

# while VS do-while 1/2

while (em PT: enquanto) verifica antes a condição para executar o bloco. Já o do-while executa o bloco e verifica depois a condição para repetir o laço.

GEEK 2 CODE

```
while (!naBorda( )) {  
    corra();  
}
```

```
do {  
    corra();  
} while (!naBorda( ));
```



\*Ambos iniciando a corrida na borda

# Agora...

## Classes abstratas x Interfaces;

# Debate

## Classe Abstrata

- Permitem novas funcionalidades sem modificar um código existente;
- Polimorfismo simplifica o código;
- Força a hierarquia para todas as subclasses;
- Não pode ser instanciada!
- Pode possuir construtor que pode ser utilizado pelas classes filhas;
- Eficiente na reutilização de código;
- Facilidade de manutenção;
- Organização de código;
- Permite declarar métodos sem implementação.

Pergunta:

- Herança: É possível herdar 2 classes abstratas?
- Como ela permite declarar métodos já que não tem implementação?  
ex: public abstract void exemplo();

# Debate

## Interface

- Flexibilidade;
- Múltiplas implementações;
- Não podem ser herdadas, nem instanciadas;
- Desacoplamento;
- CONTRATO!
- Padronização;
- Eficiente em documentação.

## Pergunta:

- É possível adicionar um novo método após sua construção?
- Qual são os comportamentos das variáveis e dos métodos na interface?  
public class animal implements SerVivo{}

# Classes Abstratas

- São classes feitas especialmente para servirem de modelo para suas classes derivadas;
- Estas classes **não** permitem realizar qualquer tipo de instância;
- As classes derivadas deverão **sobrescrever os métodos** para realizar a implementação dos mesmos;
- As classes derivadas das classes abstratas são conhecidas como classes concretas;
- Caso um ou mais métodos abstratos estejam presentes nessa classe abstrata, a classe filha será, então, forçada a definir tais métodos para que ela não se torne abstrata.

# Interfaces

- Interfaces são padrões definidos através de contratos ou especificações;
- Um contrato define um determinado conjunto de métodos que serão implementados nas classes que assinarem esse contrato;
- Uma interface é 100% abstrata, ou seja, os seus métodos são definidos como abstract, e as variáveis por padrão são sempre constantes (**static final**);
- Uma interface é definida através da palavra reservada “interface”.
- Para uma classe implementar uma interface é usada a palavra “implements”.

# Interfaces

- A linguagem Java não tem herança múltipla (classes derivadas de mais de uma superclasse) e as interfaces ajudam nessa questão;
- Uma classe pode ser herdada apenas uma vez, mas pode implementar inúmeras interfaces;
- As classes que forem implementar uma interface terão de adicionar todos os métodos da interface ou se transformar em uma classe abstrata.

# Interfaces X Classes

Característica	Interface	Classe Abstrata
Herança múltipla	Uma classe pode implementar diversas interfaces.	Uma classe pode herdar somente uma classe.
Implementação Padrão	Uma interface não pode conter qualquer tipo de código, muito menos código padrão.	Uma classe abstrata pode fornecer código completo, código padrão ou ter apenas a declaração de seu esqueleto para ser posteriormente sobreescrita.
Constantes	Suporta somente constantes do tipo estática.	Pode conter constantes estáticas e de instância.
Componentes de terceiros	Uma implementação de uma interface pode ser incluída a qualquer classe de terceiros.	Uma classe de terceiros precisa ser reescrita para estender somente a partir da classe abstrata.
Homogeneidade	Se todas as diversas implementações compartilham a assinatura do método então a interface funciona melhor.	Se as várias implementações são todas do mesmo tipo e compartilham um comportamento e status comum, então a classe abstrata funciona melhor.
Velocidade	Lento, requer trabalho extra para encontrar o método correspondente na classe atual.	Rápido
Clareza	Todas as declarações de constantes em uma interface são presumidamente públicas e estáticas.	Você pode pôr código compartilhado e usar código para computar o valor inicial de suas constantes e variáveis de instância ou estáticas.
Funcionalidades Adicionais	Se você incluir um novo método em uma interface você precisa ajustar todas as implementações da interface.	Se você incluir um novo método em uma classe abstrata você tem a opção de fornecer uma implementação padrão para ele.

# Resumindo...

Classe abstrata:

- não pode ser instanciada.

Método abstrato:

- quem está herdando do pai tem que ter sobrescrita do método @Override;
- só tem definição, não tem implementação.

Interface:

- não pode ser instanciada.

A diferença da interface para a classe abstrata é que a interface não possui construtores e nem implementações de métodos, somente definições de atributos que podem conter um valor fixo ou constante e, também, definições de métodos. Atributos e definições de métodos sempre serão públicos e abstratos. Interfaces permite herança múltipla, significa um extends de qualquer outra interface. Em uma classe a palavra reservada implements serve para assinar um contrato com aquela interface.