FACULDADE DE COMPUTAÇÃO E INFORMÁTICA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Projeto de Software – Aula 1 – 2º SEMESTRE/2018
Análise, Projeto e Desenvolvimento de Sistemas de Software – Objetos e Classes

---

Objetivos:

- Apresentar os conceitos da orientação a objetos
- Entender os conceitos de Objetos e Classes
- Discutir os conceitos da orientação a objetos: abstração, encapsulamento, modularização, concorrência e persistência.

O que é Orientação a Objetos?

Organizar o software como uma coleção de objetos distintos, que incorporam estrutura de dados e comportamento.

Um pouco de história...

Alay Curtis Kay – um dos pais da Orientação a Objetos

Formulou a analogia biológica: Sistema de software funciona-se como um ser vivo:

→ Cada célula interagiria com outras células através do envio de mensagens para realizar um objetivo em comum. Além disso, cada célula se comportaria como uma unidade autônoma.

Princípios da Orientação a Objetos – Alan Kay

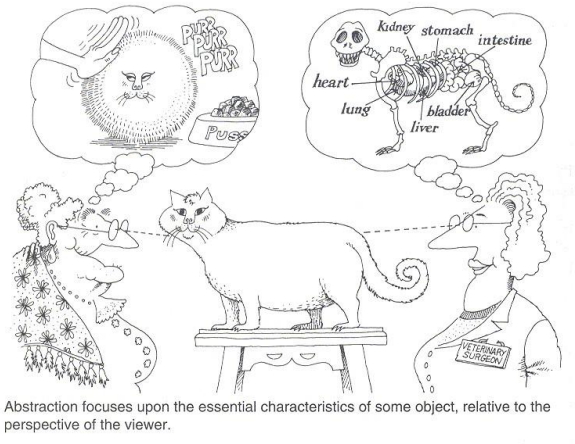
1. Qualquer coisa é um objeto
2. Objetos realizam tarefas por meio da requisição de serviços a outros objetos
3. Cada objeto pertence a uma determinada classe. Uma classe agrupa objetos similares
4. A classe é um repositório para comportamento associado ao objeto
5. Classes são organizadas em hierarquias

Extraído de: *BEZERRA, E. Princípios de análise e projeto de sistemas com UML. 2ª ed. Rio de Janeiro: Elsevier; Campus. 2007.*

Princípio da Abstração:

Permite que você se concentre nos aspectos essenciais de uma aplicação, ignorando os detalhes. Focalizar o que um objeto é e faz, antes de implementá-lo.

Abstração depende do observador



Extraído de: *BOOCH, G. Object-oriented analysis and design with applications. 3ª.ed. Addison-Wesley, 2007.*

Vamos observar as bicicletas da Figura 1. Todas elas são diferentes, mas possuem características em comum: todas tem uma cor (vermelha, preta, rosa,verde ...), são de um modelo (passeio, competição, mais esportivas, ...), marca (Caloi, Monark, Track Bike ...) e assim por diante; além disso, elas server para nos deslocar de um lugar para outro, para isso, aceleram, freiam, mudam de marcha, mediante a comandos de pedaladas, comandos de freios e mudanças de marcha.

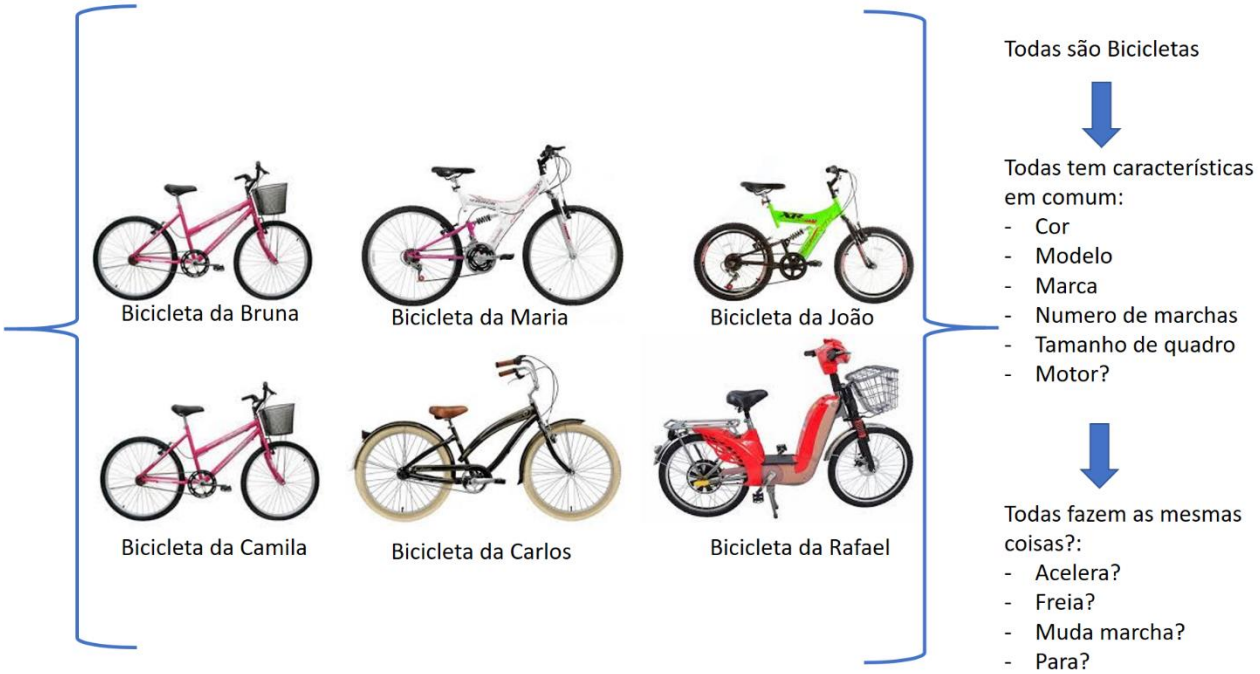


Figura 1: Uma coleção de bicicletas de várias pessoas.

Essas bicicletas representam objetos do mundo real que pertencem a mesma classe, todas elas são Bicicletas que possuem características (Cor, Modelo, Marca, Número de Marchas, Tamanho de Quadro, algumas tem motor elétrico, outras não) e fazem as mesmas coisas (aceleram, freiam e mudam de marcha.

Ou seja, os objetos com a mesma estrutura de dados (características) e comportamento (ações que fazem) são agrupados em uma classe. A **classe** é uma abstração que descreve propriedades importantes em um determinado contexto e ignora as demais que não são relevantes naquele contexto. Cada **objeto** é considerado uma instância de uma classe, ou seja, uma ocorrência daquela classe no mundo real.

A **identidade de um objeto** significa que os dados são quantizados em entidades distintas e distinguíveis chamadas objetos, como por exemplo as bicicletas da Bruna e da Camila (Figura 1), cada uma delas tem uma identidade única e inerente, ou seja, dois objetos distintos mesmo que todos os seus valores de propriedades sejam idênticos, são objetos distintos.

No mundo real o objeto simplesmente existe. Mas em linguagem de programação, cada objeto possui uma referência única pela qual ele pode ser acessado. Essas referências dos objetos são uniformes e independentes do conteúdo dos objetos, permitindo a criação de coleções mistas de objetos.

Veja alguns conceitos importantes da Orientação a Objetos

Encapsulamento - Ocultamento de informações

- Separa os aspectos externos de um objeto, que são acessíveis a outros objetos, dos detalhes internos da implementação, que são escondidos em outros objetos.

Para Booch (2007) o **encapsulamento** é o processo de compartimentar os elementos de uma abstração que constitui sua estrutura e comportamento; encapsulamento serve para separar a interface contratual de uma abstração e sua implementação.

Podemos concluir que abstração e encapsulamento são conceitos complementares? Reflitam:

- Abstração foca sobre o comportamento observável de um objeto
- Encapsulamento foca sobre a implementação que dá origem a este comportamento. O encapsulamento fornece barreiras explícitas entre as diferentes abstrações e assim trata de uma clara separação de preocupações (interesses).

Observe a Figura 2.

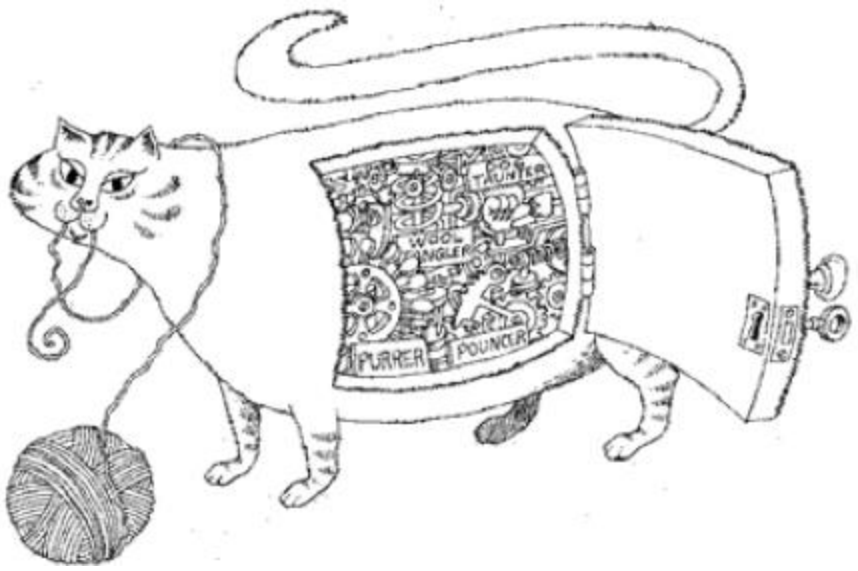


Figura 2: Representação do encapsulamento, extraído de BOOCH (2007).

"Encapsulamento esconde detalhes da implementação de um objeto." (BOOCH, 2007).

Modularização

Modularização consiste em dividir um programa em módulos que podem ser compilados separadamente, mas que tem conexões com os outros módulos. As conexões entre módulos são as suposições que os módulos fazem a respeito dos outros

A modularidade é uma propriedade de um sistema que tem sido decomposto em um conjunto coeso e fracamente acoplados de módulos.

Figura 3: Representação da Modularização, extraído de Booch (2007).

"A modularidade empacota as abstrações em unidades discretas" (BOOCH; 2007).

Os princípios de abstração, encapsulamento e modularidade são sinérgicos?

Concorrência

Concorrência é a propriedade que distingue um objeto ativo d um objeto não ativo.

Concorrência foca sobre o processo de abstração e sincronização. Mas o que isso significa?

- Cada objeto (elaborado a partir de uma abstração do mundo real) pode representar um segmento separado de controle (uma abstração de processo). Tais objetos são chamados de ativos. Em um sistema baseado em projeto orientado a objetos, podemos conceituar o mundo como consistindo de um conjunto de objetos em cooperação, alguns dos quais estão ativos e, assim, servem como centros de atividade independente.

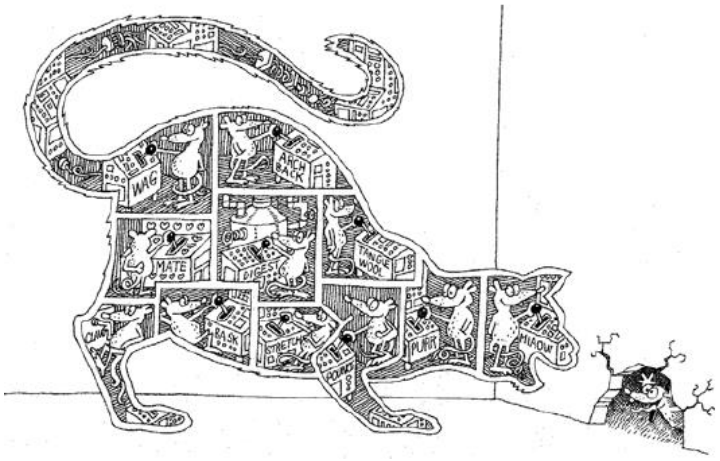


Figura 4: Representação da concorrência, extraído de Booch (2007).

"" A concorrência permite que diferentes objetos atuar ao mesmo tempo." (BOOCH; 2007)

Persistência

Um objeto em software ocupa certa quantidade de espaço e existe para um determinado propósito por período de tempo.

Persistência é a propriedade de um objeto através do qual sua existência transcende

- tempo: o objeto continua a existir depois de o seu criador deixa de existir e / ou;
- espaço: localização movimentos do objeto do espaço de endereço no qual foi Criado.

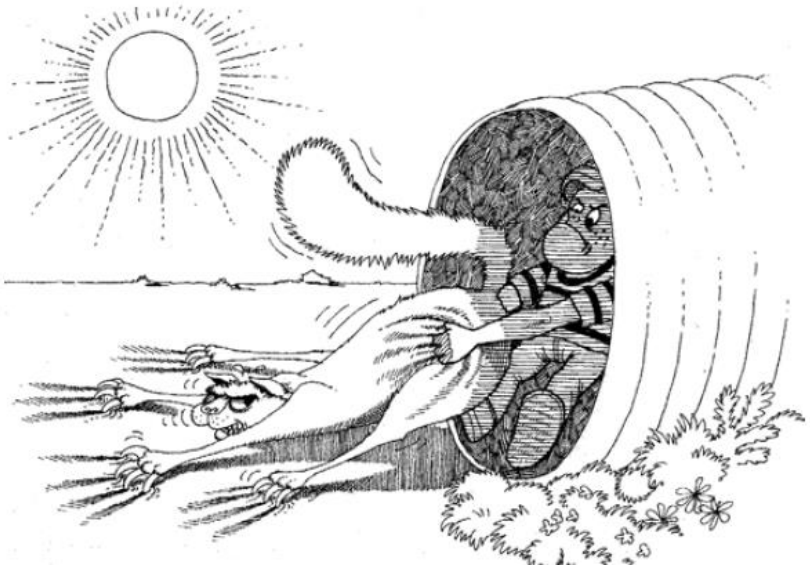


Figura 5: Representação da persistência, extraído de Booch (2007).

"A persistência salva o estado e classe de um objeto através do tempo e espaço."
(BOOCH; 2007)

Referências bibliográficas

BOOCH, G. Object-oriented analysis and design with applications. 3ª.ed. Addison-Wesley, 2007.