

Classes Abstratas em Java

1

1

Classe Abstrata

- As classes abstratas têm por finalidade estabelecer padrões de comportamento que precisam ser implementados pelas subclasses;
- Elas são utilizadas quando não é possível implementar um comportamento geral dentro da superclasse, portanto se delega sua implementação para as suas subclasses;
- Classes abstratas possuem métodos abstratos;
- Um método abstrato é um método que não possui implementação, apenas a sua assinatura;
- Por não ter todos os métodos implementados, classes abstratas não podem ser instanciadas.

2

2

Classe Abstrata

- Classes concretas que estendem classes abstratas devem implementar todos os métodos abstratos herdados, caso isso não ocorra, ela também será abstrata;
- Uma classe abstrata pode estender uma classe concreta. Entretanto, ela só se torna abstrata se definir um método abstrato;
- Pode-se criar uma hierarquia de classes abstratas. No entanto, a primeira classe concreta da hierarquia deverá implementar todos os métodos abstratos de todas as classes abstratas de tal hierarquia.

3

3

Métodos Abstratos

- Métodos abstratos possuem somente a declaração de suas assinaturas:
 - Portanto, eles têm apenas:
 - Modificadores, tipo de retorno, nome e lista de parâmetros.
- Métodos abstratos não têm um corpo com a lógica de implementação de seu comportamento;
- Métodos abstratos são declarados com o modificador **abstract**;
- Se existir um método abstrato, a classe deve ser declarada como abstrata por meio do uso do modificador **abstract**.

4

4

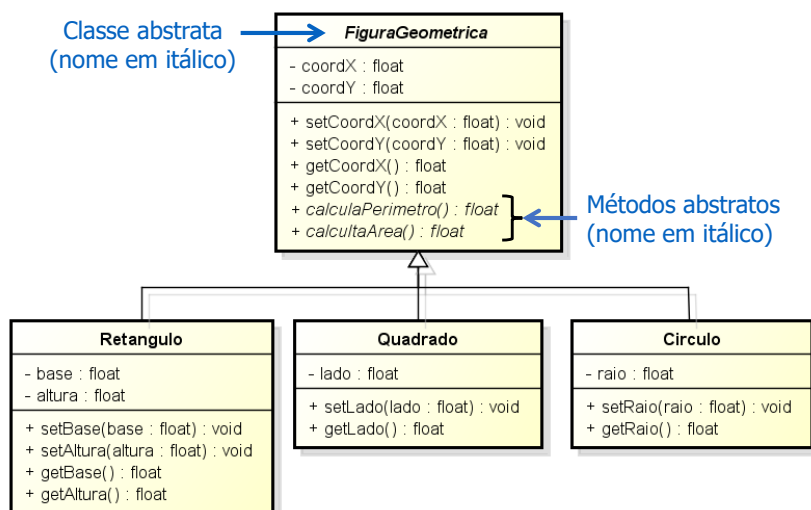
Características

- A subclasse herda atributos e métodos de sua superclasse abstrata;
- Os atributos e métodos privados não são herdados;
- Todos os métodos abstratos devem ser implementados da subclasse, senão a subclasse deverá ser declarada abstrata.

5

5

Modelagem de Herança com Classe Abstrata em UML



powered by astah

6

6

Sintaxe de Declaração de Classe e Métodos Abstratos

- Sintaxe de declaração de uma Classe Abstrata em Java:

[mod] abstract class NomeDaClasse { ... }

- Sintaxe de declaração de um Método Abstrato em Java:

[mod] abstract tipoRetorno nomeDoMétodo([parâmetros]);

7

7

Exemplo de Classe Abstrata

Classe abstrata →

```
public abstract class FigurasGeometricas {
    private float coordX;
    private float coordY;

    public abstract float calculaPerimetro();

    public abstract float calculaArea();

    public void setCoordX(float coordX) {
        this.coordX = coordX;
    }

    public float getCoordX() {
        return coordX;
    }
    ...
}
```

8

8

Sintaxe de Utilização de uma Classe Abstrata

- Sintaxe de utilização de uma Classe Abstrata em Java:

```
[mod*] class NomeDaSubClasse extends NomeDaSuperClasseAbstrata  
{  
    //implementação dos métodos abstratos  
}
```

- Exemplo:

```
public class Retangulo extends FigurasGeometricas  
{  
    ...  
}
```

9

9

Exemplo de Utilização de uma Classe Abstrata

```
public class Retangulo extends FiguraGeometrica  
{  
    private float base;  
    private float altura;  
  
    public float calculaPerimetro() {  
        return base()*2 + altura()*2;  
    }  
  
    public float calculaArea() {  
        return base() * altura();  
    }  
}
```

10

10

Polimorfismo em Classes Abstratas

- Assim como acontece na herança de classe concreta, a herança de classe abstrata possibilita o polimorfismo;
- Entretanto, neste caso somente entre as subclasses da superclasse abstrata;
- Cada método abstrato implementado em uma subclasse concreta dará um comportamento diferente para um método definido na superclasse abstrata;
- A utilização do polimorfismo é facilitada pela utilização da ligação tardia.

11