



High performance. Delivered.



Application Delivery Fundamentals: Java

Module 4: Classes and Objects

[illegible]

“Classes and Objects”

-

[illegible]

- Copyright © 2023 Accenture All Rights Reserved. 4

The Customer Properties and Behaviors

- Properties
 - Name
 - Address
 - Age
 - Order number
 - Customer number
- Behaviors
 - Shop
 - Set address
 - Add item to cart
 - Ask for a discount
 - Display customer details



Atributos e Comportamento

- Variáveis declaradas como parte da classe representam os dados contidos pelas instâncias dessa classe;
- Os valores armazenados por essas variáveis representam os 'atributos' ou 'estados' atuais do objeto
- Os métodos declarados como parte da classe representam as operações que as instâncias da classe podem executar;
- Diz-se que essas operações são o 'comportamento' do objeto

214869676820706572666f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820

Identity

- Um objeto deve descrever os recursos que permitirão a seus usuários diferenciá-lo de outros objetos. Precisa ter uma identidade. Mesmo que dois objetos compartilhem os mesmos recursos, cada objeto tem uma identidade única.

Attributes

- Um objeto deve ser capaz de se descrever. Esse tipo de informação é armazenado nos atributos de um objeto que forma a estrutura do objeto..

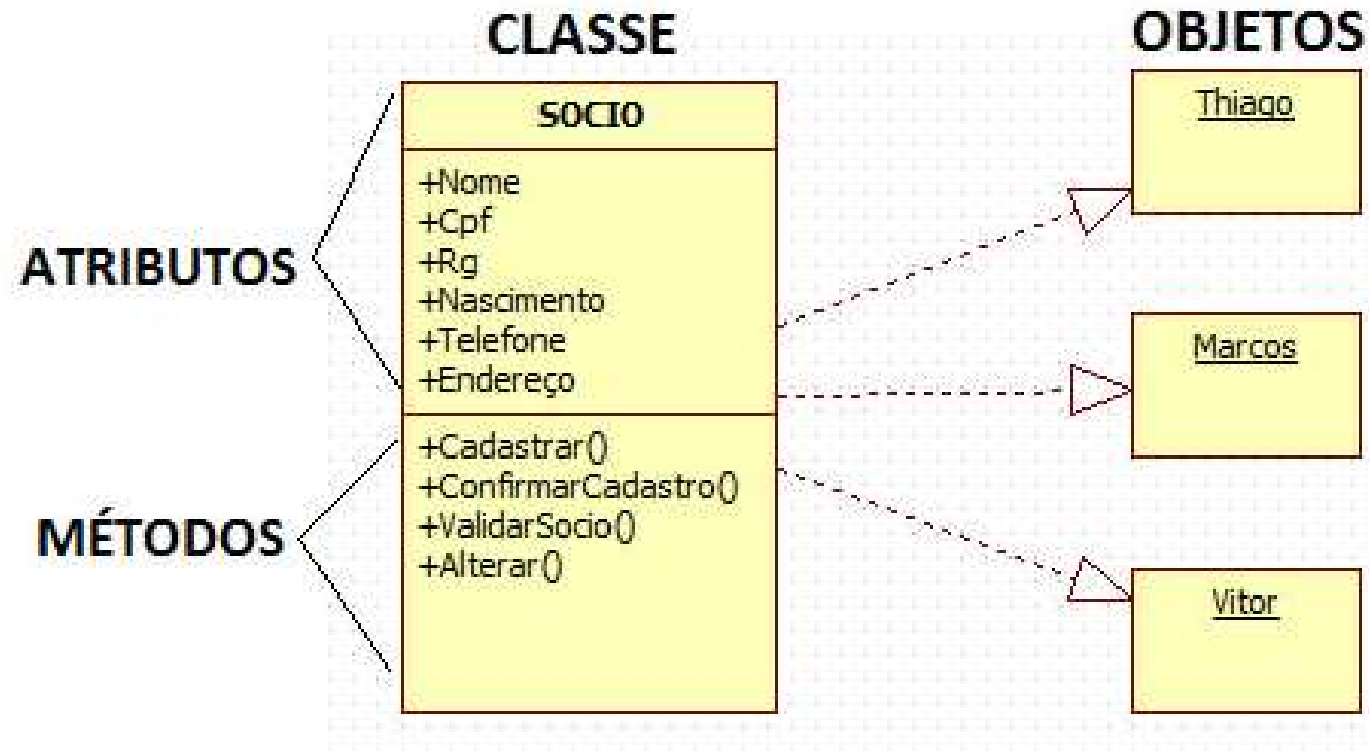
State

- Um objeto deve ser capaz de descrever sua condição atual, chamada estado. Às vezes, o estado do objeto é representado pelos valores de cada um de seus atributos.

Classes vs. Instances

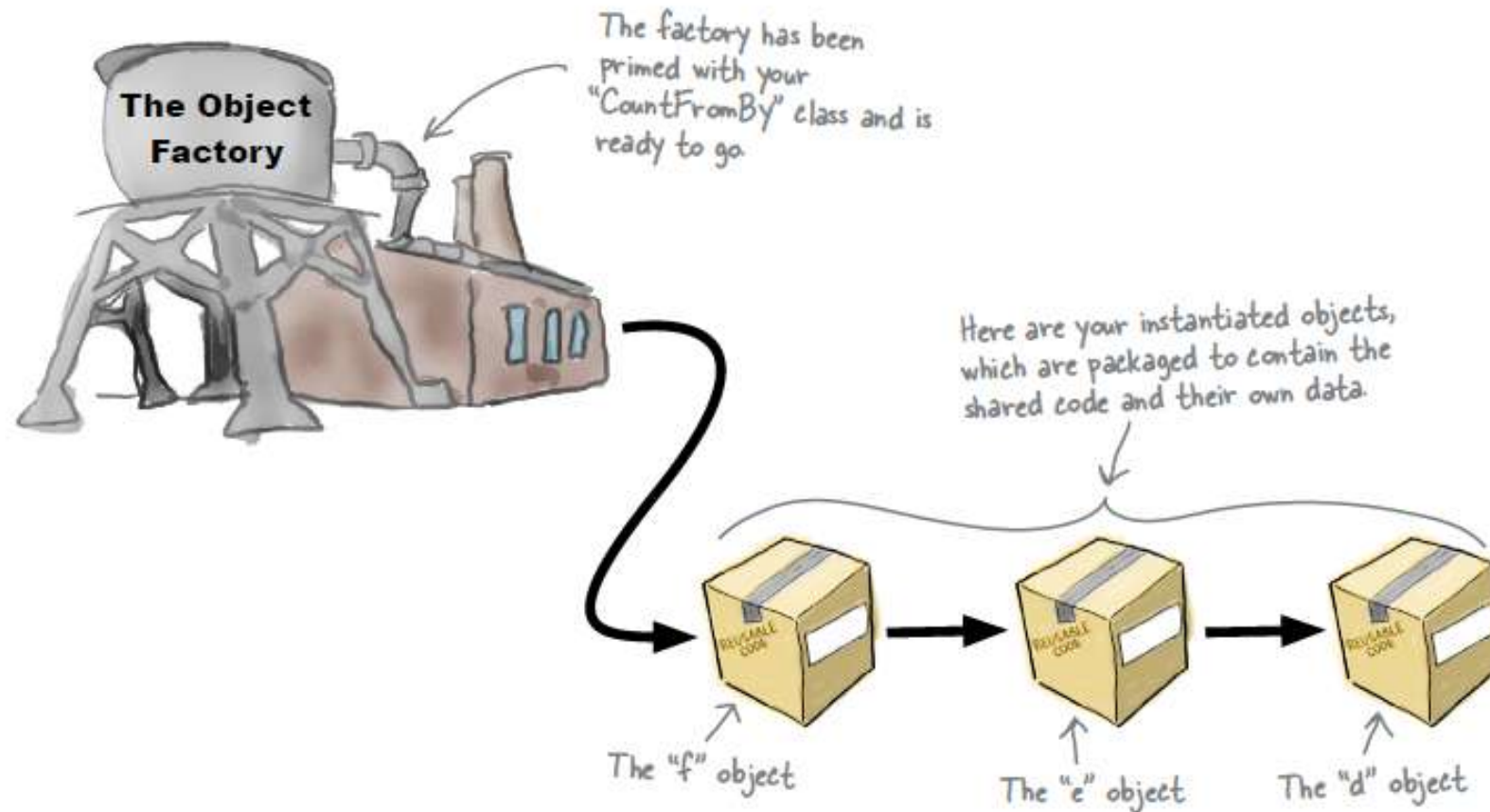
- Uma classe é um modelo para um objeto específico.
- Uma classe define os atributos e métodos que todos os objetos pertencentes à classe possuem.
- Os atributos e métodos de uma classe são chamados de "campos" ou "membros".
- Uma instância refere-se a um objeto que é membro de uma classe específica. Todos os objetos que pertencem a uma classe são instâncias dessa classe.

Classes vs. Instances





Classes vs. Instances



214869676820706572666f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820

Class Name = Substantivos Fields = Adjetivos Methods = Verbos

Copyright © 2023 Accenture All Rights Reserved.





Criando e Manipulando Instancias da Classe

- Instâncias de uma classe podem ser criadas usando a palavra-chave '**new**'. Esse processo é chamado de "instanciação".
- A palavra-chave '**new**' cria o objeto com base na classe especificada e retorna uma referência ao objeto recém-criado.
- A referência é então recebida por um identificador apropriado.

```
Person him = new Person("John Doe", 25, 'M');
```

Refer to the `ClassInstanceSample.java` sample code.

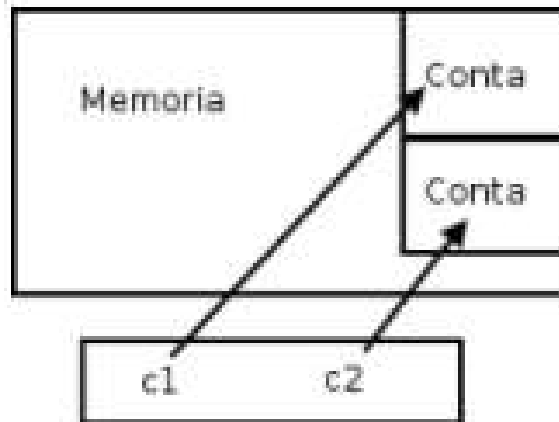
Criando e Manipulando Instancias da Classe (cont.)

- Os campos das instâncias de objetos podem ser acessados através dos nomes dos identificadores.
- O acesso aos campos e métodos dos membros é feito através do uso da notação 'ponto'.

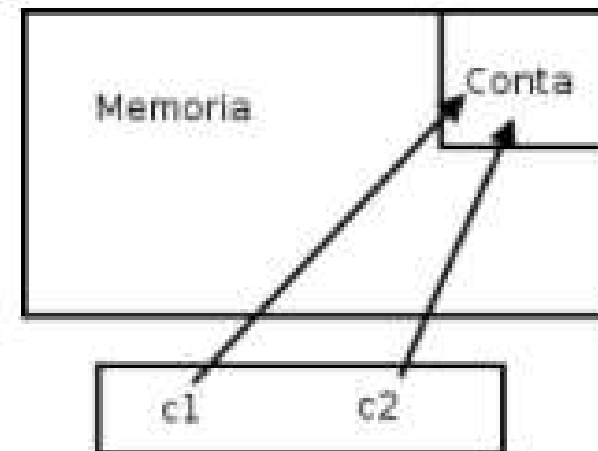
Refer to the `ClassInstanceSample.java` and `Person.java` sample code.

[illegible]

```
Conta c2;  
c2 = new Conta();
```



```
Conta c1 = new Conta();  
Conta c2 = c1;
```



- Copyright © 2023 Accenture All Rights Reserved. 18

- Copyright © 2023 Accenture All Rights Reserved. 19

Package Structure and the Class Path

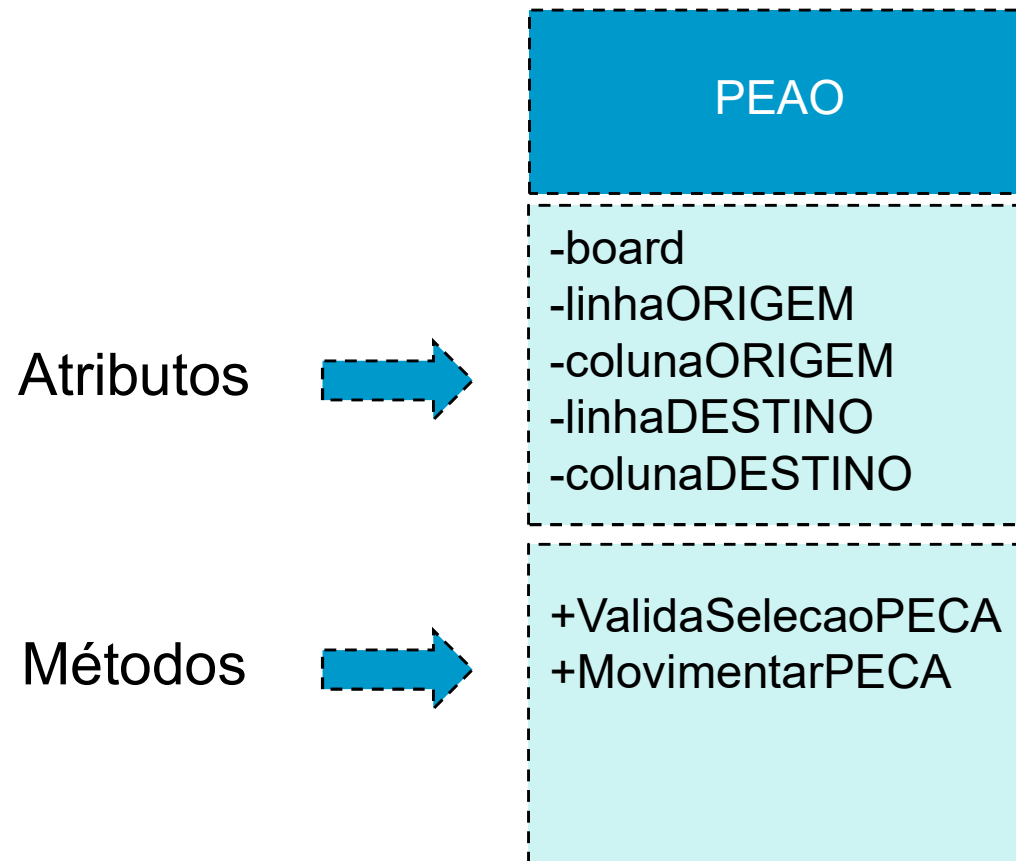
- Packages são coleções de classes.
- Packages são usados para agrupar classes com base na funcionalidade ou relacionamento.
- As classes podem ser atribuídas a um determinado pacote usando o método *package* keyword:

- A estrutura da package é representativa da estrutura de diretório em que a definição de classe está localizada.

214869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f48696768207

- ```
import <package_name>; OR import <class-name>;
import java.util.*;
import java.util.Calendar;
```

# Classes vs. Instances



- ```
public class Cube1{
    private int tamanho;

    Cube1(int tamanho)
    {
        this.tamanho = tamanho;
    }
}
```

Copyright © 2023 Accenture All Rights Reserved. 23

-
- A woman in a dark business suit is standing and writing on a whiteboard. She is holding a marker in her right hand and a folder in her left. Two men in business suits are seated at a glass table in the foreground, looking towards the whiteboard. The man on the left is holding a laptop, and the man on the right has his hand on his chin, appearing to be in deep thought. The room has large windows in the background, letting in bright light.

[illegible]

-
- A woman in a dark business suit is standing and writing on a whiteboard. She is holding a marker in her right hand and a folder in her left. Two men in business suits are seated at a glass table in the foreground, looking towards the whiteboard. The man on the left is holding a laptop, and the man on the right has his hand on his chin, appearing to be in deep thought. The room has large windows in the background, letting in bright light.

- **Atividade**

- Crie um Classe java Carro.java dentro do package sef.module4.activity
- Com os atributos: marca, CodigoCor, preço
- Com os métodos: constructor, Buzinar e Exibir;
- No método constructor inicialize os atributos;
- No método exibir exibe as atributos;
- No método Buzinar emitir som



[illegible]

- **Atividade**

- Crie um classe java CarroPrincipal.java dentro do package sef.module4.activity com o método main
- Instancie a classe Carro;
- Execute o método Exibir();
- Execute o método Buzinar();



[illegible]

- 



Garbage Collect (cont.)



- | Name | Running | Total |
|----------------------------------|-------------------|-----------|
| ThreadsWorker | 11.331 ms (64.5%) | 17.557 ms |
| Image Fetcher 1 | 0 ms (0%) | 4.542 ms |
| Image Fetcher 0 | 0 ms (0%) | 4.542 ms |
| Inactive RequestProcessor thread | 4.972 ms (28.3%) | 17.557 ms |
| Default RequestProcessor | 12.580 ms (71.7%) | 17.557 ms |
| pool-5-thread-1 | 0 ms (0%) | 17.557 ms |
| pool-4-thread-1 | 0 ms (0%) | 17.557 ms |
| Timer-0 | 5.761 ms (32.8%) | 17.557 ms |
| pool-3-thread-1 | 0 ms (0%) | 17.557 ms |
| Thread-4 | 0 ms (0%) | 17.557 ms |
| Default RequestProcessor | 17.557 ms (100%) | 17.557 ms |

214869676820706572666f726d616e63652e2044656c6976657265642e2f486967682070657266f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820706572666f726d616e63652e2044656c6976657265642e2f4869676820

- **JVisualVM**

```
Directory of C:\Program Files\Java\jdk1.8.0_181\bin
25/09/2018  09:22                197.464 jvisualvm.exe
                1 File(s)                197.464 bytes
```

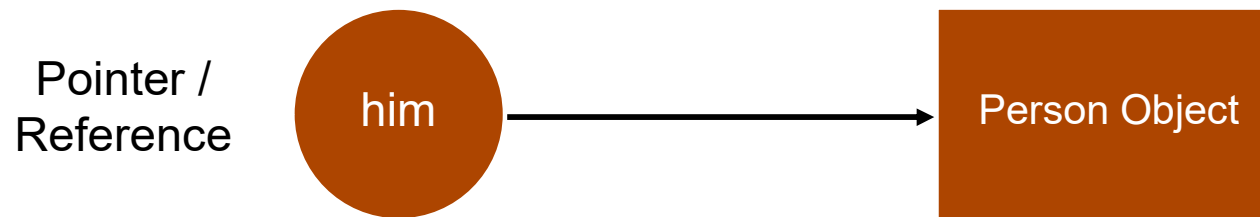


Refer to the jvisualvm.exe sample code.

Garbage Collect (cont.)

- Objetos estão seguros da garbage collect se houver pelo menos uma variável de referência que se refira ao objeto

```
Person him = new Person();
```

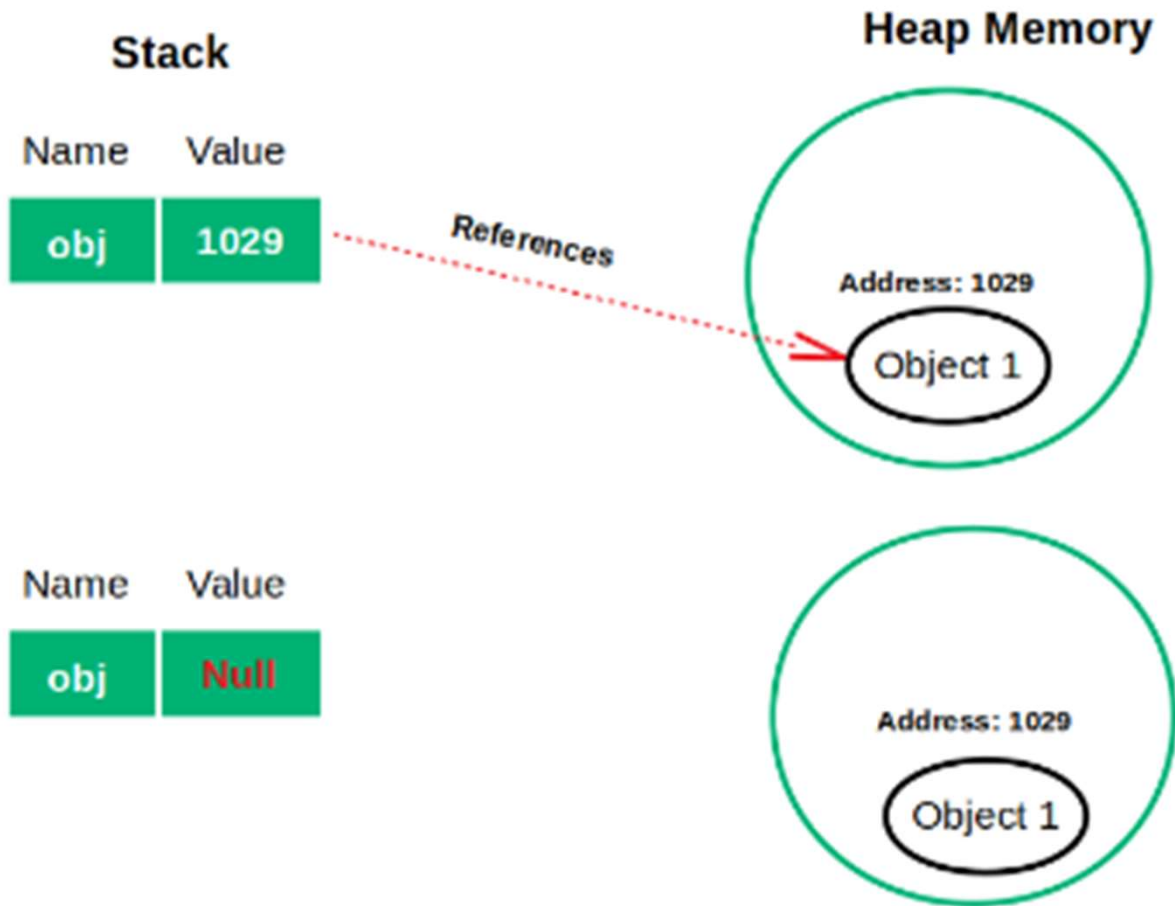


- Objetos são elegíveis para garbage collect quando não houver mais identificadores que se refiram ao objeto

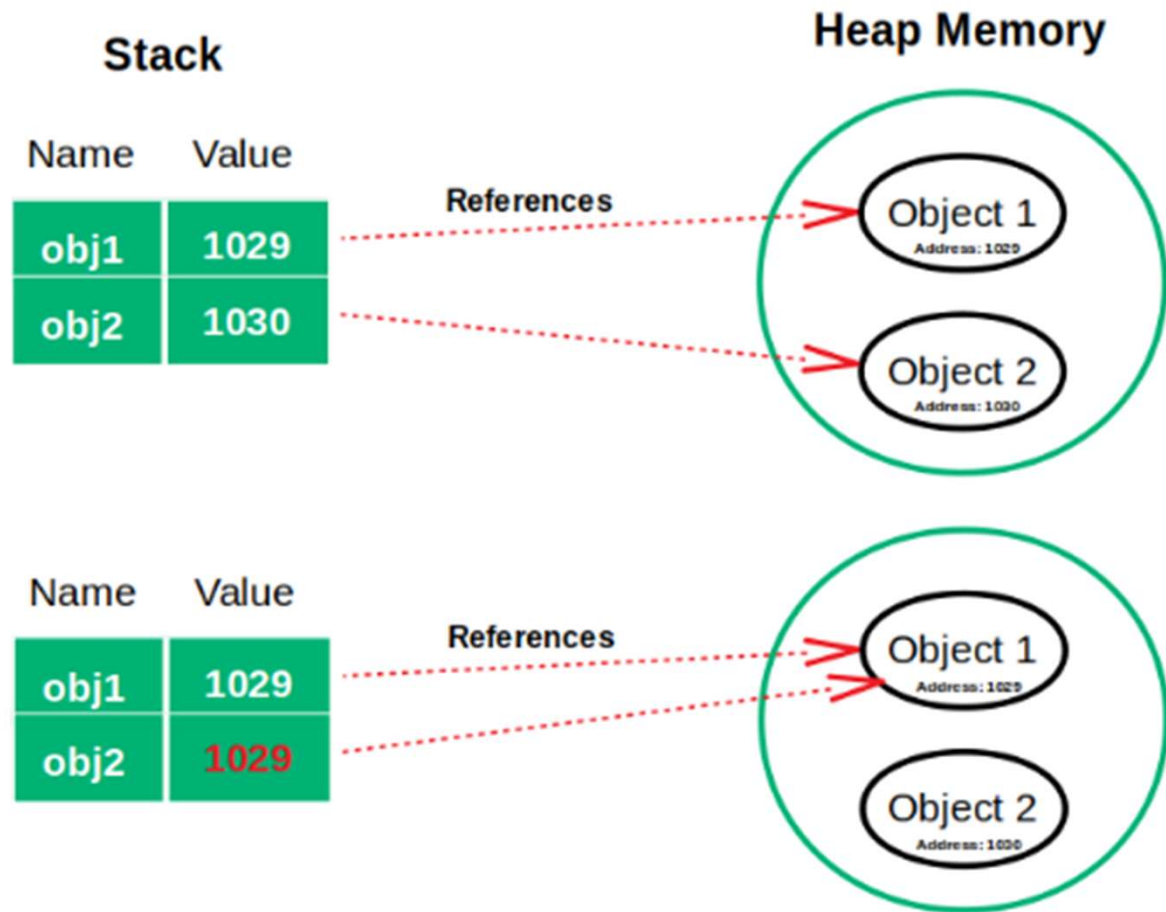
```
him = null;
```



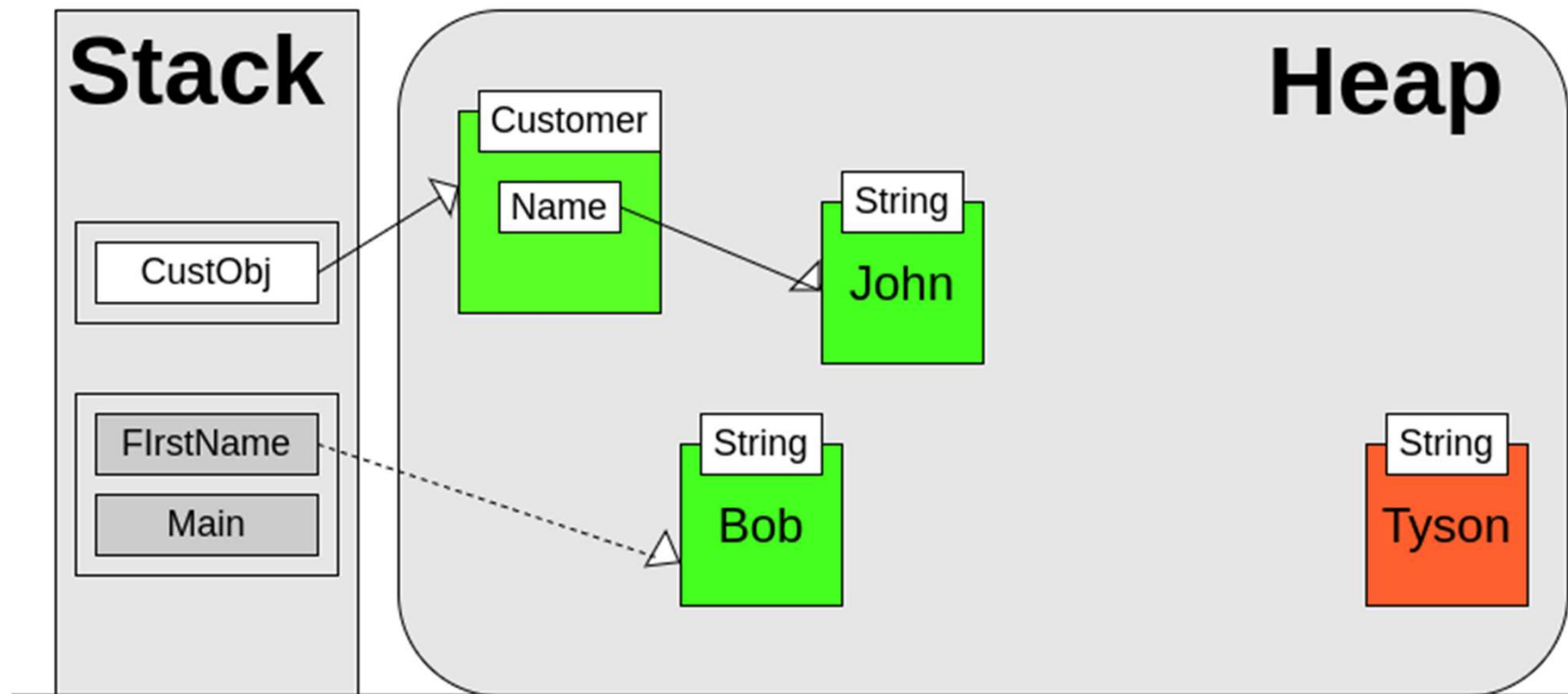
Garbage Collect (cont.)



Garbage Collect (cont.)



Garbage Collect (cont.)



[illegible]

É uma coleção de classes.

A coleta de lixo é o processo de limpar automaticamente a memória usada por objetos que não serão mais usados pelo programa.

- ```
protected void finalize() throws Throwable
{
 try
 {
 close(); // close open files
 }
 finally
 {
 super.finalize();
 }
}
```

- [illegible]

