

# بخشی از ترجمه مقاله

عنوان فارسی مقاله :

ابزاری برای پشتیبانی از روش طراحی CRC (دسته، مسئولیت و همکاری)

عنوان انگلیسی مقاله :

## A Tool to Support the CRC Design Method

Steve Roach, Department of Computer Science, University of Texas at El Paso, sroach@cs.utep.edu  
Javier C Vásquez, Department of Computer Science, University of Texas at El Paso,

jvasquez@cs.utep.edu

توجه !



این فایل تنها قسمتی از ترجمه میباشد.

برای تهیه مقاله ترجمه شده کامل با فرمت ورد (قابل ویرایش) همراه با نسخه انگلیسی

مقاله، [اینجا](#) کلیک نمایید.

# بخشی از ترجمه مقاله

## Responsibilities and Collaborations

Objects exhibit behavior. A *responsibility* is an operation that an object can perform or knowledge that the object must maintain. For example, a *Cell* object in a business application might be responsible for displaying its value and knowing its location. A *SpreadSheet* object might be responsible for keeping track of a set of *Cell* objects and asking them to display themselves. An object can be viewed as a provider of services. Every object in a class can provide the same set of services. Therefore, responsibilities are associated with classes.

When an object requires the services of some other object in order to achieve a responsibility, a *collaboration* is formed. In the previous example, the class *SpreadSheet* collaborates with the class *Cell* in order to fulfill *SpreadSheet*'s responsibility to update the display. By separating the responsibilities for displaying the contents of a cell from the responsibility for displaying the entire spreadsheet, the design is more cohesive. It is possible to create different types of *Cells* (e.g., for text, numbers, and formulas), each with its own formatting rules, without the need to modify the *SpreadSheet* class. One fundamental goal of the design process is to identify the objects and the collaborations between them. When a responsibility is only used within an object, the responsibility is a *private responsibility*. Responsibilities that are not private form the basis for class interfaces, and in a language such as Java, these interfaces are implemented using public methods.

## مسئولیت ها و همکاری ها

اجسام رفتارها را نمایش می دهند. یک مسئولیت عملکردی است که یک جسم می تواند انجام دهد یا دانشی که جسم باید داشته باشد. برای مثال، یک جسم سلولی در یک کاربرد تجاری ممکن است مسئول نمایش ارزش آن و نشانگر موقعیت آن باشد. یک جسم اسپریشیت ممکن است مسئول ردگیری یک سری از اجسام سلول باشد و از آنها بخواهد تا خود را نشان دهند. یک جسم را می تواند بعنوان مهیا کننده خدمات در نظر گرفت. هر جسم در یک کلاس میتواند همان سری خدمات را مهیا سازد. بنابراین، مسئولیت ها همراه با کلاس ها هستند.

وقتی که یک جسم نیازمند خدمات چند جسم دیگر برای اجرای یک مسئولیت باشد، یک همکاری شکل می گیرد. در مثال قبلی، کلاس اسپریشیت با کلاس سلول برای اجرای مسئولیت اسپریشیت برای آپدیت نمایش، همکاری دارد. با جداسازی مسئولیت ها برای نمایش محتویات یک سلول از مسئولیت برای نمایش کل اسپریشیت، طراحی پیوسته تر می گردد. این امکان وجود دارد که انواع مختلفی از سلول ها را ایجاد کرد. یک هدف اصلی فرآیند طراحی اینست که اجسام و همکاری بین آنها را تشخیص داد. وقتی که یک مسئولیت فقط در یک جسم استفاده می شود، مسئولیت یک مسئولیت خصوصی می باشد. مسئولیت هایی که خصوصی نیستند، اساس میانگیرهای کلاس را تشکیل می دهند و در یک زبان مانند جاوا، این میانگیرها با استفاده از روش های عمومی اجرا می شوند.

## توجه !

این فایل تنها قسمتی از ترجمه میباشد.



برای تهیه مقاله ترجمه شده کامل با فرمت ورد (قابل ویرایش) همراه با نسخه انگلیسی مقاله، **اینجا** کلیک نمایید.

همچنین برای مشاهده سایر مقالات این رشته **اینجا** کلیک نمایید.