

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E
TECNOLOGIA DO AMAZONAS**

Helione dos Santos Meireles

**Trabalho: Abstração, Objeto, Herança, Classes, Encapsulamento e
Polimorfismo.**

Manaus, Amazonas – Brasil
2018

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIAS E TECNOLOGIA
DO AMAZONAS**

Helione dos Santos Meireles

**Trabalho: Abstração, Objeto, Herança, Classes, Encapsulamento e
Polimorfismo.**

**“Trabalho interdisciplinar do Curso
Técnico Integrado em Informática do
Instituto Federal de Educação, Ciências
e Tecnologia do Amazonas – IFAM
Campus Manaus - Centro, como
requisito para o cumprimento da
disciplina de Linguagem De
Programação III”**

Prof : ANTÔNIO FERREIRA DOS SANTOS JÚNIOR

Manaus, Amazonas – Brasil
2018

Trabalho: Abstração, Objeto, Herança, Classes, Encapsulamento e Polimorfismo.

1. O que é abstração?

É utilizada para a definição de entidades do mundo real. Sendo onde são criadas as classes. Essas entidades são consideradas tudo que é real, tendo como consideração as suas características e ações.

Entidade	Características	Ações
Carro, Moto	tamanho, cor, peso, altura	acelerar, parar, ligar, desligar
Elevador	tamanho, peso máximo	subir, descer, escolher andar
Conta Banco	saldo, limite, número	depositar, sacar, ver extrato

A **abstração** enquanto pilar da programação orientada a objetos consiste em trabalhar um objeto dentro da programação se preocupando somente com suas principais propriedades, sem se apegar a pontos acidentais. Esse conceito é utilizado em qualquer linguagem de programação OO (Orientada a Objetos).

2. O que é um objeto?

Um **objeto**, na vida real, é qualquer coisa a qual pudermos dar um nome.

Um **objeto**, em programação orientada a objetos, é uma instância (ou seja, um exemplar) de uma classe.

Um objeto é capaz de armazenar estados através de seus atributos e reagir a mensagens enviadas a ele, assim como se relacionar e enviar mensagens a outros objetos.

Atributos são características de um objeto. Basicamente a estrutura de dados que vai representar a classe.

Exemplos: um objeto da classe "Funcionário" teria como atributos "nome", "endereço", "telefone", "CPF", etc.

O conjunto de valores dos atributos de um determinado objeto é chamado de estado.

3. O que são classes?

Uma **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 destes objetos - através de **atributos**.

Em outras palavras, uma classe descreve os serviços oferecidos por seus objetos e quais informações eles podem armazenar.

4. Represente uma classe com pelo menos dois atributos e dois métodos (exceto gets e sets).

Filmes
+nome - String
+genero - String
+empresa - String
+contarEnredo() void
+contagiarPublico() void

5. Faça um projeto em java que represente a classe da questão 4. Este deverá ter duas classes, onde a outra fará uso da classe criada, a partir do método main. Explique como seu programa do método maior está usando a classe. Obs: não usar encapsulamento neste exemplo, ou seja, os atributos deverão ser acessados diretamente.

Uma variável pode mudar o estado de um objeto, comunicando-se com o método invocado.

Na questão 4 mostra as características de filmes, mas os valores que são guardados nas variáveis são diferentes variando para cada filme.

Foi usada uma classe Filme instância da Classe Cinema para dessa maneira garantir o acesso aos atributos diretamente.

Disponível pasta 5 no Git

6. Explique o conceito de encapsulamento e como o mesmo é implementado em java. Crie uma classe com pelo menos três atributos usando este conceito. Faça uma outra classe, contendo o método do main, e que faça uso desta classe. Explique como o método main faz acesso aos atributos da classe. Explique também o que acontece quando se tenta acessar os atributos diretamente.

É a técnica utilizada para esconder uma ideia, ou seja, não expor detalhes internos para o usuário, tornando partes do sistema mais independentes possível.

Em um processo de encapsulamento os atributos das classes são do tipo **private**. Para acessar esses tipos de modificadores, é necessário criar métodos **setters** e **getters**. Por isso quando tentamos acessar diretamente o atributo não é possível, pois está privado.

Por entendimento os métodos setters servem para alterar a informação de uma propriedade de um objeto. E os métodos getters para retornar o valor dessa propriedade.

Disponível pasta 6 no Git

7. Crie uma classe que, usando o conceito de encapsulamento, implemente um vetor de objetos da classe da questão 6. Esta classe deverá permitir cadastro, listagem, exclusão e alteração de objetos no vetor. Observe no vetor. Observe que é necessário criar um método toString () na classe. Para uso desta classe, crie um menu de opções.

Disponível pasta 7 no Git

8. Explique o conceito de herança. Apresente um exemplo contendo pelo menos 3 classes, definindo, dentro do conceito de herança, os tipos desta classes. Esse exemplo deve ser descrito implementado em java. Deverá ser feita uma outra classe, contendo o método main, que use as classes anteriores. Descreva como está sendo feito. Também utilize o conceito de polimorfismo neste método main.

A herança é um princípio da POO que permite a criação de novas classes a partir de outras previamente criadas. Essas novas classes são chamadas de subclasses, ou classes derivadas; e as classes já existentes, que deram origem às subclasses, são chamadas de superclasses, ou classes base. Nesse caso já existentes. Absorvem atributos e comportamentos, e incluem os seus próprios.

Disponível pasta 8 no Git

9. Descreva o que é uma classe abstrata, apresentando um exemplo tal como foi feito na questão 8, incluindo o polimorfismo. Explique o que acontece caso um método abstrato não seja implementado.

As classes abstratas são as que não permitem realizar qualquer tipo de instância. São classes feitas especialmente para serem modelos para suas classes derivadas. As classes derivadas, via de regra, deverão sobrescrever os métodos para realizar a implementação dos mesmos.

É possível, porém, não atribuímos uma funcionalidade a esses métodos abstratos. Neste caso, faz-se necessário, pelo menos, declarar tais métodos. Método abstrato não pode ter implementação, por isso não podemos criar seu corpo, usando chaves ({ e }), como num método comum. o método Reajustar() não pode declarar um corpo porque está marcado como abstrato.

Disponível pasta 9 no Git

10. Explique o que é polimorfismo e como este conceito foi usado nas questões 8 e 9.

No caso de polimorfismo, é necessário que os métodos tenham exatamente a mesma identificação, sendo utilizado o mecanismo de redefinição de métodos, que é o mesmo que sobrescrita de métodos em classes derivadas. Nesses casos a redefinição ocorre no método cuja assinatura já tenha sido especificada e recebe uma nova definição, ou seja, um novo corpo, em uma classe derivada.

Trabalho

Obs:

- Não usar classes já definidas em exercícios feitos em sala de aula
- Nenhum aluno poderá usar classe que outros já usavam.
- Trabalho deverá ser entregue através do git. Encaminhe um e-mail para antonio@ifam.edu.br, bem como sua conta no git.
- Além dos projetos e programas em java, adicione ao git um arquivo word ou pdf com as explicações de todas as questões
- Trabalho valendo nota 10
- Não será cedido tempo em sala de aula para fazer este trabalho
- Entrega dia 13/11/18
- Poderão ser tiradas dúvidas ao fim das aulas.

Referências

Sites

Disponível em: <https://www.devmedia.com.br/abstracao-encapsulamento-e-heranca-pilares-da-poo-em-java/26366>

Acessado em 10 de Outubro de 2018 às 16h16min.

Disponível em: <https://www.scriptcaseblog.com.br/abstracao-programacao-orientada-a-objetos/>

Acessado em 11 de Outubro de 2018 às 19h33min.

Disponível em:

https://pt.wikibooks.org/wiki/Programa%C3%A7%C3%A3o_Orientada_a_Objeto/Classes_e_Objeto

Acessado em 11 de Outubro de 2018 às 20h14min.