PARBEGIN
    CALL alarm PROCEDURE WITH "on", alarm. time in seconds;
    CALL phone PROCEDURE WITH message [alarm. type], phone.
number;
  ENDPAR
  ELSE skip
  ENDIF
ENDFOR
ENDREP
END security.monitor
```

توجه کنید که طراح جزء ناظر امنیتی<sup>۱</sup> یک سازنده جدید یعنی PARBEGIN ... ENDPAR را به کار برده است که مشخص کننده یک بلوک مولزی است. تمامی اعمال تعیین شده در داخل بلوک PARBECIN به طور مولزی اجراء می شوند. در این مورد، جزئیات اجزاء در نظر گرفته نمی شوند.

## ۱۶-۲ مقایسه علائم طراحی

در قسمت قبلی، برخی فنون متفاوت جهت نمایش طراحی رویه‌ای را ارائه نمودیم. مقایسه بایستی مبتنی این فرض باشد که هر گونه نشانه گذاری در طراحی اجزاء اگر به درستی مورد استفاده قرار گیرد می تواند در روند طراحی مؤثر واقع شود و برعکس حتی بهترین نشانه گذاری هم اگر به درستی به کار نرود، چندان در درک تأثیری ندارد. با فرض این مسئله، معیارهایی را که در مقایسه علائم طراحی به کار می روند، مورد بررسی قرار می دهیم.

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.

خصوصیات زیر در مورد نشان گذاری طراحی در زمینه مشخصات کلی فوق الذکر تعیین شده اند:

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، می تواند 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 بر هزینه ترین مرحله در دوره زندگی نرم افزاری است. نگهداری پیکربندی نرم افزار تقریباً همواره به معنای نگهداری از نمایش طراحی رویه‌ای می باشد.



کدام معیارهای اساسی  
برای ارزیابی علائم  
طراحی باید استفاده  
شوند؟

### مسایل و نکاتی برای تفکر و تعمق بیشتر

۱-۱۶ بخش کوچکی از یک برنامه موجود را انتخاب کنید (حدود ۵۰ تا ۷۵ خط). ساختمانهای ساختافته برنامه را با رسم چهارگوش‌هایی دور آنها مشخص کنید. آیا این قطعه از برنامه ساختارهایی دارد که با فلسفه برنامه‌سازی ساختافته بخواند؟ اگر اینچنین است، این کد را به گونه‌ای که از ساختمانهای ساختافته تبعیت کند، مورد طراحی مجدد قرار دهید. و اگر نه، چه چیزی چهارگوش‌های رسم شده مورد توجه شما خواهد بود؟

۲-۱۶ همه زبان‌های برنامه‌نویسی پیشرفته، ساختمان‌های برنامه‌سازی ساختافته را پیاده‌سازی می‌کنند. مثال‌هایی از سه زبان برنامه‌نویسی بیاورید.

۳-۱۶ چرا قطعه‌بندی طی فرآیند مرور طراحی سطح اجزاء اهمیت دارد؟

مسائل ۴-۱۶ تا ۱۱-۱۶ را می‌توان با کاربرد یک (یا چند) علائم رویه‌ای که در این فصل ارائه شدند، نمایش داد. استادان برای هر مسئله علائم طراحی خاصی را تعیین خواهند کرد.

۴-۱۶ یک طراحی رویه‌ای برای اجزاء مرتب‌سازی‌های زیر را، توسعه دهید: مرتب‌سازی شل مژتر، مرتب‌سازی هرتزی، مرتب‌سازی BSST (درختی). اگر با این مرتب‌سازی‌ها آشنایی ندارید به یک کتاب ساختمان داده‌ها رجوع کنید.

۵-۱۶ یک طراحی رویه‌ای برای رابط کاربر محاوره‌ای توسعه دهید که اطلاعات اولیه مالیات بر درآمد را مورد سوال قرار می‌دهد. نیازمندیها را خود تعیین نمایید و فرض کنید کلیه محاسبات مربوط به مالیات را دیگر پیمانه‌ها عهده دارند.

۶-۱۶ یک طراحی رویه‌ای برای برنامه‌ای توسعه دهید که متنی با طول دلخواه را به‌عنوان ورودی دریافت داشته، لیستی از کلمات و تعداد تکرار آنها را به‌عنوان خروجی ایجاد می‌کند.

۷-۱۶ یک طراحی رویه‌ای برای برنامه‌ای توسعه دهید که انگرال تابع  $f(x)$  در بازه  $a$  تا  $b$  به طور عددی محاسبه می‌کند.

۸-۱۶ یک طراحی رویه‌ای برای یک ماشین تورینگ عمومی توسعه دهید که مجموعه‌ای از چهار تایی‌ها را به‌عنوان ورودی گرفته خروجی را به‌صورت مشخص شده تولید می‌کند.

۹-۱۶ یک طراحی رویه‌ای برای برنامه‌ای توسعه دهید که مسئله برج‌های هانوی را حل کند. بسیاری از کتلهای هوش مصنوعی این مسئله را به تفصیل بحث کرده‌اند.

۱۰-۱۶ یک طراحی رویه‌ای برای تمام یا بخشهای اصلی یک تجزیه‌کننده LR برای یک کامپایلر توسعه دهید. به یک یا چند کتاب درخصوص طراحی کامپایلرها مراجعه نمایید.

۱۱-۱۶ یک طراحی رویه‌ای برای یک الگوریتم به رمز کردن / رمز گشایی که به دلخواه خود انتخاب نموده اید، توسعه دهید.

۱۶-۱۲ در یک یا دو صفحه استدلال کنید که دلیل بیاورید که نسبت به کدام علائم طراحی احساس بهتری دارید. اطمینان حاصل کنید که نظرات شما با معیاری هیا بخش ۱۶-۲ می خوانند.

آثار دیگر مترجم

نام کتاب	موضوع	تاریخ چاپ	ناشر
وقتی بریدی مهر	مجموعه آثار	بهار ۸۰	نشر تبلور
هدایت تغییر سوار بر موج دگرگونی	در حوزه مهندسی منابع و تحلیل سیستمها (کتاب برگزیده مهندسین و مدیران ارشد)	بهار ۸۱	مترجم
برنامه سازی همروند زبان برنامه سازی جاوا	(در بردارنده مباحث طراحی سیستمهای عامل)	در دست چاپ	انتشارات آکسفورد



# آثار منتشر شده انتشارات گسترش علوم پایه

## ریاضی و آمار

کد	نام کتاب	مؤلف	قیمت
۱۰	۱- حسابان (۱) ۲- حسابان (۲) ۳- کاربرد ریاضیات در مدیریت و اقتصاد و حسابداری (مجموعه گزینه‌های چهارجوابی کارشناسی ارشد)	تألیف دکتر مسعود نیکوکار تألیف دکتر مسعود نیکوکار تألیف دکتر مسعود نیکوکار	چاپ اول ۱۱۰۰۰ ریال چاپ اول ۹۰۰۰ ریال چاپ اول ۱۷۰۰۰ ریال
۱۱	۴- کاربرد آمار در مدیریت و اقتصاد و حسابداری	تألیف دکتر مسعود نیکوکار	چاپ اول ۱۴۰۰۰ ریال
۱۲۳	۵- ریاضیات و کاربرد آن در مدیریت (۲)	تألیف دکتر مسعود نیکوکار	چاپ هفتم ۲۵۰۰۰ ریال
۱۱۵	۶- راهنمای حل ریاضیات و کاربرد آن در مدیریت (۲)	تألیف دکتر مسعود نیکوکار	چاپ سوم ۱۷۵۰۰ ریال
۱۲۲	۷- ریاضیات و کاربرد آن در مدیریت (۱)	تألیف دکتر مسعود نیکوکار	چاپ هشتم ۲۵۰۰۰ ریال
۲	۸- راهنمای حل ریاضیات و کاربرد آن در مدیریت (۱)	تألیف دکتر مسعود نیکوکار	چاپ پنجم ۱۸۰۰۰ ریال
۹	۹- تحقیق در عملیات (۱)	تألیف دکتر مسعود نیکوکار	چاپ اول ۱۲۰۰۰ ریال
۱۰۳	۱۰- آمادگی کنکور تحقیق در عملیات (رشته ریاضی)	تألیف نیکوکار-درویش-ساروی	چاپ اول ۱۵۰۰۰ ریال
۵۵	۱۱- پژوهش عملیاتی (مدیریت، حسابداری، اقتصاد، بازرگانی)	تألیف نیکوکار-درویش-ساروی	چاپ اول ۲۰۰۰۰ ریال
۱۰۹	۱۲- آمادگی کنکور (رشته صنایع) پژوهش عملیاتی	تألیف نیکوکار-درویش	چاپ اول ۱۵۰۰۰ ریال
۱۳۵	۱۳- آمار و کاربرد آن در مدیریت (۱)	تألیف دکتر مسعود نیکوکار	چاپ ششم ۲۶۰۰۰ ریال
۶	۱۴- راهنمای حل آمار و کاربرد آن در مدیریت (۱)	تألیف دکتر مسعود نیکوکار	چاپ چهارم ۱۵۰۰۰ ریال
۱۱۲	۱۵- آمار و کاربرد آن در مدیریت (۲)	تألیف دکتر مسعود نیکوکار	چاپ ششم ۲۸۰۰۰ ریال
۱۱۷	۱۶- راهنمای حل آمار و کاربرد آن در مدیریت (۲)	تألیف دکتر مسعود نیکوکار	چاپ دوم ۱۶۰۰۰ ریال
۱۳	۱۷- ریاضیات گسسته	تألیف دکتر مسعود نیکوکار	چاپ اول ۲۵۰۰۰ ریال
۱۰۷	۱۸- ریاضیات پیش دانشگاهی (مدیریت، حسابداری، اقتصاد و بازرگانی)	تألیف دکتر مسعود نیکوکار	چاپ هفتم ۲۵۰۰۰ ریال
۶۷	۱۹- راهنمای حل ریاضیات پیش دانشگاهی (مدیریت، حسابداری...)	تألیف دکتر مسعود نیکوکار	چاپ دوم ۱۹۰۰۰ ریال
۱۴	۲۰- ریاضیات پیش دانشگاهی (فنی مهندسی)	تألیف دکتر مسعود نیکوکار	چاپ ششم ۲۶۰۰۰ ریال
۱۰۰	۲۱- راهنمای حل ریاضیات پیش دانشگاهی (فنی مهندسی-علوم)	تألیف دکتر مسعود نیکوکار	چاپ دوم ۲۴۰۰۰ ریال
۱۵	۲۲- تست جبر و جبر خطی (کارشناسی ارشد)	تألیف دکتر عرفانیان	چاپ دوم ۱۰۰۰۰ ریال
۱۷	۲۳- سوالات کارشناسی ارشد رشته ریاضی	تألیف دکتر مسعود نیکوکار	چاپ دوم ۲۴۰۰۰ ریال
۲۱	۲۴- آمار و احتمال (۱)	تألیف دکتر مسعود نیکوکار	چاپ دوم ۲۲۰۰۰ ریال
۲۲	۲۵- آمار و احتمال (۲)	تألیف دکتر مسعود نیکوکار	چاپ اول ۲۳۰۰۰ ریال
۲۳	۲۶- جبر و جبر خطی بنیادی	ترجمه دکتر رحمتی-اسلام زاده	چاپ اول ۱۵۰۰۰ ریال
۲۴	۲۷- معادلات انتگرال	ترجمه دکتر مهدی دهقان	چاپ اول ۱۳۰۰۰ ریال
۲۲	۲۸- مجموعه سؤالات ریاضی (ارشد فنی مهندسی ۷۹-۷۸)	تألیف دکتر مسعود نیکوکار	چاپ اول ۱۷۰۰۰ ریال
۳۵	۲۹- مجموعه سؤالات ریاضی (ارشد فنی مهندسی دانشگاه آزاد)	تألیف دکتر مسعود نیکوکار	چاپ اول ۱۲۰۰۰ ریال
۳۶	۳۰- جبر (۱)	تألیف نیکوکار-خاقری	چاپ اول ۲۵۰۰۰ ریال
۶۳	۳۱- جبر (۲)	تألیف نیکوکار-خاقری	چاپ اول ۲۳۰۰۰ ریال
۳۷	۳۲- راهنمای حل مسائل جبر (۱)	تألیف نیکوکار-خاقری	چاپ اول ۲۴۰۰۰ ریال
۱۰۲	۳۳- محاسبات عددی	تألیف نیکوکار-درویش	چاپ چهارم ۱۸۰۰۰ ریال
۴۰	۳۴- راهنمای حل مسائل آمار ریاضی ۱-۷ (جلد اول)	تألیف نیکوکار-سلکی-غازانی	چاپ اول ۲۶۰۰۰ ریال
۸۲	۳۵- راهنمای حل مسائل آمار ریاضی ۸-۱۲ (جلد دوم)	تألیف نیکوکار-سلکی-غازانی	چاپ اول ۲۶۰۰۰ ریال
۱۲۶	۳۶- راهنمای حل مسائل آمار ریاضی ۱۴-۱۶ (جلد سوم)	تألیف نیکوکار-سلکی-غازانی	چاپ اول ۱۵۰۰۰ ریال
۴۳	۳۷- زبان تخصصی رشته ریاضی	تألیف دکتر مسعود نیکوکار	چاپ اول ۱۱۰۰۰ ریال
۴۵	۳۸- گزیده مسائل جبر خطی	تألیف دکتر نیکوکار-سومنی	چاپ اول ۱۶۰۰۰ ریال
۴۴	۳۹- ۶۰۰ مسئله آنالیز (سرژانگ)	ترجمه نیکوکار-ساروی	چاپ اول ۲۰۰۰۰ ریال
۴۷	۴۰- وارون ماتریسهای غیرمربعی	تألیف نیکوکار-درویش	چاپ اول ۱۵۰۰۰ ریال
۴۶	۴۱- ریاضیات گسسته پیش دانشگاهی	تألیف نیکوکار-فردوسیان	چاپ اول ۱۷۵۰۰ ریال
۵۶	۴۲- آمار صنایع و مدیریت سیستم بهره‌وری (مجموعه گزینه‌های چهارجوابی کارشناسی ارشد)	تألیف دکتر مسعود نیکوکار	چاپ اول ۲۰۰۰۰ ریال

۱۶۰۰۰ ریال	چاپ اول	ترجمه دکتر علیرضا پیشوایی	۱۷۵	۱۲۸- مسائل سینتیک شیمیایی
۶۰۰۰ ریال	چاپ اول	تألیف محمدزمانی	۸۱	۱۲۹- آشنائی با شوره‌های اسلامی
۱۲۰۰۰ ریال	چاپ اول	تألیف بهمن‌گره‌بان-شیرا مولرینا	۶۴	۱۳۰- فرهنگ ضرب‌المثل‌های معاصر
۶۰۰۰ ریال	چاپ اول	تألیف مرجان حاج عطیسی	۱۱۱	۱۳۱- داستانهای واقعی برای زبان‌آموزان مقدماتی انگلیسی
۸۵۰۰ ریال	چاپ اول	تألیف مقدسیان-سزمانی	۱۳۲	۱۳۲- سیری در قانون تجارت
۹۵۰۰ ریال	چاپ اول	تألیف مهندس مقسمی	۱۳۳	۱۳۳- سایه عشق

این کتاب تنها به خاطر حل مشکل دانشجویان پیام نور تبدیل به پی دی اف شد. همین جا از ناشر و نویسنده و تمام کسانی که با افزایش قیمت کتاب ما را مجبور به این کار کردند و یا متحمل ضرر شدند عذرخواهی می‌کنم.  
گروهی از دانشجویان مهندسی کامپیوتر مرکز تهران