Scala Fundamentals

Uma linguagem de programação escalável

Michael Silva

Código Escalável

Forma de medir a quantidade de alterações que devem ser feitas para implementar um novo requisito.

1. Dificuldade das alterações
2. Quantidade de linhas de código alteradas
3. Quantidade de partes (códigos) repetidas (não reuso)
4. Modelo de dados

Programação Orientada a Objetos

É um paradigma de programação (forma de programar) ao qual o código pode ser estruturado em módulos **reusáveis**. Os componentes, chamados de objetos, podem conter propriedades e comportamentos - *Alan Kay*.

- A programação Orientada a Objeto (POO) encoraja:
  - **Modularização:** A aplicação pode ser decomposta em módulos
  - **Reuso:** Uma aplicação pode ser composta por novos módulos e módulos existentes

Programação Orientada a Objetos

- Conceitos Básicos
  - **Classe:** Descrição formal de um “objeto da vida real”. Composta por atributos e métodos
  - **Objeto:** Instância de uma classe em memória (área reservada para comportar seus dados únicos)
    - » Os dados de um objeto são manipulados pelos seus próprios métodos (o dono sabe o que faz)

Programação Orientada a Objetos

- Pilares
  - **Abstração:** Representação, no sistema, de um objeto real
  - **Encapsulamento:** Camada de segurança para os dados (atributos) dos objetos (*getters* e *setters*)
  - **Herança:** Reuso de código, uma classe herda (é filha) características de outra classe
  - **Polimorfismo:** As classes/objetos podem alterar sua forma conforme a necessidade (alteração interna de um comportamento herdado)

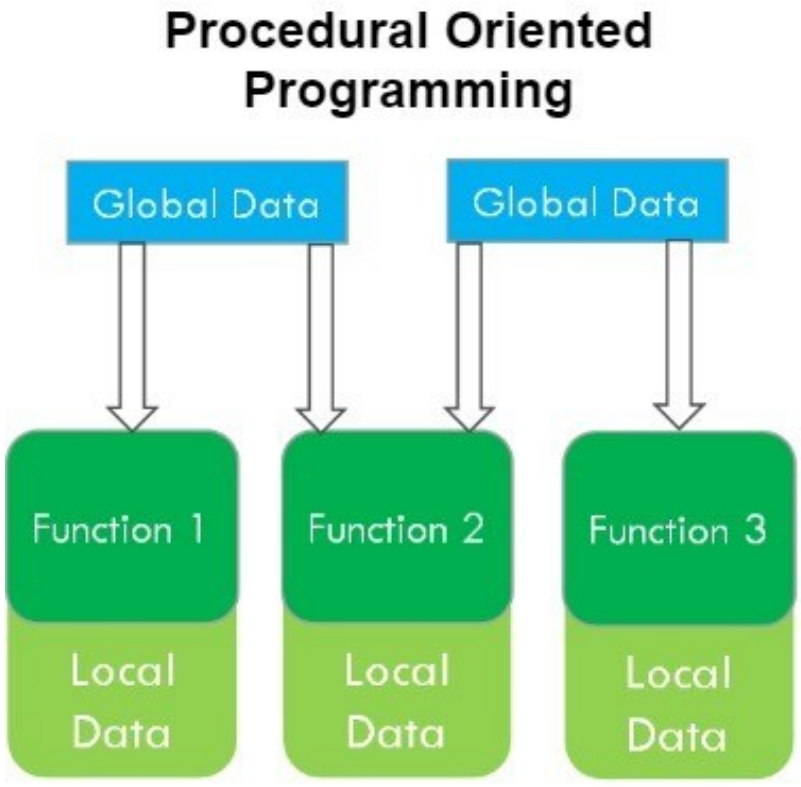
Programação Orientada a Objetos

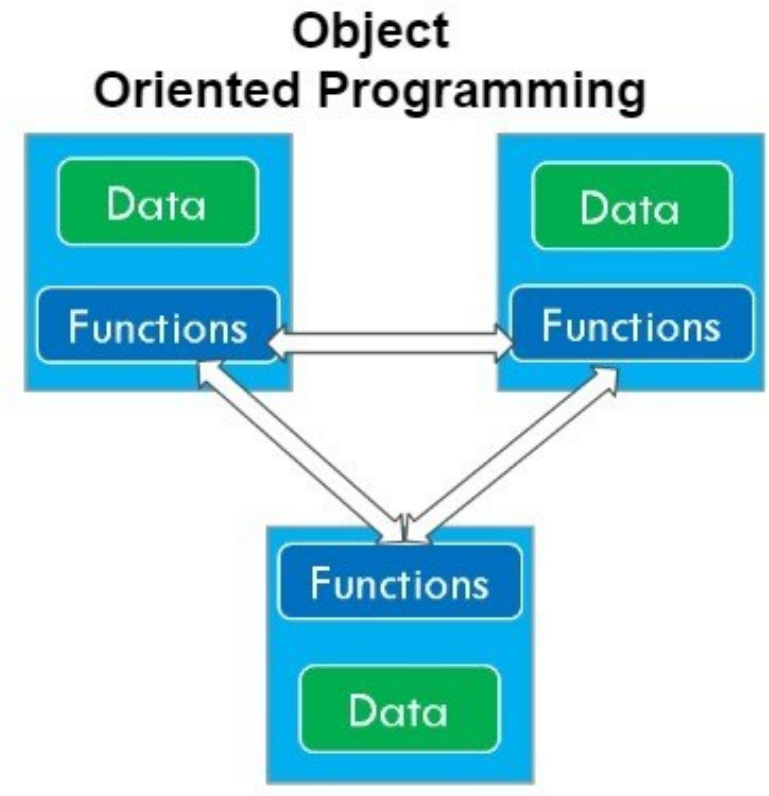
Pensando Juntos

Contribui para escalabilidade do código?

Programação Orientada a Objetos

Diferença para o paradigma procedural





Programação Orientada a Objetos

Diferença para o paradigma procedural

- Paradigma Procedural
  - Dividido em funções
  - **Funções** têm escopo geral
  - Abordagem top-down
    - » O sistema é uma entidade que se decompõe para atingir mais de um subsistema, cada subsistema atinge outros e assim por diante.

Programação Orientada a Objetos

Diferença para o paradigma procedural

- Paradigma Orientado a Objetos
  - Divididos em pequenas partes chamadas de objetos
  - **Métodos** têm escopo restrito
  - Abordagem bottom-up
    - » O sistema começa a ser desenvolvido a partir de componentes mais específicos/básicos, outros componentes, com regras de negócio mais complexas, são desenvolvidos utilizando os mais básicos

Programação Funcional

É um paradigma de programação ao qual sua operação fundamental é a aplicação de funções aos seus argumentos.

- Não contém declarações de atribuição, ou seja, sem variáveis (imutabilidade)
- Não contém efeitos colaterais
  - Os processadores lidam melhor com programas previsíveis
- **First Class Function:** Funções que são tratadas como valor
  - Serem atribuídas a uma variável/constante

Programação Funcional

- **High Order Function:** Funções têm pelo menos uma *first class function* como parâmetro
- **Exemplos:**
  - Recursividade

```
def sum(num: Int): Int = {
  if (num == 1) 1 else sum(num - 1) + num
}
```

  - Java Lambda

```
ArrayList<Integer> arrL = new ArrayList<Integer>();
arrL.forEach(n -> System.out.println(n));
```

Programação Funcional

Pensando Juntos

Contribui para escalabilidade do código?

Linguagem de programação Scala

- O nome Scala foi dado por ser uma *linguagem escalável*
- A linguagem foi projetada para crescer com as demandas dos usuários
- Pode ser utilizada tanto para projetos pequenos quanto para projetos grandes
- O compilador gera o Bytecode, esse, por sua vez, é interpretado pela JVM

Linguagem de programação Scala

- Combina o paradigma de programação funcional e orientação a objetos
  - Apartir do paradigma funcional, é possível desenvolver funcionalidades complexas rapidamente
  - Como também utiliza POO, é fácil construir uma estrutura para sistemas grandes que se adaptam a novas demandas

Linguagem de programação Scala

- Motivação para o uso
  - A linguagem é escalável?
  - O código gerado tende a ser enxuto e legível

```
// Scala
def factorial(x: BigInt): BigInt =
  if (x == 0) 1 else x * factorial(x - 1)
```

  - A estimativa mais conservadora diz que um código em Scala tem pelo menos a metade do número de linhas de um código em Java

Linguagem de programação Scala

Exemplo Java x Scala

```java
//Java
class MyClass {
   private int index;
   private String name;

   public MyClass(int index, String name) {
     this.index = index;
     this.name = name;
   }
}

//Scala
class MyClass(index: Int, name: String)
```

Linguagem de programação Scala

Exemplo Java x Scala

```java
//Java
boolean nameHasUpperCase = false;
for (int i = 0; i < name.length(); ++i) {
 if (Character.isUpperCase(name.charAt(i))) {
   nameHasUpperCase = true;
   break;
 }
}

//Scala
val nameHasUpperCase = name.exists(_.isUpper)
```

Linguagem de programação Scala

- Motivação para o uso
  - Compatibilidade total com Java. Podem compartilhar classes, tipos e métodos
  - Adiciona métodos muito úteis aos tipos do Java

```
//Scala
str.toInt

//Java
Integer.parseInt(str)
```

Linguagem de programação Scala

Prática utilizando *command line*

`github.com/MichaelRSilva/scala-fundamentals-course`

./part-i/command-line-exercises

Linguagem de programação Scala

Prática utilizando a IDE

`github.com/MichaelRSilva/scala-fundamentals-course`

./part-i/ide-exercises

Linguagem de programação Scala

Contato

michael@michaelsilva.io

Bibliografia

[1] Odersky, Martin and Spoon, Lex and Venners, Bill. *Programming in Scala*. Artima Incorporation, 2016.