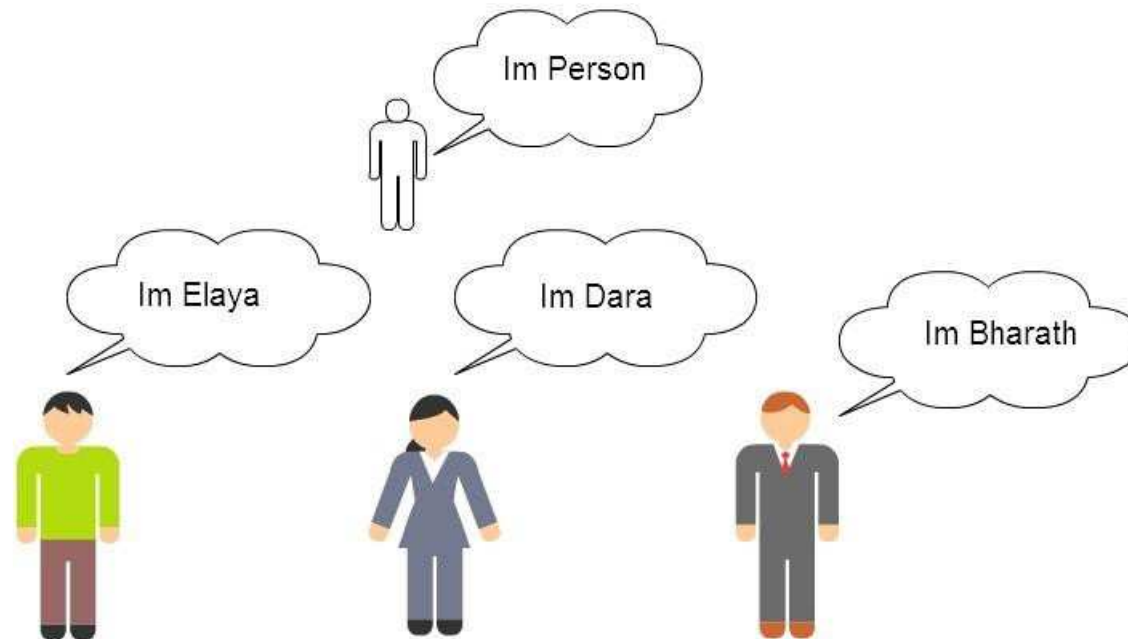


Bài 4

Lớp và đối tượng

Mục tiêu

- Giới thiệu Class và Object
- Khởi tạo đối tượng
- Constructors
- Quản lý bộ nhớ
- Đặc tả truy cập



Class And Objects

Giới thiệu

Class là gì?

- Là **Thành phần** chính xây dựng lên lập trình hướng đối tượng (OOP).
- Có cấu trúc luận lý (**logic**) định nghĩa lên **hình dạng** và **thuộc tính** của đối tượng.
- Định nghĩa **kiểu dữ liệu mới** (bên cạnh kiểu dữ liệu cơ bản có sẵn như: int, float, double...).
- **Thuộc tính** của lớp là các trường (field) thể hiện **trạng thái** của đối tượng.



Giới thiệu

Class bao gồm:

Fields

- Định nghĩa trạng thái đối tượng khi tạo từ class.
- Được gọi là biến.

Methods

- Thực thi hành vi của đối tượng.
- Được gọi là phương thức.

Giới thiệu

Cú pháp tạo **Class**:

```
public class Student {
```

← Khai báo tên lớp

```
String rollNumber; // Mã sinh viên  
String name; // Tên sinh viên  
int yearOfBirth; // Năm sinh
```

← Khai báo các trường

```
public void inputInfo() {
```

```
// Nhập thông tin sinh viên  
Scanner scanner = new Scanner(System.in);  
System.out.println("Nhập tên sinh viên");  
this.name = scanner.nextLine();  
}
```

← Khai báo các phương thức

```
public void outputInfo() {
```

```
System.out.println("Chào bạn " + this.name);  
}
```

```
}
```

Giới thiệu

Quy tắc tạo và đặt tên **Class**:

- Khai báo lớp cần phải bắt đầu bằng từ khóa "**class**" trước **tên** của lớp.
- Tên cần là **danh từ**.
- Tên lớp có thể hỗn hợp chữ hoa-thường, **viết hoa** chữ cái đầu tiên mỗi từ.
- Tên phải **đơn giản**, mang tính **mô tả** và có **ý nghĩa**.
- Không trùng **từ khóa** Java
- Tên không **bắt đầu là một số**.
- Tên không chứa ký tự đặc biệt, tuy nhiên vẫn có thể sử dụng ký tự dolla (\$) hoặc gạch dưới (_)

Giới thiệu

Khai báo và tạo **Object**:

Đối tượng (Object) được tạo bằng từ khóa **new**.

1. **JVM** cấp phát bộ nhớ cho đối tượng.
2. Trả về **tham chiếu** hoặc **địa chỉ ô nhớ** của đối tượng phân bổ.
3. Dữ liệu trả về trên được lưu trữ vào biến gọi là **biến tham chiếu**.

Giới thiệu □

Cú pháp:

Syntax

```
<class_name> <object_name> = new <class_name> ();
```

Ví dụ: Khai báo đối tượng objStudent:

```
Student objStudent = new Student ();
```

Lớp Student

Đối tượng đặt
tên là
objStudent

Từ khóa

Hàm khởi tạo
(constructor)

Giới thiệu



Class

```
public class Book{  
    String name;  
    String author;  
}
```

1. Khai báo
lớp



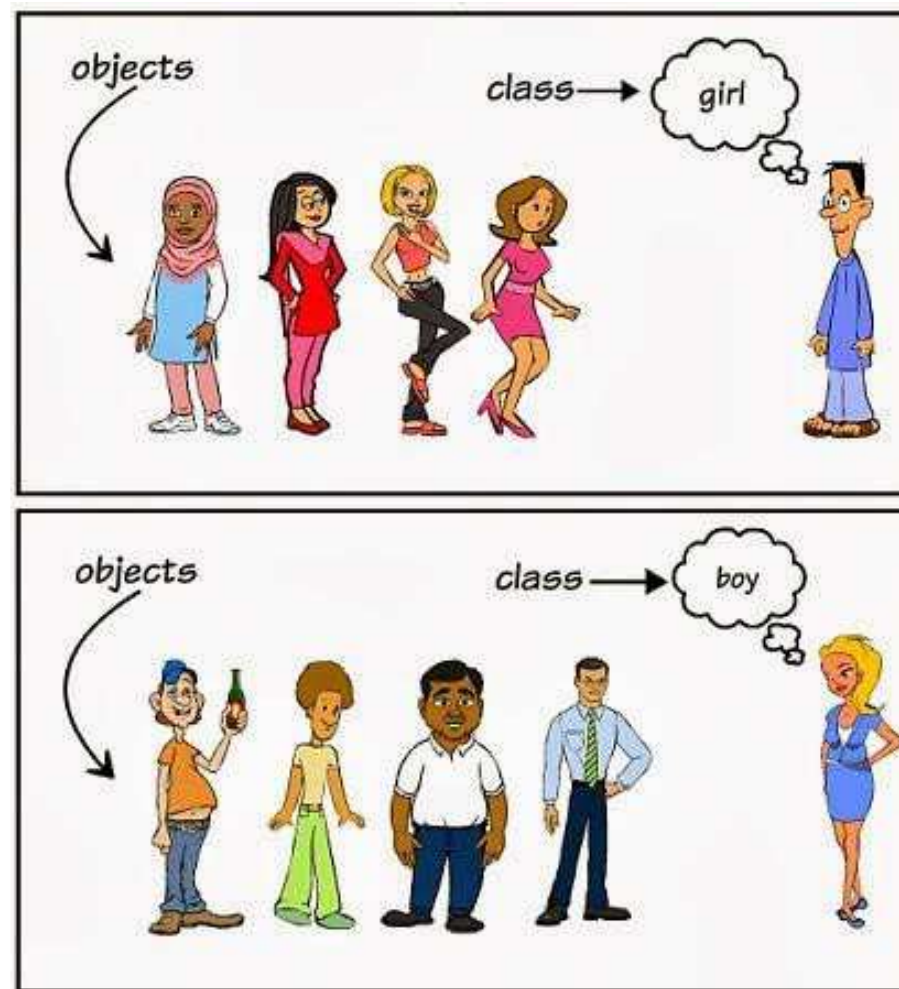
Object

```
Book objBook = new Book();
```

2. Khởi tạo đối
tượng

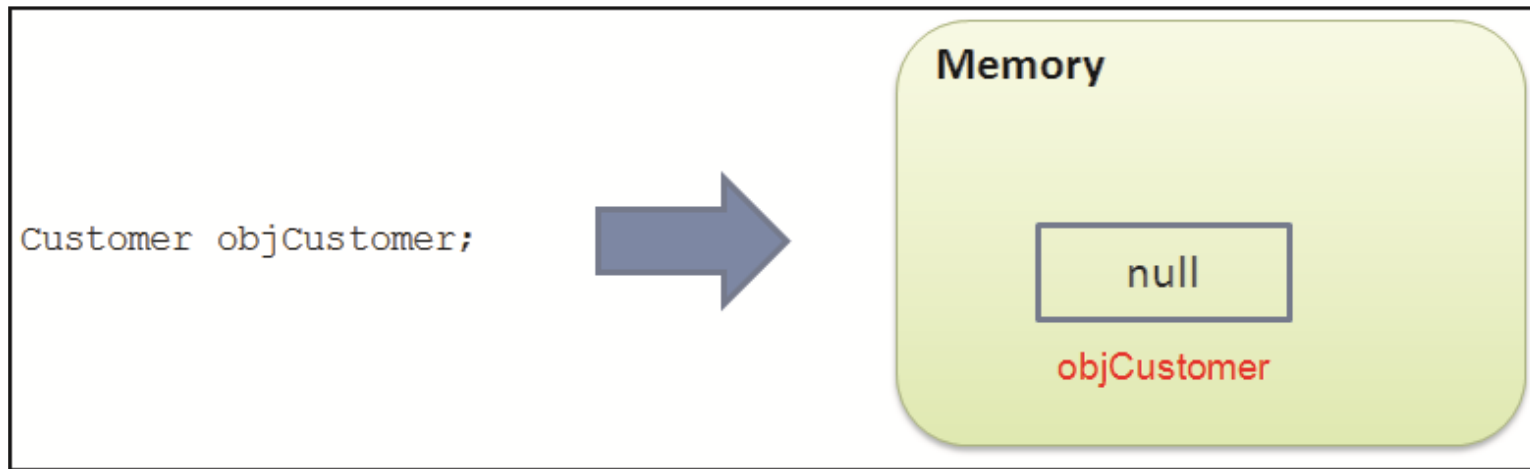


Giới thiệu

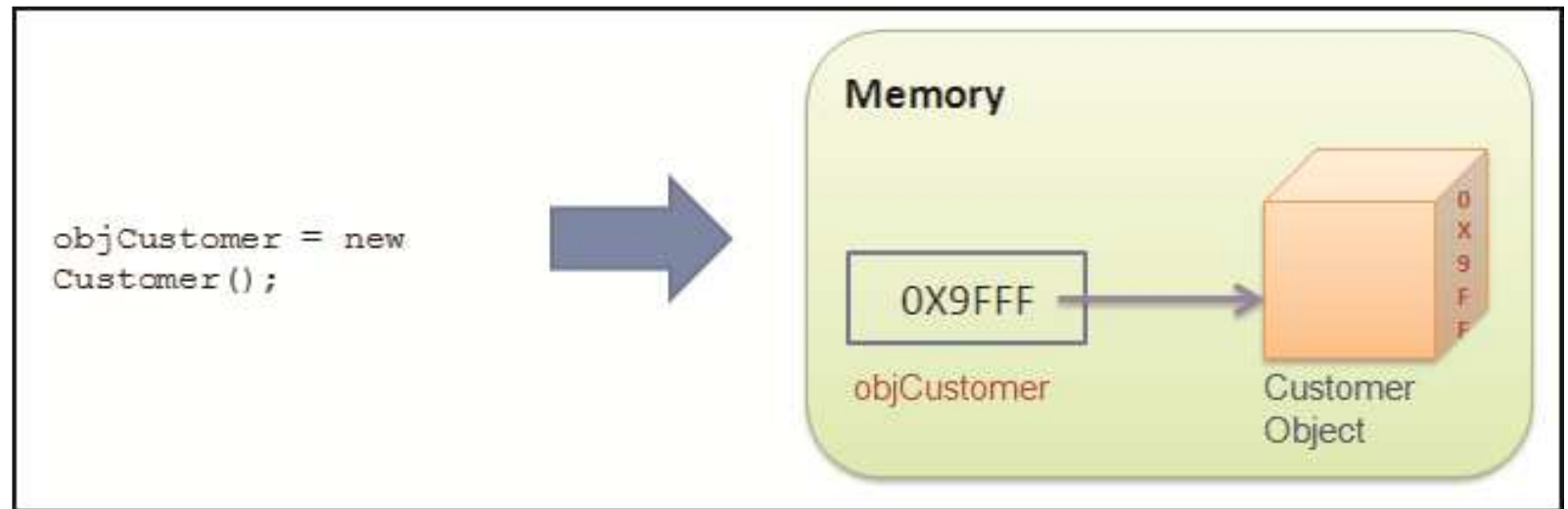


Giới thiệu

Quy trình tạo **Object** có 2 giai đoạn:

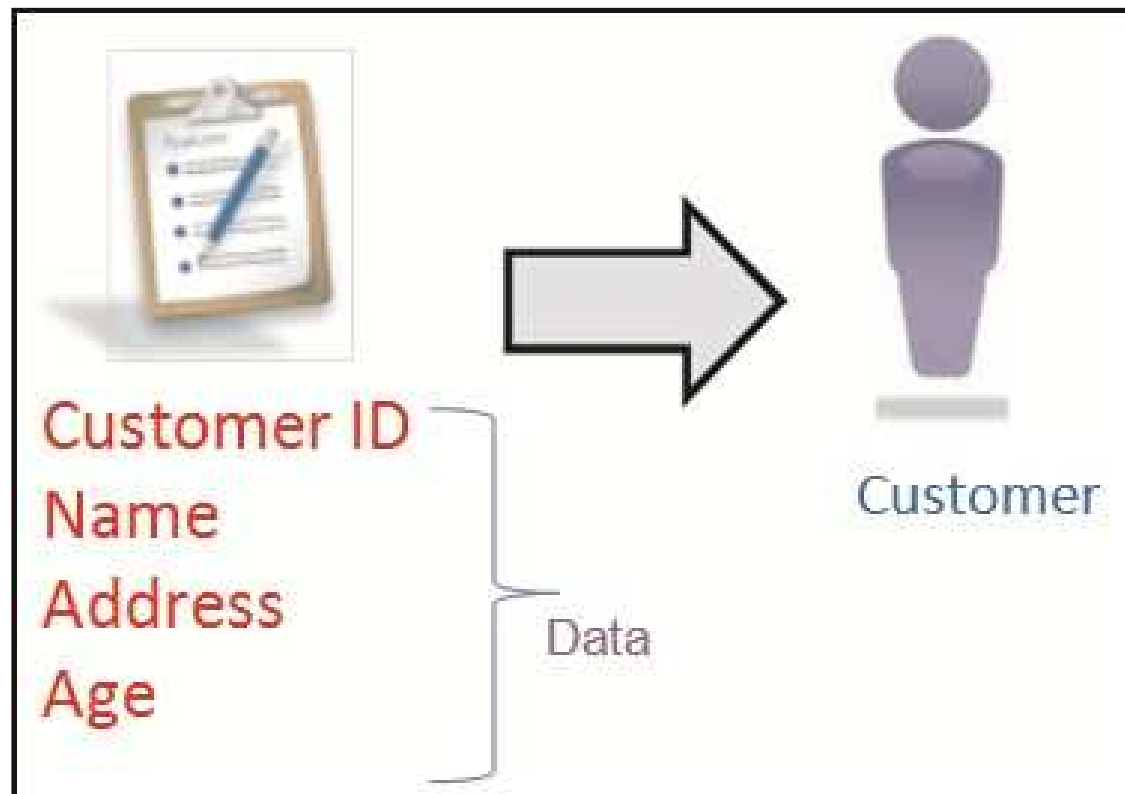


2



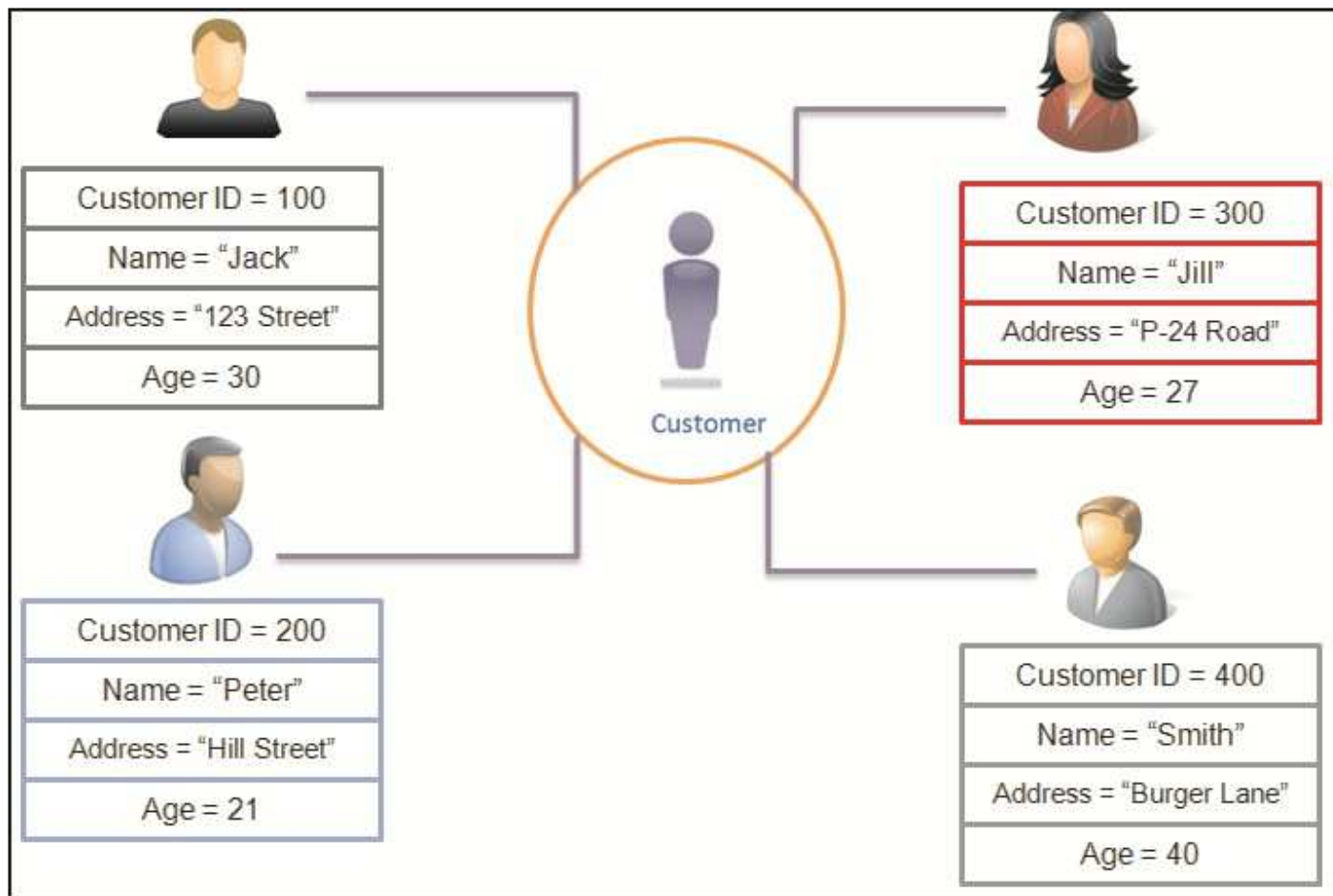
Biến của Class

Biến của **Class** sử dụng để **lưu trữ dữ liệu**.



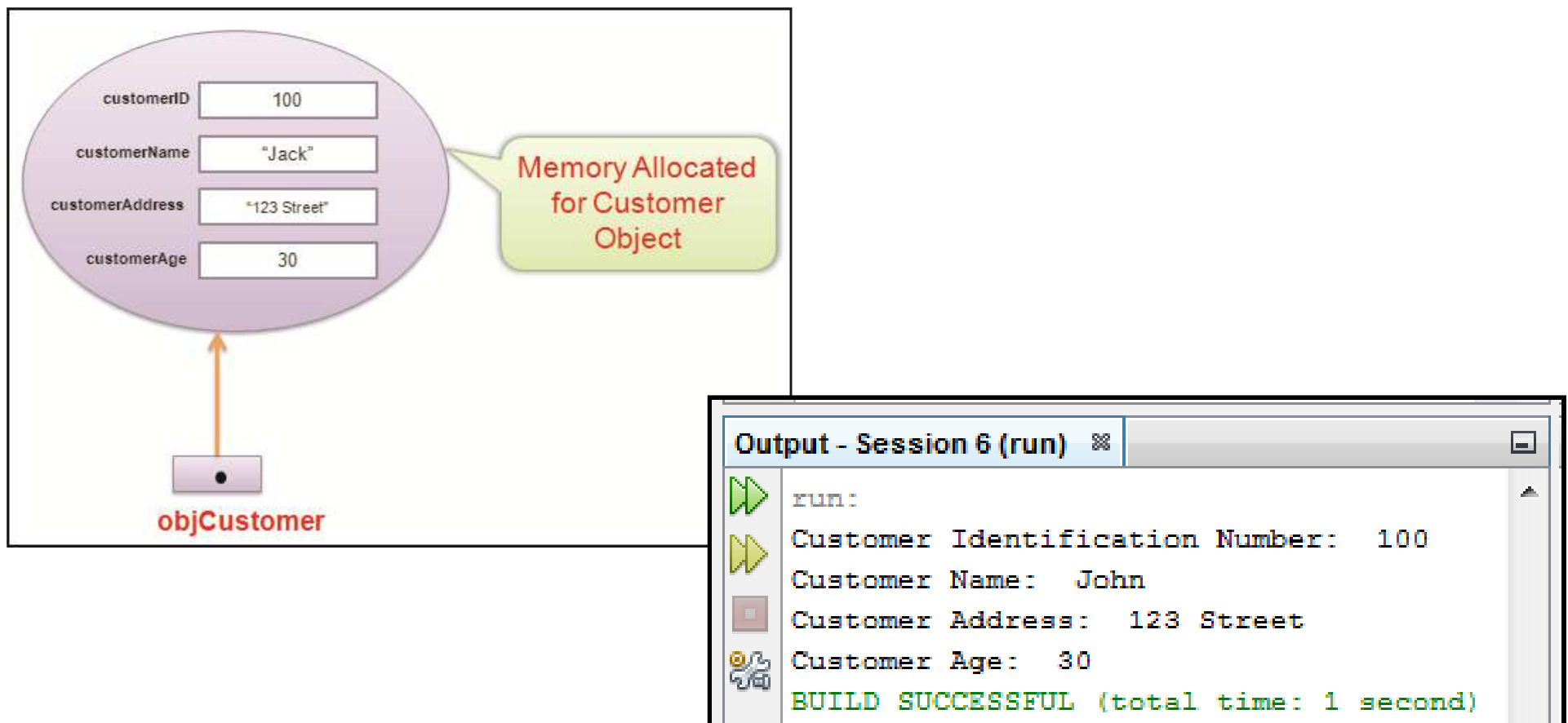
Biến của Class

Mỗi đối tượng được **tạo mới** từ **class** sẽ có một **bản sao riêng biệt** biến lưu trữ dữ liệu này.



Biến của Class

Dữ liệu **biến** của **Class** được cấp phát bộ nhớ để lưu trữ dữ liệu:



Phương thức của Class

Phương thức của **Class** là:

- Mô tả **chức năng**
- Thực thi **hành vi** của đối tượng
- **Tương tác, tính toán...** biến.
- Có thể truy cập từ đối tượng qua toán tử "." – chấm.

Quy tắc đặt tên phương thức:

- Không trùng từ khóa Java.
- Không chứa khoảng trắng.
- Không bắt đầu là số.
- Có thể bắt đầu là ký tự, gạch dưới, hoặc ký tự '\$'.
- Nên là một động từ viết thường.
- Nên có ý nghĩa miêu tả cụ thể.
- Viết tên theo quy tắc camel-case.

Phương thức của Class

Cú pháp của **Phương thức**:

Syntax

```
[access_modifier] <return_type> <method_name> ([list  
of parameters]) {  
  
// Body of the method  
  
}
```


Phương thức của Class

Cú pháp của **Phương thức**:

```
public class Student {
```

```
    String rollNumber; // Mã sinh viên
```

```
    String name; // Tên sinh viên
```

```
    int yearOfBirth; // Năm sinh
```

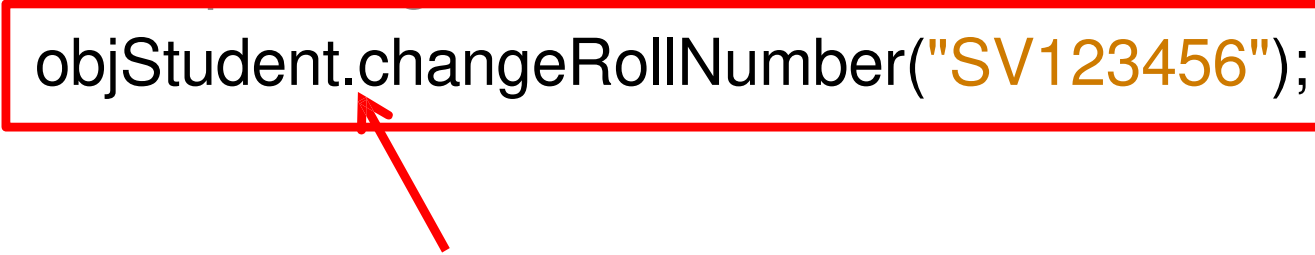
```
    void changeRollNumber(String newRollnumber) {  
        rollNumber = newRollnumber;  
    }
```

```
}
```

Phương thức của Class

Gọi **Phương thức** thông qua toán tử “.”:

```
public class StudentManager {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        Student objStudent = new Student();  
        objStudent.rollNumber = "SV123";  
        // Gọi phương thức  
        objStudent.changeRollNumber("SV123456");  
    }  
}
```



Constructor

Constructor?

- Là **phương thức** có tên trùng với tên class.
- Đóng vai trò **khởi tạo** giá trị các biến của lớp hoặc thực hiện các hoạt động khởi động (**chỉ 1 lần duy nhất** khi đối tượng được tạo).
- **Tự động** kích hoạt.
- Có hoặc không có **tham số** và không có kiểu trả về **return**.

```
class <ClassName>
{
    <ClassName> () ← Constructor
    {
        // Initialization code
    }
}
```

Constructor

Constructor **không tham số** = Constructor **mặc định**.

```
public class Student {  
    String rollNumber; // Mã sinh viên  
    String name; // Tên sinh viên  
    int yearOfBirth; // Năm sinh
```

```
    public Student() {  
    }  
}
```

Constructor

Constructor **có tham số**.

```
public class Student {
```

```
    String rollNumber; // Mã sinh viên
```

```
    String name; // Tên sinh viên
```

```
    int yearOfBirth; // Năm sinh
```

```
    public Student(String rollNumber, String name, int yearOfBirth) {  
        this.rollNumber = rollNumber;  
        this.name = name;  
        this.yearOfBirth = yearOfBirth;  
    }
```

```
}
```

Quản lý bộ nhớ

Bộ nhớ trong Java gồm 2 thành phần cụ thể là:

Stack

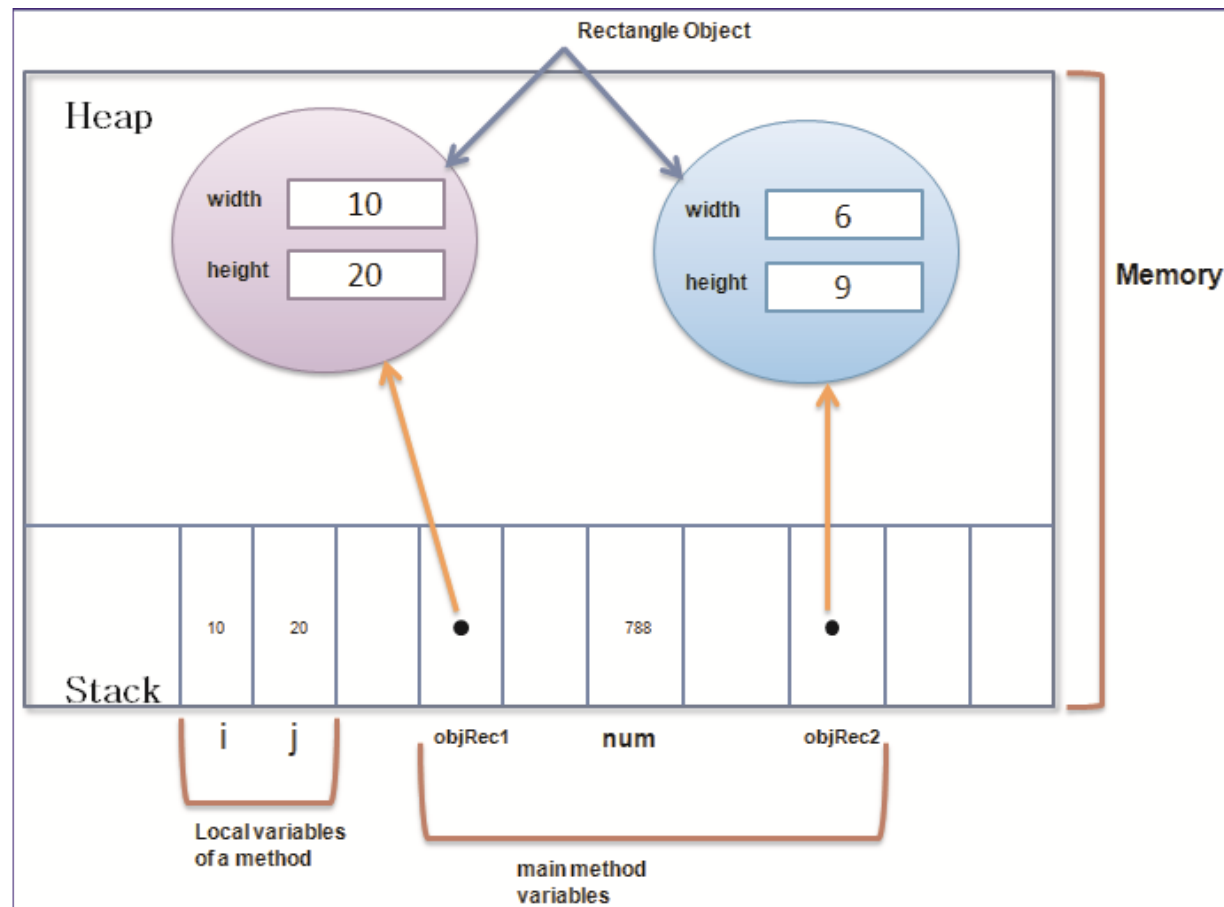
- Là vùng bộ nhớ lưu trữ tham chiếu đối tượng và thông tin phương thức.
- Lưu trữ tham số của phương thức và biến cục bộ.

Heap

- Là vùng bộ nhớ được cấp phát động.
- Trong Java, đối tượng được phân bổ không gian bộ nhớ vật lý trên heap theo thời gian thực, có nghĩa là bất cứ khi nào JVM thực hiện toán tử **new**.

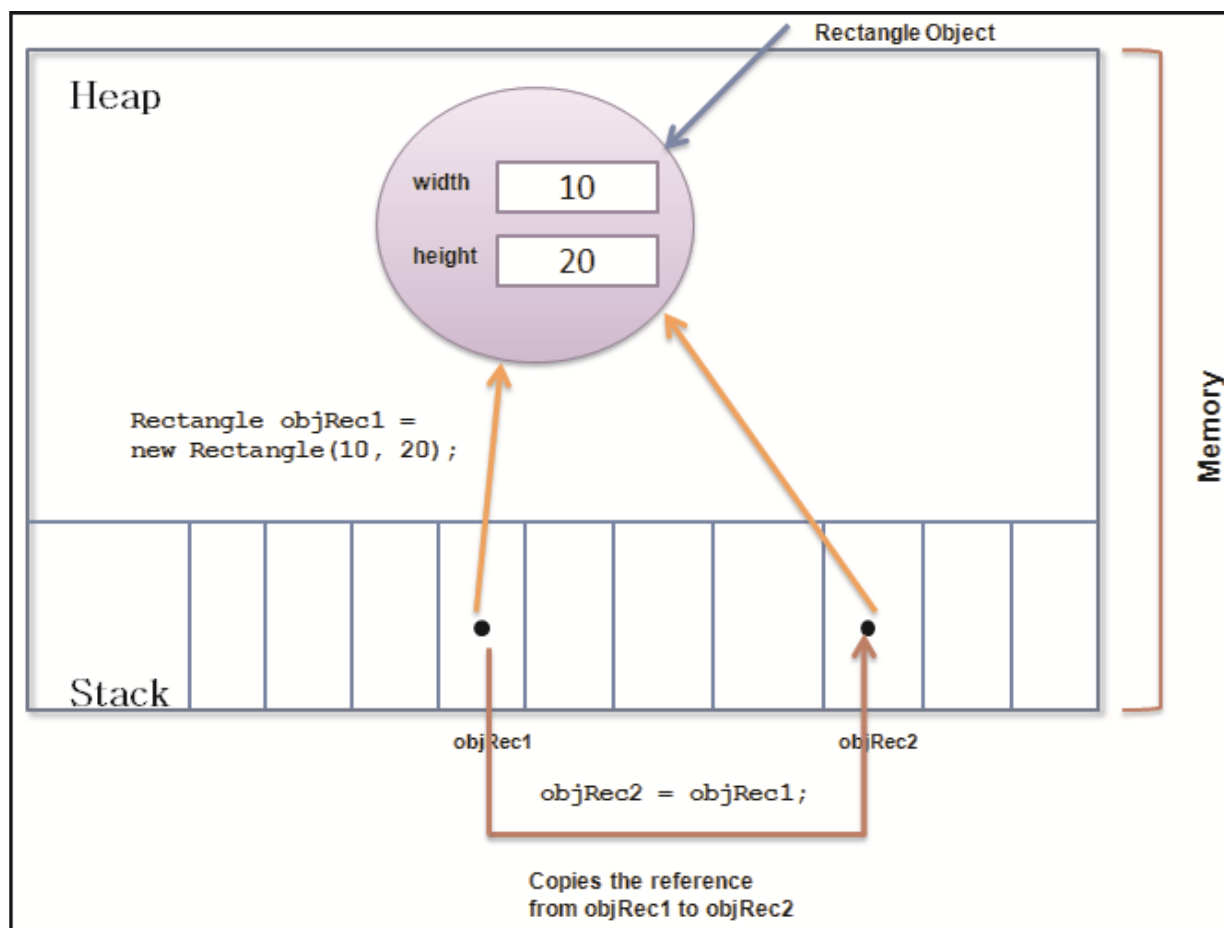
Quản lý bộ nhớ

Bộ nhớ phân bổ lưu trữ:



Quản lý bộ nhớ

Bộ nhớ phân bổ lưu trữ khi gán **object** cho **object**: chỉ có địa chỉ được gán.



Bổ từ truy cập

Đặc tả truy cập quyết định phạm vi truy cập với Class tương tự như với biến và phương thức. Có 4 đặc tả:

public

- Có phạm vi truy cập rộng nhất ở bất cứ đâu.

protected

- Phạm vi truy cập giới hạn trong cùng package.

package (default)

- Tương tự protected.
- Là khai báo mặc định

private

- Phạm vi truy cập chỉ giới hạn trong class.

Tóm tắt bài học

- ✓ **Class** là cấu trúc logic định nghĩa hình dạng, tính chất của đối tượng.
- ✓ **Object** là cá thể thực sự của class, nó được tạo bởi từ khóa new. Toán tử new yêu cầu JVM cấp phát bộ nhớ cho đối tượng.
- ✓ Class bao gồm **field** và **method**.
- ✓ **Constructors** là phương thức khởi tạo cho đối tượng từ class.
- ✓ Java quản lý vùng lưu trữ dữ liệu với 2 thành phần là **stack** và **heap**.
- ✓ Có 4 bộ từ truy cập: **public** > **protected** > **default** > **private**