Slide 10

Ocultação de informação - É um aspeto importante relacionado com a abstração, que consiste em reduzir a informação e conteúdo a apenas o que é relevante. Esta ocultação pode ser implementada na forma de módulos, em que a informação de cada módulo é ocultada do resto do sistema e o acesso controlado. Isto permite manter um maior nível de abstração no software, sendo que alterações a um módulo não devem afetar o resto, o que permite que o código seja mais fácil de alterar.

Independência de funcionalidades - Baseia-se na independência entre módulos, ou seja, um módulo focado num objetivo deve conseguir atingi-lo com a menor interação possível com outros módulos. Um exemplo pode ser quando se realiza um trabalho de grupo, as tarefas são dividas de modo que todos os membros consigam realizar as suas tarefas sem estar dependentes nas tarefas a ser implementadas pelos restantes membros do grupo.

Refinamento ao longo dos passos - Princípio de que parte da ideia de desenvolver software por camadas de abstração, começando em níveis altos e incrementalmente, aperfeiçoando o mesmo em cada nível de abstração, com mais detalhe a cada nível. Em níveis altos o software é meramente o modelo de design, enquanto em níveis mais baixos tem-se cada vez mais código.

Refactoring - Consiste em modificar o código para melhor seguir a estrutura interna do código ou efetuar otimizações sem alterar as funcionalidades implementadas. Isto pode ser utilizado para melhorar a facilidade de compreensão do código.

Classes de design - fornecem detalhes de design que permitirão que as classes de análise sejam implementadas.

Slide 11

Neste slide podemos observar um exemplo de um design de classes.

Neste exemplo temos a "classe principal" Floorplan, com os seus métodos e atributos. Esta classe floorplan possui uma associação de 1 para muitos com a classe Camera, ou seja, assumimos que uma instância de FloorPlan pode ter várias instâncias de Camera.

Neste exemplo a classe Camera possui diversos atributos, mas não possui métodos.

A classe FloorPlan está tanbém relacionada com a classe Segment (1 para muitos). Esta classe Segment, possui os seus atributos e métodos, e pode ser de vários tipos (generalização?), como WallSegment ou Window.

Slide 12

Características do Design de Classes

Completa- Inclui todos os atributos e métodos que precisa. Não incluindo nada que não seja usado, e se houver algo que não esteja a ser utilizado, ou que não haja a necessidade de usar, é retirado.

Primitividade- cada método da classe fornece um serviço, existindo uma ligação entre classe e serviço.

Alta coesão– Classes pequenas, focadas num objetivo específico.

Baixo acoplamento– Mínima colaboração entre classes, ou seja, existe pouca interdependência entre as classes.

Realizado por:

Alexandre Gonçalves

Duarte Conceição

Inês Martins

Marco Teodoro