

Linguagem de Programação C++

Universidade Federal de Santa Catarina

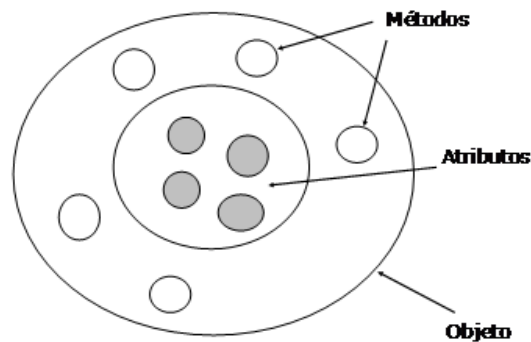
Departamento de Engenharia Elétrica, CTC

Prof. Eduardo Augusto Bezerra

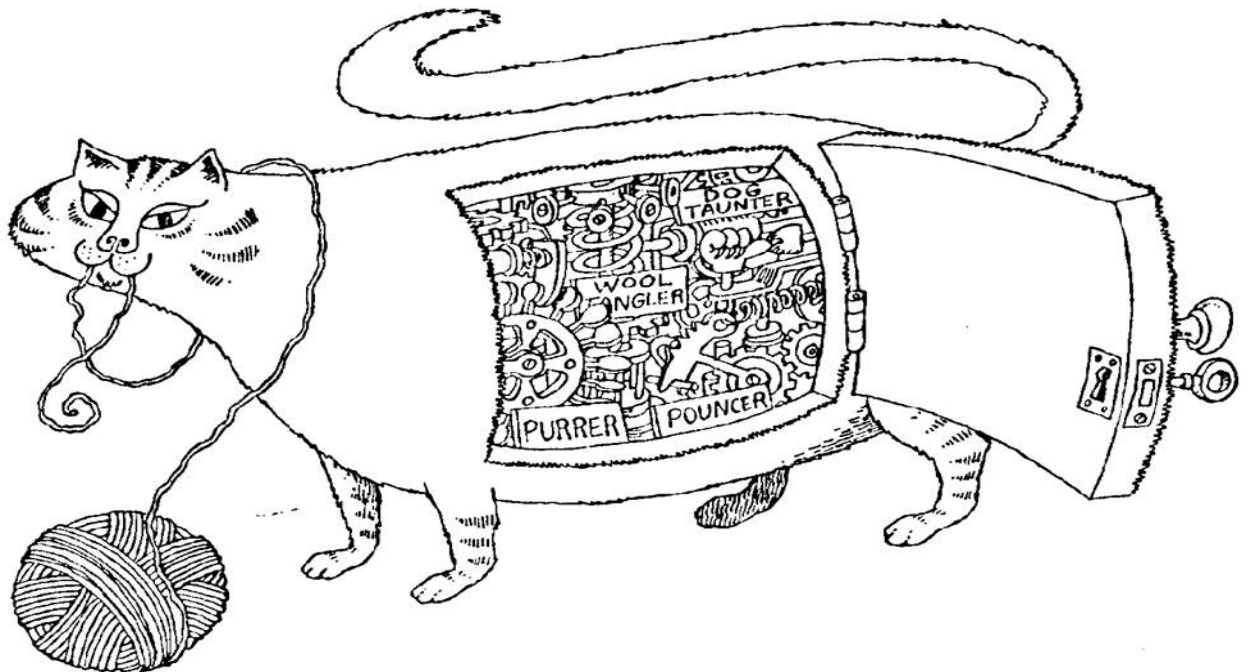
Profa. Cristiane Woszezenki

Encapsulamento

Encapsulamento é o empacotamento dos atributos e dos métodos numa mesma classe. Isso protege os dados contra a corrupção, pois somente os métodos dessa classe poderão acessar/alterar suas estruturas de dados, conforme ilustrado a seguir.



Concepts



Encapsulation hides the details of the implementation of an object.

Por exemplo, existe um número determinado de coisas que se pode fazer com um equipamento de DVD. Pode-se avançar, voltar, gravar, tocar, parar, interromper e ejetar o DVD. Dentro do equipamento, porém, existem outras funções sendo realizadas como: acionar o motor, desligá-lo, acionar o cabeçote de gravação, liberar o cabeçote, e outras operações mais complexas. Essas funções são escondidas dentro do mecanismo do equipamento de DVD e não é possível acessá-las diretamente. Quando o botão *play* é pressionado, o motor é ligado e o cabeçote de reprodução acionado, mas não é necessário saber como isso é feito para usar o equipamento de DVD.

Uma das principais vantagens do encapsulamento é esconder a complexidade do código, outra, é proteger os dados. Permitindo o acesso a eles apenas através de métodos evita que seus dados sejam corrompidos por aplicações externas.

No exemplo do DVD, quando é utilizado o método gravar, sabe-se que o aparelho irá gravar informações no DVD em um formato padrão que poderá ser lido por outros aparelhos similares. Se não existisse o método gravar e fosse necessário acessar diretamente o DVD, cada pessoa poderia estabelecer uma maneira diferente de gravar informações (usando laser, ímãs, etc.). Gravar um DVD passaria a ser uma tarefa complicada, pois o usuário teria que conhecer toda a complexidade do aparelho. Depois, perderia a compatibilidade, pois não haveria uma forma de proteger os dados para que fossem sempre gravados em um formato padrão. O encapsulamento funciona então, tanto para proteger os dados como para simplificar o uso do objeto.

Declaração e visibilidade de atributos e métodos

público (+): significa que o atributo ou método faz parte da interface do objeto, podendo ser acessado a qualquer instante.

protegido (#): significa protegido de acesso externo. Só pode ser acessado pelos métodos da classe e pelos métodos das classes derivadas.

privado (-): só pode ser acessado pelos métodos da classe (métodos internos da classe e métodos *friend*).