Bootcamp Java 12/10/22

Objetivos

* Possibilitar ao aluno compreender todos os conceitos relativos à orientação a objetos

Percurso

* Por que usar?
* Os fundamentos
* A estrutura
* As relações
* A organização
* Próximos passos

Requisitos

* Lógica de programação
* Vontade de aprender

Aula 1

A linguagem C é do paradigma de programação estruturado, o Linux é feito quase totalmente em C e o Windows também.

Porém ele possui algumas limitações

Programação estruturado x orientação a objetos

* Estruturado tem uma representação mais simplista, já a orientação a objeto uma mais realista
* Estruturado foca em operações (funções) e dados, já a orientação foca na modelagem de entidades e nas interações entre elas.
* Estruturada foca mais no como fazer, a orientação foca mais no que fazer

Vantagens

* Melhor Coesão
* Melhor acoplamento
* Diminuição do Gap semântico
* Coletor de lixo

Resumo

Desenho de personagem de desenho animado

Descrição gerada automaticamente com confiança média

Fundamentos

Objetivos

* Definir o que é a orientação a objetos
* Entender os pilares no qual o paradigma se sustenta

Definição

A orientação a objeto é um paradigma de análise, projeto e programação de sistemas de software baseado na composição e interação entre diversas unidades de software chamada de objetos.

Fundamentos

* Abstração: processo no qual se isola as características de um objeto, considerando os que tenham em comum certos grupos de objetos. Muito focado no reuso, pegando o essencial
* Reuso: capacidade de criar unidades de código a partir de outras já existentes.
* Encapsulamento: capacidade de esconder complexidades e proteger dados, ocultando os valores de uma classe protegendo seus dados.

Exercitando

Modelando uma entidade livro

Capa

Páginas

preço

Tendo o numero de páginas e o preço como dados

Resolução

Gráfico

Descrição gerada automaticamente com confiança baixa

Concluindo, não há um padrão de quantidade de elementos. Mas sim pegar os que são necessários.

Sempre modele nas necessidades que temos hoje, para não adiciona coisas que não serão usadas.

Trabalhe com o presente, e acrescente quando venham as necessidades. Fazendo manutenções evolutivas.

Estrutura básica da Orientação a objetos

Objetivo

* Apresentar os conceitos que criam as estruturas básicas da OO:
* Classe, atributo, método, objeto, mensagem

Classe

É uma estrutura que abstrai um conjunto de objetos com características similares. Uma classe define o comportamento de seus objetos através de métodos e os estados possíveis através de atributos. Em outros termos, uma classe descreve os serviços providos por seus objetos e quais informações eles podem armazenar.

A classe é a base de tudo, onde os objetos ficam e suas características também.

Exemplos de classes: carros, bolas, computadores, venda, comprador, cliente, venda.

Podemos representar entidades concretas e abstratas.

Concretas objetos físicos e abstratas ações e atividades.

Dicas de criação:

* Substantivos
* Nomes significativos
* Contexto deve ser considerado

Códigos: como criar em Linguagens

Gráfico de dispersão

Descrição gerada automaticamente com confiança média

Exercício

Uma imagem contendo Interface gráfica do usuário

Descrição gerada automaticamente

Atributo

Elemento de uma classe responsável por definir sua estrutura de dados. O conjunto destes será responsável por representar suas características e fará parte dos objetos criados a partir da classe.

Características que representa o conceito do mundo real, como exemplo em carro temos cor, ou o comprador o nome.

Atributo x variável

Atributo: o que é próprio e peculiar a alguém ou a alguma coisa

Variável: Sujeito a variação ou mudança, que pode variar; inconstante, instável

Dicas

* Substantivos e adjetivos
* Nomes significativos
* Contexto dever ser considerado
* Abstração
* Tipos adequados

Definição de atributos Código

Uma imagem contendo Gráfico de dispersão

Descrição gerada automaticamente

Exercício 2

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

Método

Uma porção de código que é disponibilizada pela classe. Este é executado quando é feita uma requisição a ele. Um método serve para identificar quais serviços, ações, que a classe oferece. Eles são responsáveis por definir e realizar um determinado comportamento.

Exemplos

Venda: calcular total

Comprador: realizar troca

Carro: Ligar

Computador: desligar

Criação de métodos

Interface gráfica do usuário

Descrição gerada automaticamente

Dicas

* Verbos
* Nomes significativos
* Contexto deve ser considerado

Criação de métodos em código

Uma imagem contendo Gráfico de dispersão

Descrição gerada automaticamente

Métodos especiais

* Construtor

Uma imagem contendo Texto

Descrição gerada automaticamente

* Destrutor

Interface gráfica do usuário

Descrição gerada automaticamente com confiança baixa

Sobrecarga de Métodos

Mudar a assinatura de acordo com sua necessidade, assinatura = nome+ parâmetro.

Texto

Descrição gerada automaticamente

Para manter o método, adicionando vários Parâmetros.

Exercício 3

Interface gráfica do usuário, Texto

Descrição gerada automaticamente