

TDE 1 – Resumo

Lucas Azevedo Dias

1. Encapsulamento:
É a simplificação de vários conceitos em uma palavra, assim, abstraindo diversos problemas atrás dessa única palavra. Em Orientação a Objetos, seria esconder toda a complexidade de uma operação atrás de um atributo na interface disponibilizada.
2. Ocultação de informações e implementações:
É a ocultação de informações e implementações que o encapsulamento traz a partir da supressão desses detalhes privados que ficam invisíveis ao lado exterior por não comporem a interface. Dessa forma, a maneira como esses detalhes são tratados internamente acaba sendo de pouco interesse para quem acessará sua interface, contanto que seja feito o que o atributo da interface propõe a fazer.
3. Retenção de estado:
Objetos, mesmo após sua execução, retêm suas informações e aguardam uma nova chamada.
4. Identidade de objeto:
Cada objeto possui uma identidade própria e é distinto de todos os outros, independentemente da sua classe ou do seu estado. Assim, cada objeto possui um identificador único atribuído a ele.
5. Mensagens:
É a forma de comunicação entre objetos, o qual um objeto remetente indica a um objeto destinatário para executar um de seus métodos. Sendo que há três tipos de mensagens: informativas (as quais fornecem informações novas para o objeto se atualizar), interrogativas (as quais requerem uma informação do objeto) e imperativas (as quais indicam uma ordem para o objeto cumprir).
6. Classes:
São a base genérica a qual define o padrão de onde serão criados os objetos, que, por sua vez, serão classificados como instâncias dessas classes. Além disso, na prática, as classes ajudam a economizar memória de alocação por generalizarem os objetos.
7. Herança:
É uma forma de uma classe (chamada subclasse) se basear em outra (a superclasse) para sua definição, assim copiando seus atributos e suas operações. Dessa forma, a herança facilita o trabalho do programador e a manutenção de código.
8. Polimorfismo:
Ocorre quando um nome de atributo define implementações diferentes em classes distintas. Ele fica associado às técnicas de *overriding* (redefinição), onde uma operação definida em uma superclasse é redefinida em uma subclasse, assim cancelando a implementação anterior.
9. Generalização:
Ocorre quando se utiliza uma classe contêiner (superclasse que engloba muitas classes) para representar de forma genérica todas elas. Assim, o código acaba sendo mais reaproveitável e permite tratá-las como iguais nas referências futuras.