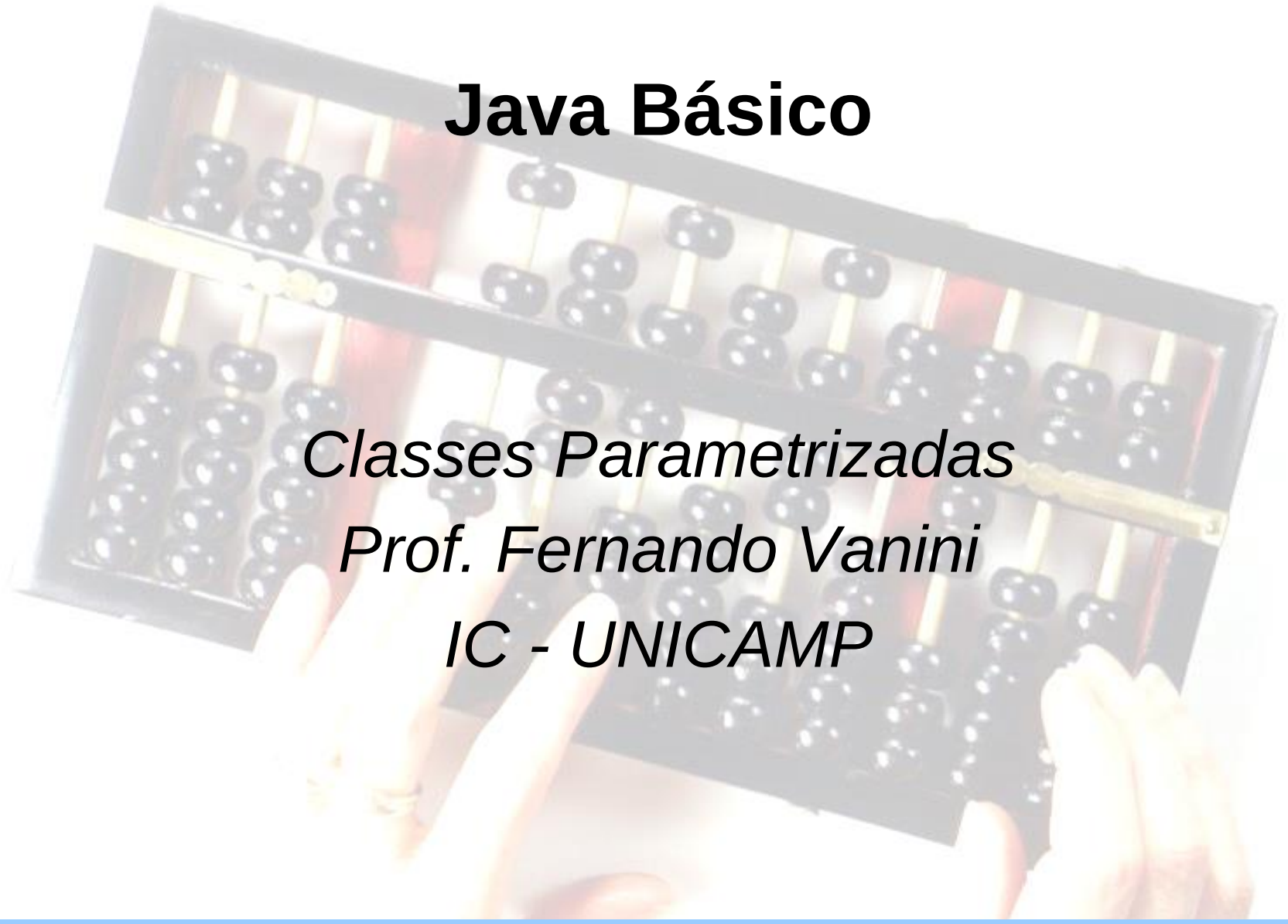

Java Básico

Classes Parametrizadas

Prof. Fernando Vanini

IC - UNICAMP

Roteiro

- Classes Parametrizadas
  - Problemas com coleções de objetos
  - Classes Parametrizadas

Objetos e Generalização

- Em Java todas as classes são derivadas direta ou indiretamente da classe Object.
- É portanto natural que se pense no uso dessa classe como parte de classes ‘genéricas’.
- Um exemplo comum:
  - as coleções (ex. ArrayList, LinkedList, etc), são voltadas a manter coleções de objetos.
  - Uma aplicação pode usar uma dessas classes para manter objetos das classes que precisar.
  - Ao acessar um objeto de uma dessas coleções, o programador precisa se lembrar qual a classe do mesmo e fazer um ‘casting’ para utilizá-lo de forma adequada.
  - O compilador não consegue verificar a validade do ‘casting’ em tempo de compilação.
  - Eventuais erros levam a exceções em tempo de execução.

Um exemplo

```
public class TopSort {
  LinkedList nodeList; /* lista c/ os nós do grafo (Node) */
  HashMap    nodeMap;  /* mapa de pares (Object,Node)     */

  private Node getNode(Object obj){
    Node node = (Node)nodeMap.get(obj);
    if(node == null) {
      node = new Node(obj);
      nodeList.addLast(node);
      nodeMap.put(obj,node);
    }
    return node;
  }
  ...
}
```

Classes Parametrizadas

- A partir da versão 5 de Java, é possível declarar uma ‘classe parametrizada’, onde cada ‘parâmetro’ indica uma classe de objetos utilizada.
- Exemplo: as classes ArrayList, LinkedList, HashMap e HashTree podem ser utilizadas de forma parametrizada.

Um exemplo

```
class Exemplo {
  LinkedList<String> lista = new LinkedList<Strings>();
  HashMap<String,Node> mapa = new HashMap<String,Node>();
  String nomes = {"um","dois","tres","quatro","cinco" };
  ...
  public Node setNode(String s, Node n){
    Node res = mapa.get(s);
    if(res == null) mapa.put(s,n);
  }

  void makeList() {
    for(int i = 0; i < nomes.length; i++) lista.add(nomes[i]);
  }

  public String nextName(){ return lista.removeFirst(); }
  ...
}
```

Criando uma classe parametrizada

- Ao declarar uma classe é possível indicar que a mesma é parametrizada colocando o nome da ‘classe parâmetro’ entre ‘<‘ e ‘>’ após o nome da mesma.
- Dentro da classe, o nome da ‘classe parâmetro’ pode ser utilizado como uma classe já definida.

Um exemplo

```
class Node<Type> {
  int preds;
  int curPreds;
  Type obj;
  LinkedList<Node<Type>> succs;

  public void addSucc(Node<Type> succ){
    succs.addLast(succ);
  }

  public Type getObj(){ return obj; }
  public int getPreds() { return preds; }
  public LinkedList<Node<Type>> getSuccs(){ return succs; }
  ...
}
```

Iteradores e o comando *for*

- A partir da versão 5 de Java, o comando *for* admite uma forma que dispensa o uso de um iterador para percorrer uma coleção. Um exemplo:

```
...
LinkedList<String> lista = new LinkedList<String>;
...
for(String nome : lista) System.out.println("==>"+nome);
...
```

Um exemplo

```
public List<Type> sort(){
   LinkedList<Type> res = new LinkedList<Type>();
   LinkedList<Node<Type>> queue = new LinkedList<Node<Type>>();
   for ( Node<Type> node: nodeList){
   node.reset();
   if(node.getPreds() == 0) queue.addLast(node);
   }
   if(queue.size() == 0) return null;
   do{
     Node<Type> node = queue.removeFirst();
     res.addLast(node.getObj());
     for(Node<Type> node2: node.getSuccs()){
       if(node2.decPreds() == 0) queue.addLast(node2);
     }
   }while (queue.size() > 0);
   if(res.size() < nodeList.size()) return null;
   return res;
 }
```