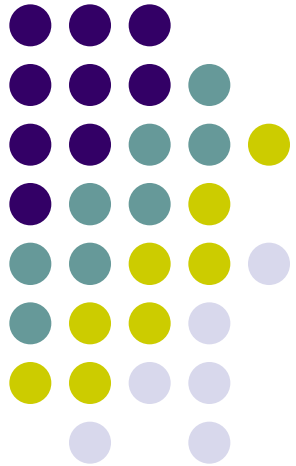
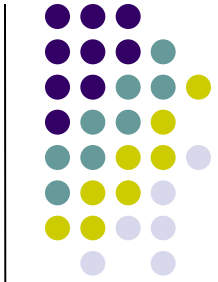
COM220
Aula 12: Manipulação de vetores com a classe Vector

Prof. Laércio Baldochi

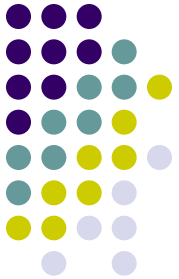


Classe Vector



- Permite criar uma coleção de objetos heterogêneos capazes de variar de tamanho dinamicamente
- Construtores
  - **public** `Vector()`
    - constrói um vetor vazio;
  - **public** `Vector(`**int** *`initialCapacity`*`)`
    - constrói um vetor vazio com a capacidade especificada;
  - **public** `Vector(`**int** *`initialCapacity`*`, int` *`capacityIncrement`*`)`
    - constrói um vetor vazio com a capacidade e incremento especificados.

Classe Vector
Métodos



- Os principais métodos dessa classe são
  - `public int size()`
  - `public boolean contains(Object elem)`
  - `public int indexOf(Object elem)`
  - `public synchronized Object elementAt(int index)`
  - `public synchronized void setElementAt(Object obj, int index)`
  - `public synchronized void removeElementAt(int index)`
  - `public synchronized void addElement(Object obj)`
  - `public synchronized void insertElementAt(Object obj, int index)`

Classe Vector
Exemplo

```
import java.util.*;

public class ExemploVector {

  private Vector vetor = null;

  public ExemploVector() {
    vetor = new Vector(1,1); //Define a capacidade e incremento
  }

  public void adicionaInt(int i) {
    vetor.addElement(new Integer(i));
  }

  public void adicionaDouble(double d) {
    vetor.addElement(new Double(d));
  }
...
```

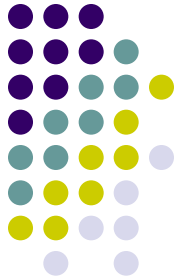
API Java. Importação de classe da API java.util.*;

Aqui está incluída a classe Vector()

Utiliza o construtor que permite definir o tamanho inicial do vetor e o incremento

O método addElement(Object) permite adicionar qualquer objeto para um determinado vetor.

Classe Vector
Exemplo



```
....
  public static void main (String par[]) {
     ExemploVector v = new  ExemploVector();
       int intValor = 5;
     double dblValor = 5.45;
     char chrArray[] = {'1','2','3','4','5'};
     String strValor = new String("Java");
     Conta nCta = new Conta(111,35458,1,150.23,100);

     //Adiciona elementos ao vetor
     v.adicionaInt(intValor);
     v.adicionaDouble(dblValor);
     v.adicionaString(strValor);
     v.adicionaCharArray(chrArray);
     v.adicionaConta(nCta);
     v.imprimeVetor();
  }
  ....
}
```

Chama os diversos métodos da classe ExemploVector para adicionar diferentes elementos para o vetor

Classe Vector
Exemplo

Define uma classe Conta no mesmo arquivo ExemploVector.java

```
static class Conta {
  private int agencia, conta, digito;
  private static double saldo, limite;
```

Construtor da classse

```
   public Conta(int pagencia, int pconta, int pdigito, double psaldo, double plimite) {
    agencia = pagencia;
    conta = pconta;
    digito = pdigito;
    saldo = psaldo;
    limite = plimite;
  }
```

Método imprime da classe Conta

```
  public void imprime() {
    System.out.println("Agencia: "+agencia+" Conta: "+conta+"-"+digito+" Saldo:
     "+saldo);
  }
}
```

Classe Vector Exemplo

```
....
 public void imprimeVetor() {
    Object obj;
    int tamanho = vetor.size();
    System.out.println("O número de elementos do vetor é "+vetor.size());
     for (int i = 0; i < vetor.size(); i++) {
       obj = vetor.elementAt(i);
        if (obj instanceof char[]) {
          System.out.println(String.copyValueOf((char[]) obj));
        } else if (obj instanceof Conta) {
          Conta cta = (Conta)obj;
          cta.imprime();
     } else { System.out.println(obj.toString()); }
    }
  }
  ....
}
```

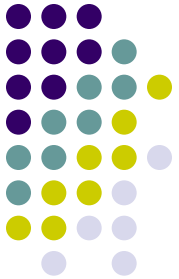
Percorre todo o vetor utilizando o método size()

Recupera um elemento específico do vetor através do método elementAt(int)

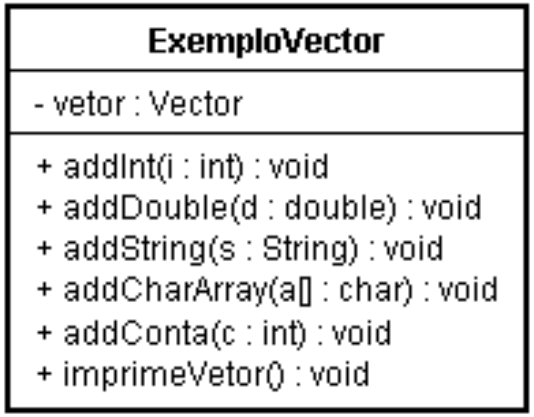
Para acessar os métodos de determinado objeto que está no vetor é preciso realizar um ***type cast*** utilizando o nome da classe entre ()

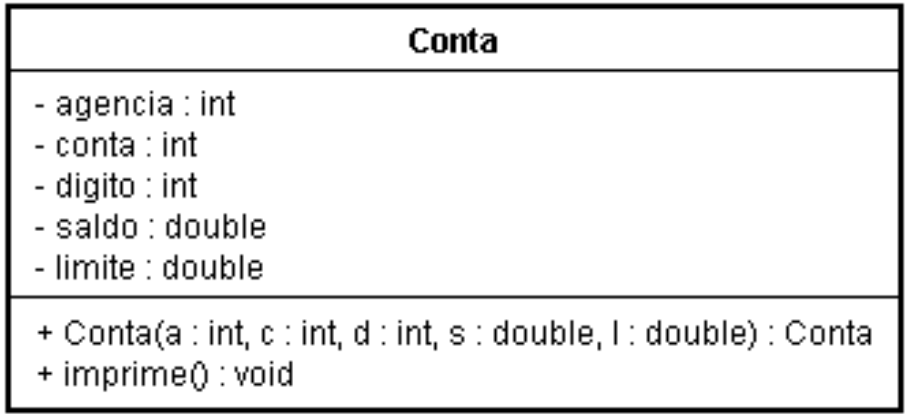
O comando **instanceof** permite descobrir o tipo de determinado objeto

Classe Vector Funcionamento

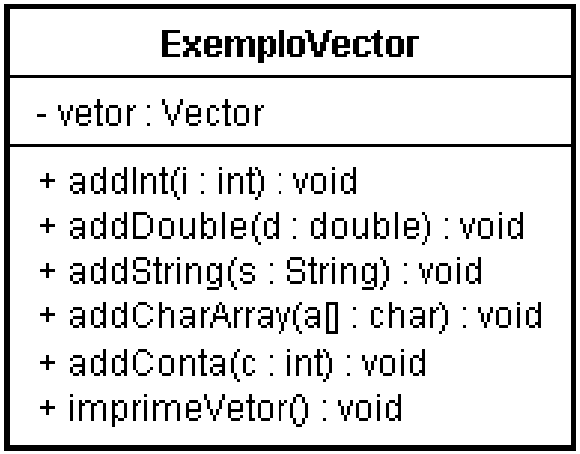


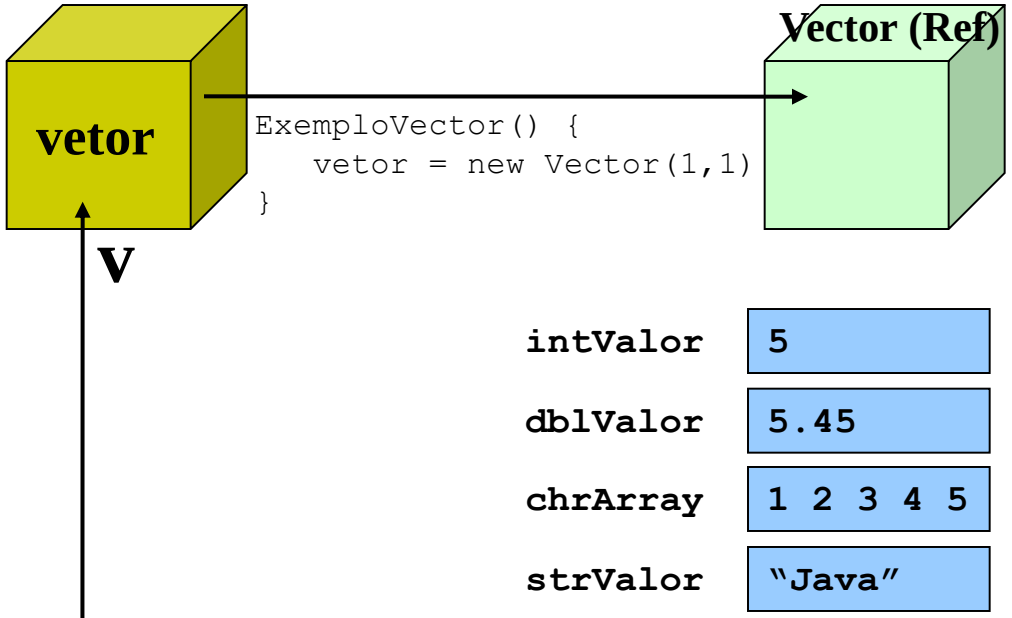
- As classes definidas em nosso exemplo são ilustradas a seguir na notação de classes UML





Classe Vector - funcionamento

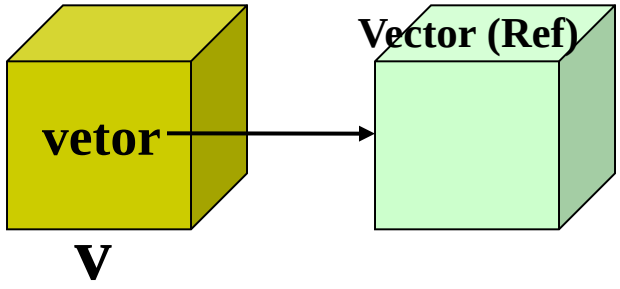


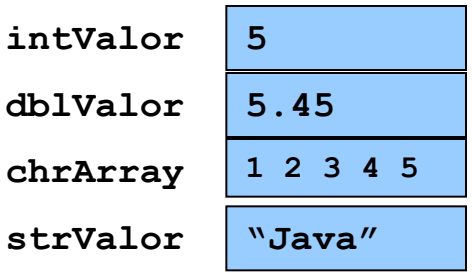



```
public static void main (String par[]) {
   ExemploVector v = new  ExemploVector();
   int intValor = 5;
   double dblValor = 5.45;
   char chrArray[] = {'1','2','3','4','5'};
   String strValor = new String("Java");
   ...
```

O construtor de ExemploVector instancia um objeto Vector definindo um tamanho inicial (1) e o incremento (1)

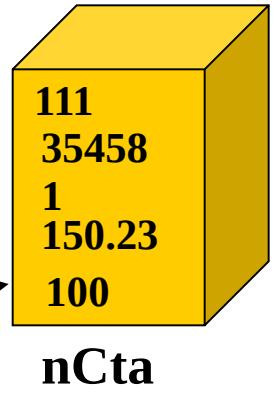
Classe Vector - funcionamento





Conta
- agencia : int - conta : int - digito : int - saldo : double - limite : double
+ Conta(a : int, c : int, d : int, s : double, l : double) : Conta + imprime() : void

```
public Conta(...) {
     agencia = pagencia;
     conta = pconta;
     digito = pdigito;
     saldo = psaldo;
     limite = plimite;
   }
```

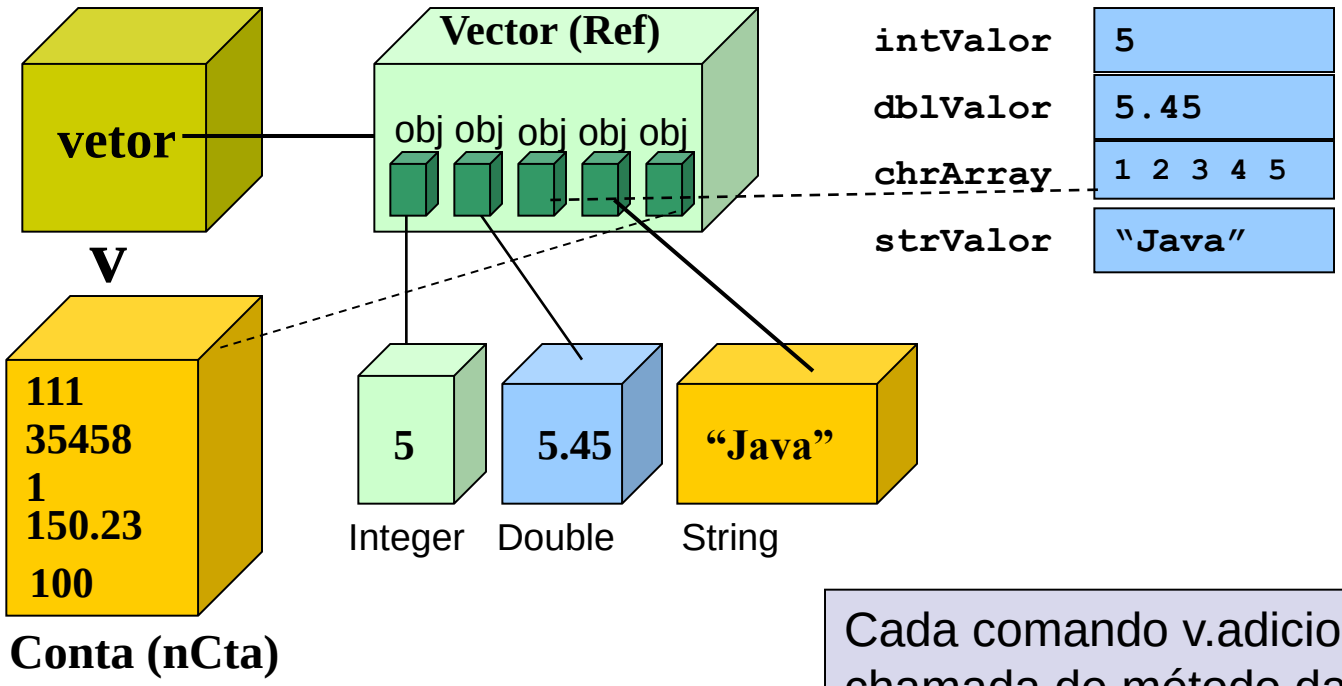


Conta nCta = new Conta(111,35458,1,150.23,100);

...

A seguir, o programa define o objeto nCta,
usando a classe interna (Inner Class) Conta

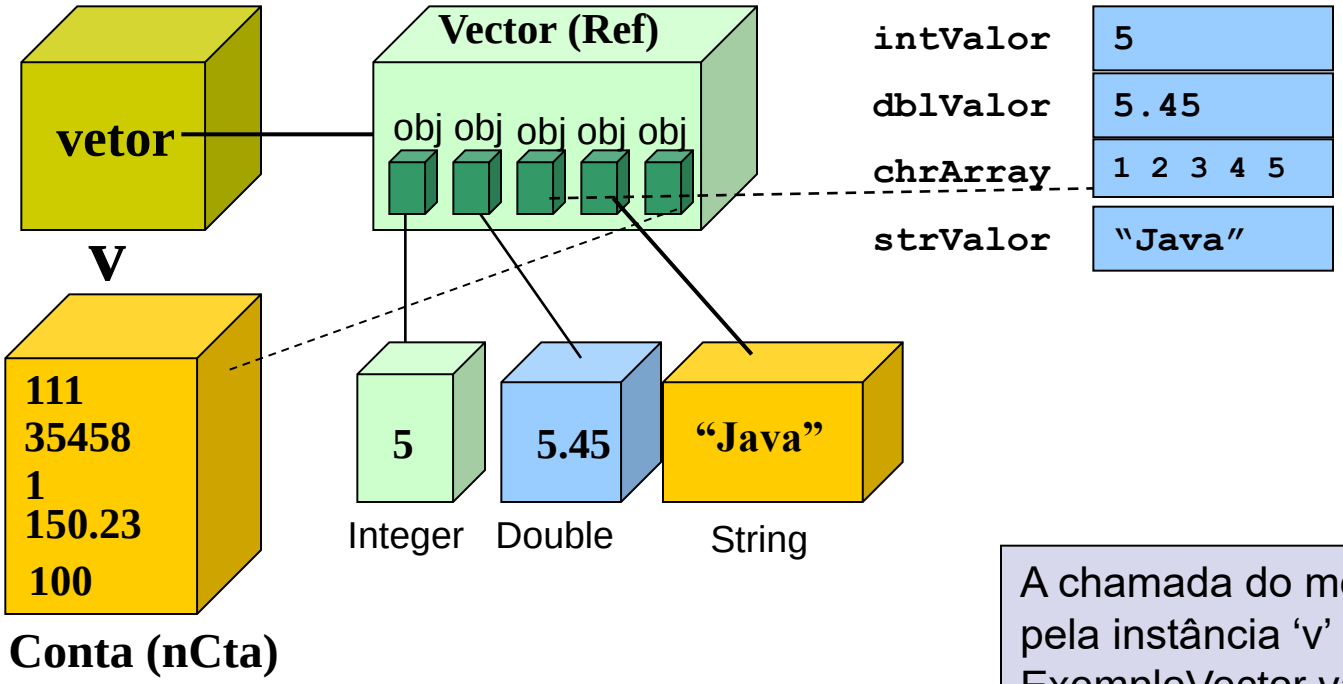
Classe Vector - funcionamento




```
//Adiciona elementos ao vetor
v.adicionaInt(intValor);
v.adicionaDouble(dblValor);
v.adicionaString(strValor);
v.adicionaCharArray(chrArray);
v.adicionaConta(nCta);
...
```

Cada comando v.adiciona*(...) é uma chamada de método da classe ExemploVector, em que cada método faz uma chamada ao método correspondente da classe Java Vector, convertendo os parâmetros de tipo básico para objetos de classes Java correspondentes (ex. convertendo **int** para objeto de *Integer*).

Classe Vector - funcionamento

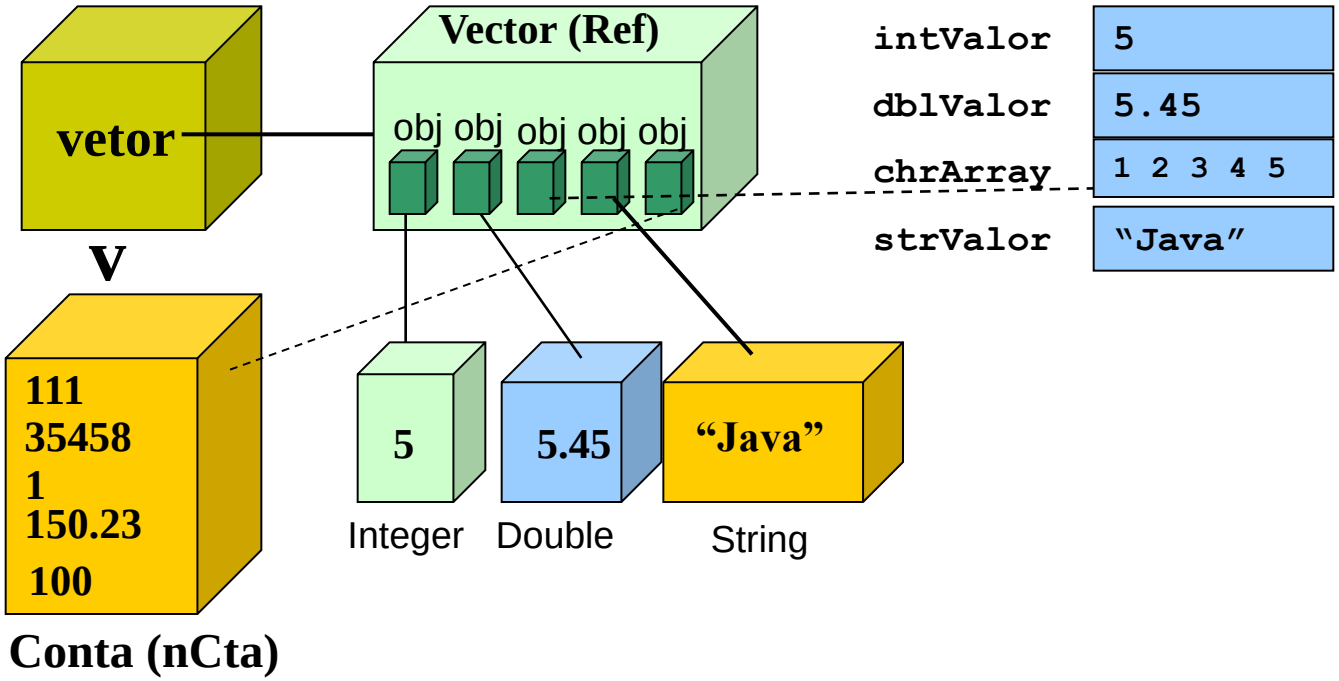



```
...
for (int i = 0; i < vetor.size(); i++) {
  obj = vetor.elementAt(i);
  if (obj instanceof char[]) {
    System.out.println(String.copyValueOf((char[]) obj));
  } else if (obj instanceof Conta) {
      Conta cta = (Conta)obj;
      cta.imprime();
    } else { System.out.println(obj.toString()); }
}
```

A chamada do método 'imprimeVetor()' pela instância 'v' da classe ExemploVector vai desencadear a varredura do atributo 'vetor'.

- Nesse processo de varredura será verificado que tipo de objeto está em cada posição do vetor
  - Vetor de caracteres
  - Objeto Conta
  - Outro objeto

Classe Vector - funcionamento

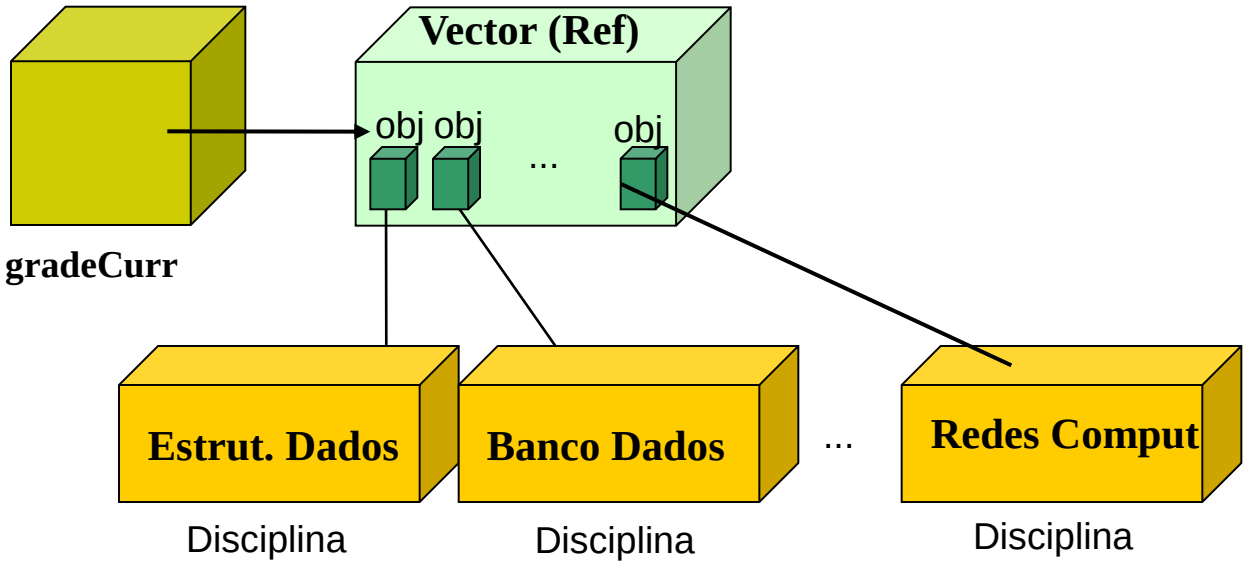


Este exemplo teve por finalidade apresentar as características da classe Vector, especialmente sua funcionalidade (métodos de inserção, posicionamento e recuperação de objetos) e conceitos importantes da programação OO:

- **Herança (implícita):** todos os objetos do programa valeram-se da herança que têm com a classe Object para poderem ser guardados em uma instância da classe Vector
- **Construtores.** As classes ExemploVector, Conta e Vector utilizaram métodos construtores para inicializarem suas instâncias (normalmente inicializando seus atributos)
- **Encapsulamento.** O vetor de objetos foi sempre acessado por métodos.

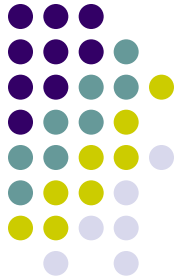
Classe Vector – Exercício
Entrega: 16/04

- Elabore um programa que guarde uma grade curricular de N disciplinas. O programa deve definir a classe Disciplina, a qual tem 3 atributos: código, nome e carga horária. Deve também definir a classe VetorDisciplina, responsável por manter um vetor com as disciplinas cadastradas. Deve ser possível inserir e remover disciplinas do vetor, bem como imprimir os dados do vetor, isto é, imprimir toda a grade curricular.



Exercício

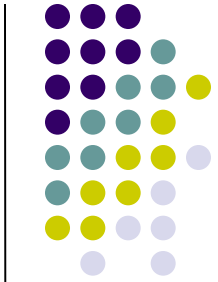
Sugere-se utilizar o código a seguir para gerar um menu



```
public void menu() {
  int escolha = 0;
  String escolhaInformada = "";
  int codigo = 0;
  String nome = "";
  int cargaHoraria = 0;
  do {
    do {
          escolhaInformada = JOptionPane.showInputDialog(
          "Escolha uma opção do menu:\n"+
          "[1] Adiciona disciplina\n"
          "[2] Remove disciplina\n"+
          "[3] Lista disciplinas\n"+
          "[4] Finaliza");
          escolha = Integer.parseInt(escolhaInformada);
} while ((escolha < 1) || (escolha > 4));
```

```
switch (escolha) {
      case 1:
         //Requisita o Código
          codigo = JOptionPane.showInputDialog ("Informe o código");
          //Requisita o Nome
          nome = JOptionPane.showInputDialog ("Informe o nome");
          //Requisita a Carga Horária
          cargaHoraria = Integer.parseInt(
                        JOptionPane.showInputDialog ("Informe o carga horária"));
           //Cria objeto disciplina
           objDisciplina.insereDisciplina(codigo, nome, cargaHoraria);
           //inserir objDisciplina no vetor
        break;
      case 2:
            //Código para remoção de uma disciplina
      case 3:
            //Código para impressão da grade curricular
      case 4:
        System.exit(0);
    }
  } while (true);
```

Créditos



- Os exemplos de código apresentados nessa aula foram elaborados por Roberto Pacheco da UFSC