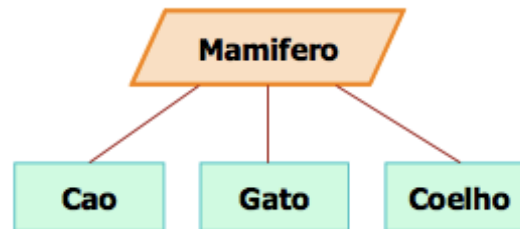


Mais sobre JAVA 8 (II)

- Package java.time
 - LocalDate; LocalTime; LocalDateTime
 - Fusos horários (Time Zones)
 - ZonedDateTime
 - Objectos são imutáveis

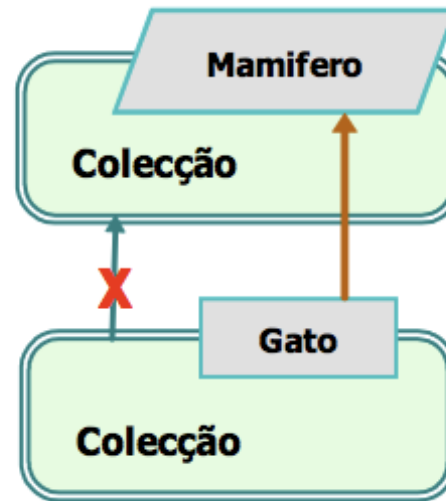
Tipos Parametrizados

- Seja a seguinte hierarquia:

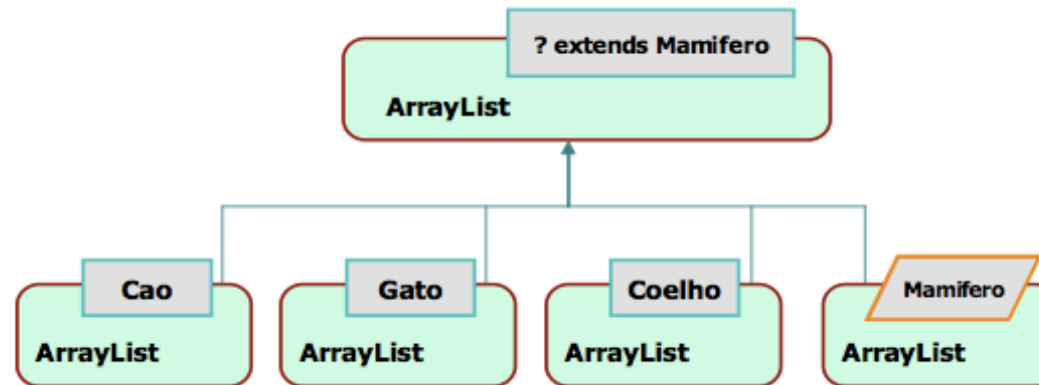


- e considere-se uma colecção de elementos do tipo Mamífero

- Um arraylist de mamíferos, `ArrayList<Mamifero>` pode conter instâncias de `Cão`, `Gato`, `Coelho`, etc.
- no entanto esse arraylist não é supertipo dos arraylist de subtipos de mamíferos!!
- A hierarquia de `ArrayList<E>` não tem a mesma estruturação da hierarquia de `E`



- O tipo dos ArrayLists de Mamifero e dos ArrayLists seus subtipos declara-se como:
`ArrayList<? extends Mamifero>`
- O tipo dos ArrayLists de super-classes de Mamifero declara-se como:
`ArrayList<? super Mamifero>`



Coleccao<**? extends Mamifero**> =
 Coleccao<Gato> **ou**
 Coleccao<Cao> **ou**
 Coleccao<Coelho>

```

public void juntaMamif(Set<? extends Mamifero> cm) {
    ...
}
  
```

- Desta forma passa a ser possível ter declarações como:

```
ArrayList<Mamifero> mamifs = new ArrayList<Mamifero>();  
mamifs.addAll(criaCaes()); // junta ArrayList<Cao>  
mamifs.addAll(criaGatos()); // junta ArrayList<Gato>
```

- o que era impossível no modelo anterior, na medida em que um `ArrayList<Cao>` não é compatível com `ArrayList<Mamifero>`

Programação Orientada aos Objectos

-- fecho do semestre --

Resultados de aprendizagem

- Compreender os conceitos fundamentais da PPO(Objectos, Classes, Herança e Polimorfismo);
- Compreender como os conceitos básicos da PPO são implementados em construções JAVA;
- Compreender princípios e técnicas a empregar em programação de larga escala;
- Desenvolver o modelo de classes e interfaces para um dado problema de software (modelação);
- Desenvolver e implementar aplicações Java de média escala, seguras, robustas e extensíveis;

Programa: teórica

- Paradigma da programação por Objectos: Abstracção de Dados, Encapsulamento e Modularidade.
- Objectos: estrutura e comportamento.
- Mensagens. Classes, hierarquia e herança. Classes abstractas.
- Herança versus Composição.
- Compatibilidade de tipos. O princípio da substituição. Dynamic binding. Polimorfismo.

Programa prática

- JAVA: Plataforma J2SE: JDK, JVM e byte-code.
- Construções básicas: tipos primitivos e operadores. Estruturas de controlo. I/O básico. Arrays.
- Nível dos objectos: Classes e instâncias. Construtores. Métodos e variáveis de instância. Modificadores de acesso. Métodos e variáveis de classe.
- Colecções genéricas. Interfaces parametrizadas. Iteradores internos e externos. Tipos List, Map e Set.
- Hierarquia de classes e herança. Overloading e overriding de métodos. Classes Abstractas. Interfaces e tipos definidos pelo utilizador. Tipo estático e dinâmico. Procura dinâmica de métodos. Polimorfismo e extensibilidade.
- Streams de input/output: de caracteres, de bytes e de objectos.
- Excepções.