

BÀI 1
GIỚI THIỆU
VỀ LẬP TRÌNH HƯỚNG
ĐỐI TƯỢNG VÀ NGÔN NGỮ JAVA

TÌNH HUỐNG DẪN NHẬP

Anh Hùng chủ cửa hàng bán xe Hùng Phát muốn xây dựng hệ thống phần mềm để theo dõi việc buôn bán xe thuận tiện hơn. Qua kinh nghiệm quản lý, anh đưa ra một số thông tin sau:

- Những yếu tố liên quan đến hoạt động bán xe: Các loại xe, nhân viên bán hàng, Khách hàng;
- Các hoạt động liên quan đến việc mua bán:
 - Nhân viên bán hàng đưa khách hàng xem mẫu xe;
 - Khách hàng lựa chọn loại xe;
 - Khách hàng đặt hóa đơn;
 - Khách hàng trả tiền;
 - Xe được trao cho khách hàng;



Vì các đại lý của anh Hùng ở các vị trí khác nhau sử dụng các trang thiết bị khác nhau, nên một yêu cầu khác là làm sao để hệ thống quản lý có thể sử dụng được trên nhiều loại máy.



Vậy theo Anh/Chị những thông tin anh Hùng quan tâm cần được biểu diễn thế nào trong máy tính?

Nên sử dụng công nghệ gì để hệ thống quản lý bán xe có thể hoạt động được trên nhiều máy tính khác nhau?

MỤC TIÊU

Mô tả về ngôn ngữ Java.

Trình bày được các khái niệm cơ bản trong lập trình hướng đối tượng.

Trình bày được các khái niệm về đối tượng và lớp trong Java.

Xây dựng được ít nhất 01 chương trình Java có hoạt động đơn giản.

Cài đặt và sử dụng được một số tính năng đơn giản của ít nhất 01 môi trường phát triển ứng dụng (IDE) – BlueJ.

NỘI DUNG

1

Giới thiệu về Java.

2

Giới thiệu về lập trình hướng đối tượng trong Java.

3

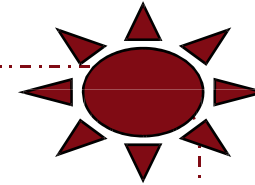
Mô tả lớp và tạo đối tượng trong Java.

4

Lập trình ứng dụng Console với Java.

1.1. GIỚI THIỆU VỀ JAVA

1. Java là gì?
2. Quá trình phát triển;
3. Công nghệ Java;
4. Các đặc trưng cơ bản của Java;
5. Cài đặt môi trường và thực thi ứng dụng Java.



1.1.1. JAVA LÀ GÌ ?

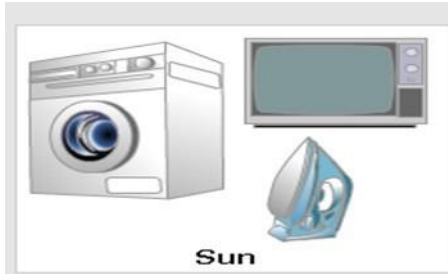


James Gosling



- Java là một công nghệ giúp phát triển các ứng dụng có khả năng thực thi trên nhiều môi trường, nền tảng khác nhau.
- Được khởi đầu bởi *James Gosling* và đồng nghiệp tại Sun Microsystem năm 1991.
- Phát hành phiên bản đầu tiên vào năm 1994.

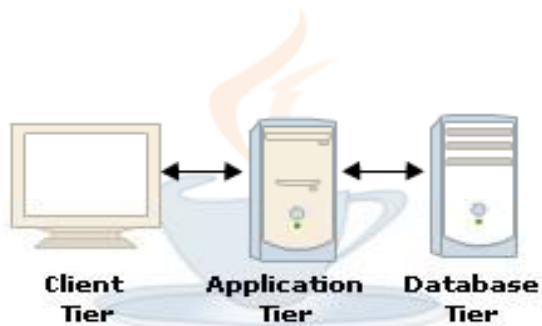
1.1.2. QUÁ TRÌNH PHÁT TRIỂN



**Embedded Systems
(1991 – 1994)**



**Java
A client – side Wonder
(1995 – 1997)**



**Moved into the Middle – tier
(1997 – to present)**



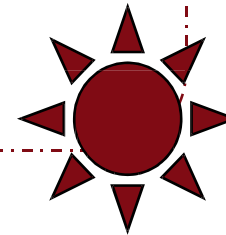
**Future:
May gain more success**

1.1.3. CÔNG NGHỆ JAVA

- Môi trường thực thi và triển khai:
 - Java Virtual Machine: Máy ảo Java, môi trường thực thi của các ứng dụng Java;
 - Java API: Bộ giao diện lập trình ứng dụng cơ bản.
- Ngôn ngữ Java: Ngôn ngữ đi kèm với nền tảng Java giúp lập trình viên phát triển các ứng dụng chạy trên JVM.
- Bộ công cụ phát triển ứng dụng (JDK).

1.1.4. CÁC ĐẶC TRƯNG CƠ BẢN CỦA JAVA

- Ngôn ngữ lập trình hướng đối tượng.
- Nền tảng độc lập.
- An toàn.
- Khả chuyển.
- Hỗ trợ xử lý đa luồng.



1.1.5. CÀI ĐẶT MÔI TRƯỜNG VÀ THỰC THI ỨNG DỤNG JAVA

- Cài đặt môi trường:
 - Môi trường thực thi: Java Runtime Environment;
 - Môi trường và công cụ phát triển: Java Development Kit (JDK).
- Cấu hình biến môi trường:
 - CLASSPATH: Đường dẫn tới các lớp, thư viện trong ứng dụng Java;
 - PATH: Đường dẫn tới thư mục chứa các công cụ của Java (javac, java...).
- Demo thực thi các ứng dụng Java (Desktop Application, Applet, Midlet, Web Application...).

CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 3 ▾

Điểm: 10

Một chương trình Java có thể chạy trên bất kỳ hệ điều hành nào?

- ☐ A. Đúng.
- ☐ B. Sai.

PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

CÂU HỎI THẢO LUẬN

Qua tìm hiểu về đặc điểm của công nghệ java, anh/chị có thể cho tôi biết để xây dựng hệ thống quản lý xe của cửa hàng anh Hùng chúng ta có thể sử dụng công nghệ java được hay không? Vì sao?

1.2. GIỚI THIỆU VỀ LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG TRONG JAVA

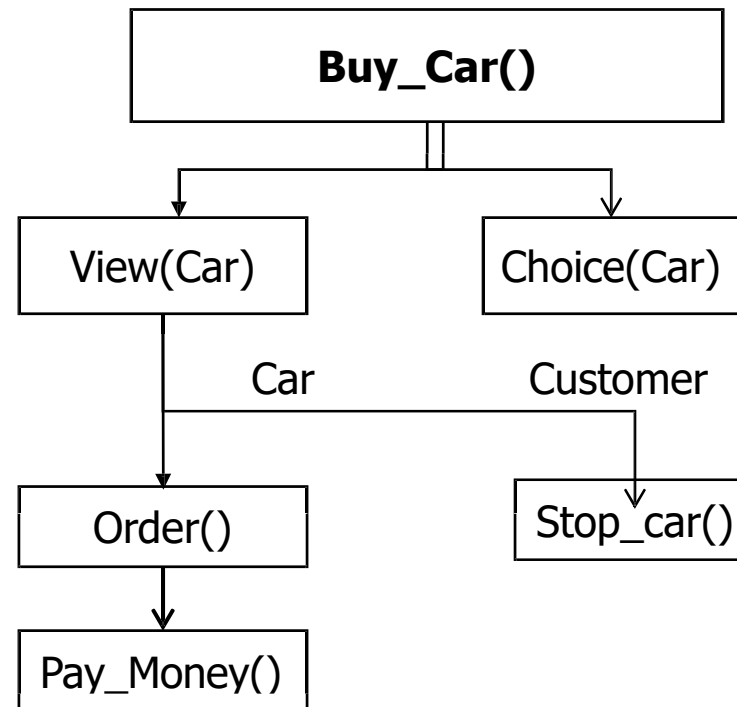
1. Lập trình hướng đối tượng là gì?
2. So sánh hướng đối tượng và hướng cấu trúc;
3. Các khái niệm cơ bản trong lập trình hướng đối tượng.

1.2.1. LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG LÀ GÌ?

- Lập trình hướng đối tượng là kỹ thuật lập trình cho phép tạo nên các ứng dụng dựa trên sự giao tiếp giữa các đối tượng phần mềm.
- Có sự tương xứng giữa các đối tượng trong chương trình và các đối tượng trên thực tế.
- Đối tượng phần mềm thông thường là sự kết hợp giữa mã lệnh và dữ liệu trong một thể thống nhất.

1.2.2. SO SÁNH HƯỚNG ĐỐI TƯỢNG VÀ HƯỚNG CẤU TRÚC

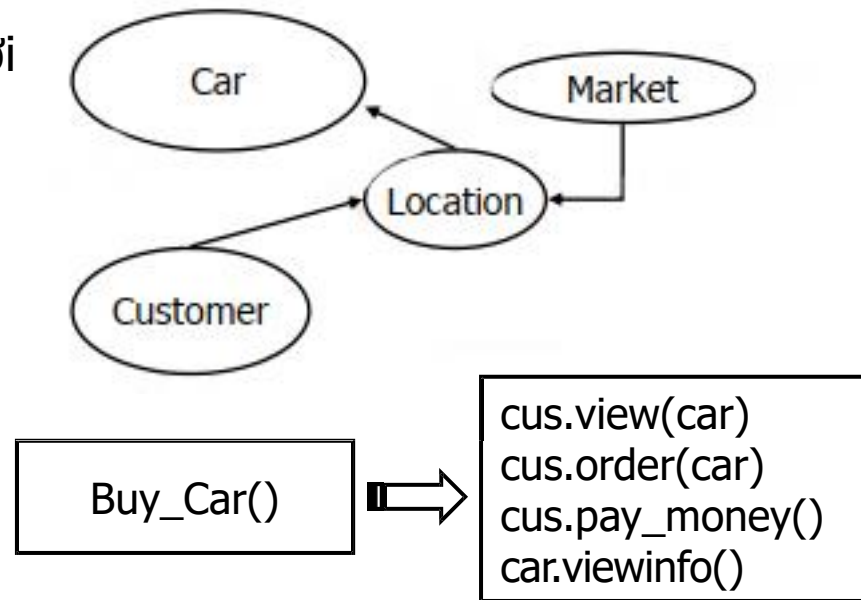
- Lập trình hướng cấu trúc:
 - Hướng tiếp cận: Từ trên xuống (Top Down);
 - Chia nhỏ bài toán thành các Module chức năng;
 - Dữ liệu và mã lệnh phân tán.
- Hạn chế:
 - Tính ổn định giảm khi hệ thống phát triển;
 - Khó bảo trì và tái sử dụng;
 - Chi phí phát triển cao.



Sơ đồ mô tả hành động mua xe của khách hàng

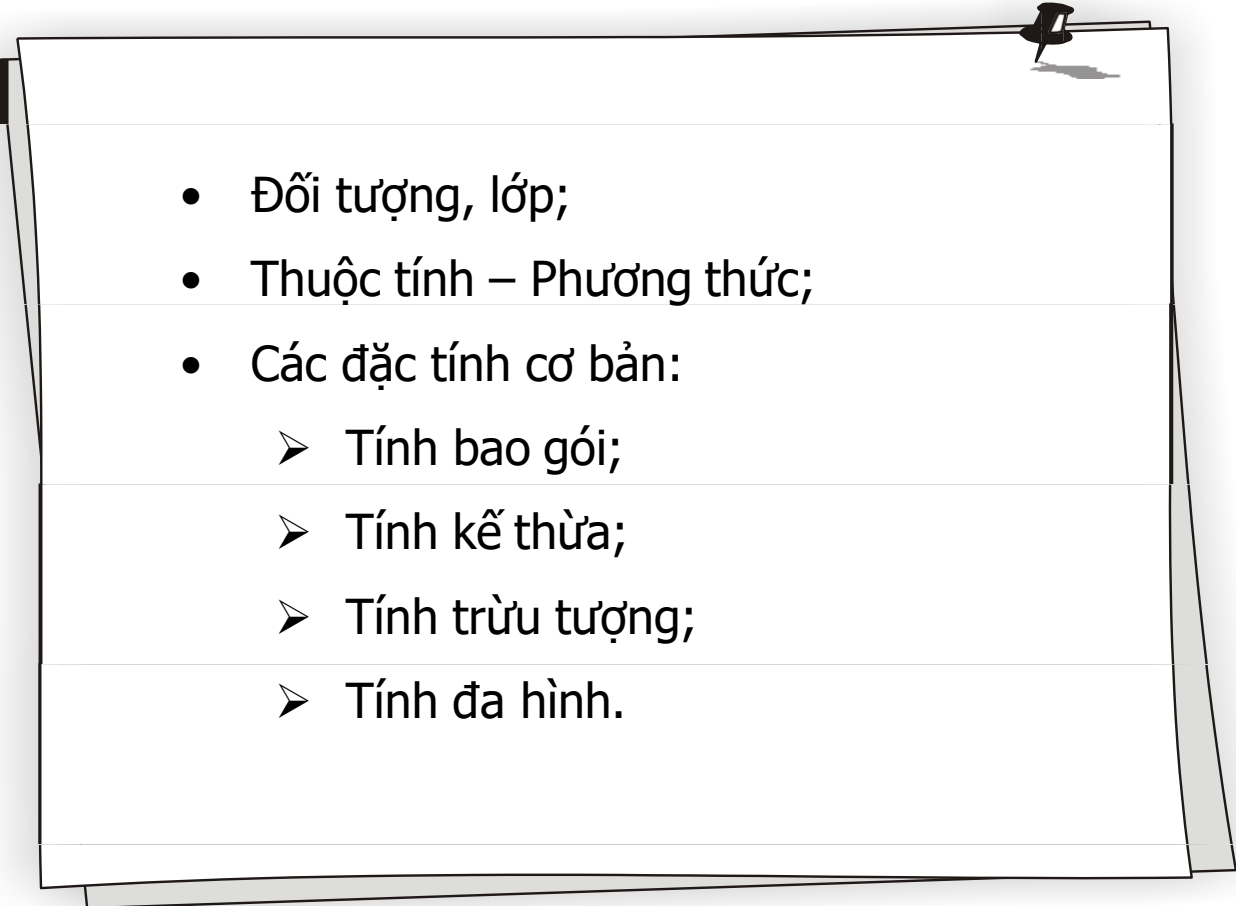
1.2.2. SO SÁNH HƯỚNG ĐỐI TƯỢNG VÀ HƯỚNG CẤU TRÚC (tiếp theo)

- Hướng đối tượng:
 - Hướng tiếp cận đa dạng gần với thực tế;
 - Tìm và phân tích mối quan hệ giữa các đối tượng trong bài toán;
 - Mã lệnh và dữ liệu liên kết trong thể thống nhất.
- Ưu thế:
 - Khả năng tái sử dụng cao;
 - Ổn định và dễ bảo trì;
 - Chi phí giảm dần.



Sơ đồ mô tả hành động mua xe của khách hàng

1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

- 
- Đối tượng, lớp;
 - Thuộc tính – Phương thức;
 - Các đặc tính cơ bản:
 - Tính bao gói;
 - Tính kế thừa;
 - Tính trừu tượng;
 - Tính đa hình.

1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (tiếp theo)

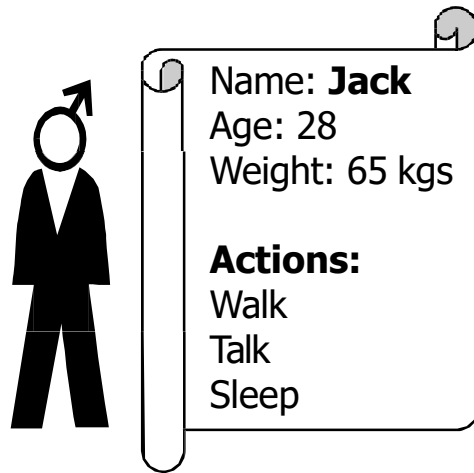


Một số đối tượng xe cần quản lý

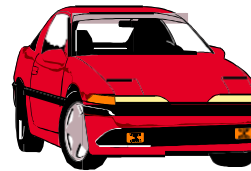
Đối tượng:

Mỗi đối tượng có những thuộc tính hay những đặc điểm mô tả cũng như những hành vi riêng nhằm phân biệt nó với các đối tượng khác.

1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (tiếp theo)



Đối tượng khách hàng



**Đối tượng Car
xe mà khách hàng mua**

Thuộc tính và hành vi của đối tượng:

- Thuộc tính là những đặc điểm đặc trưng của đối tượng, thể hiện thông qua những giá trị cụ thể.
- Hành vi là những cách thức mà qua đó đối tượng thể hiện sự hoạt động hay chức năng của chúng.

1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (tiếp theo)



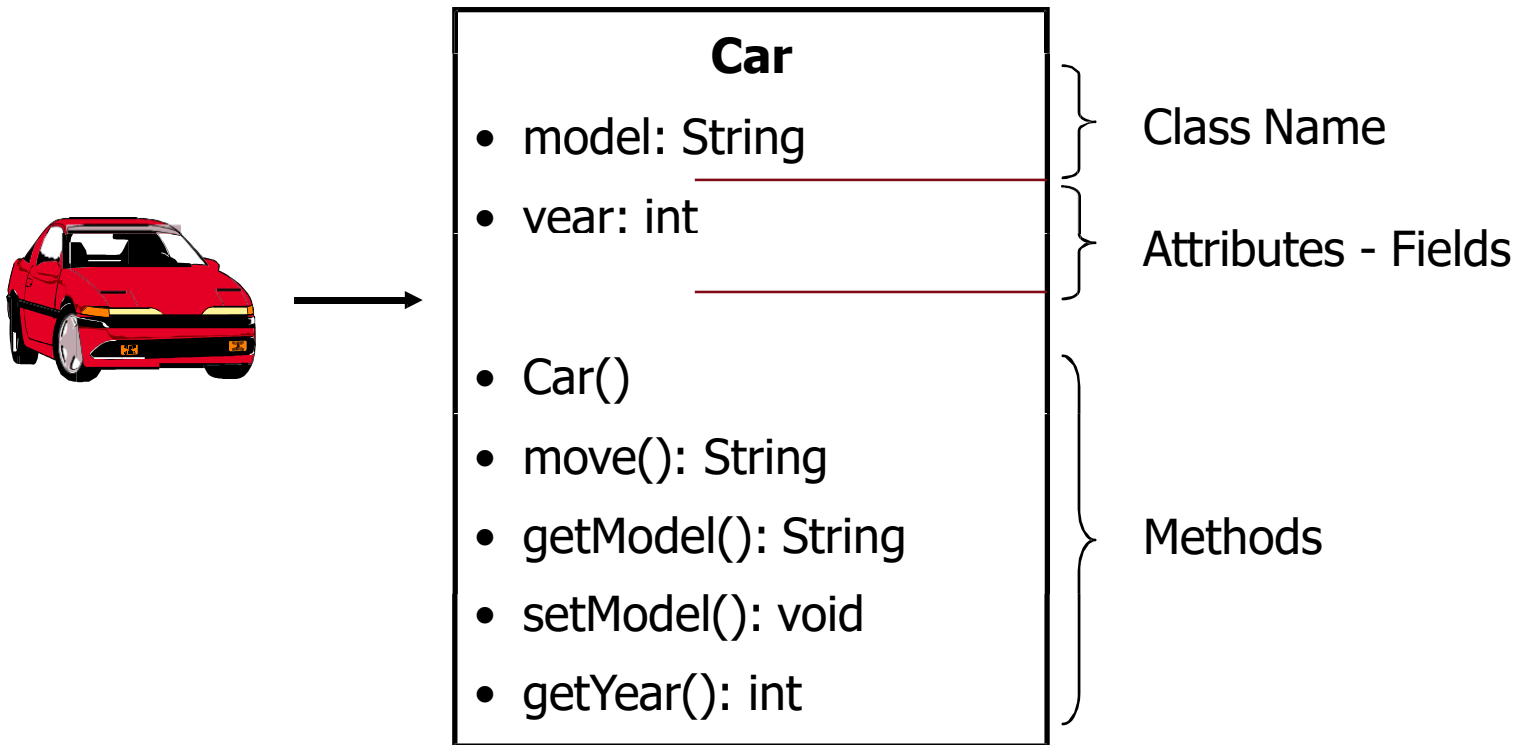
Lớp:

- Lớp là khái niệm dùng để mô tả một nhóm các đối tượng có những thuộc tính, hành vi và những mối quan hệ thông thường tương tự nhau;
- Thuật ngữ lớp có thể hiểu là cách nói vắn tắt của cụm từ “lớp các đối tượng”;
- Như vậy mỗi đối tượng được coi như là một “thể hiện” của lớp với những giá trị thuộc tính cũng như cách thức hoạt động đặc trưng.

1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (tiếp theo)

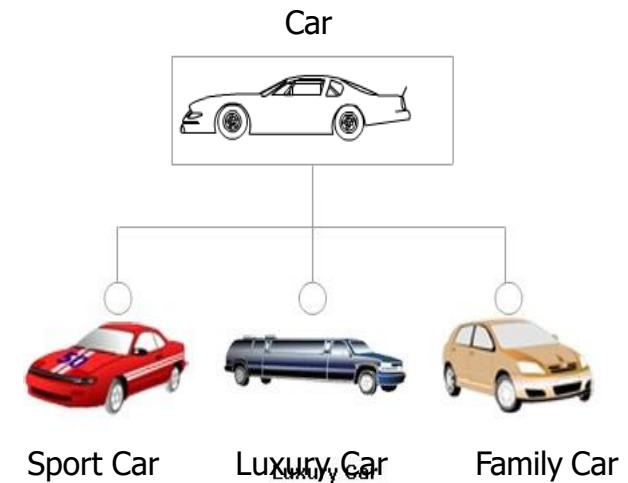
Sơ đồ mô tả lớp:

Sơ đồ lớp mô tả những đặc điểm khái quát nhất về lớp bao gồm: Tên lớp, các thuộc tính và các phương thức.



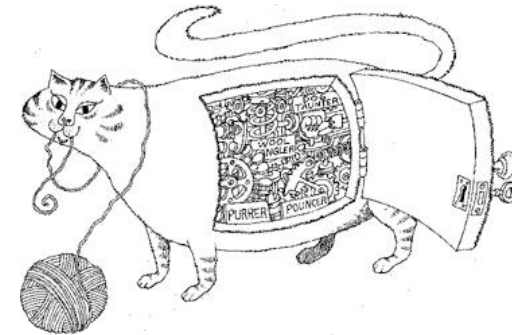
1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (tiếp theo)

- **Tính kế thừa:** Tính kế thừa cho phép các đối tượng có thể chia sẻ hay mở rộng các thuộc tính hoặc phương thức mà không phải tiến hành định nghĩa lại.



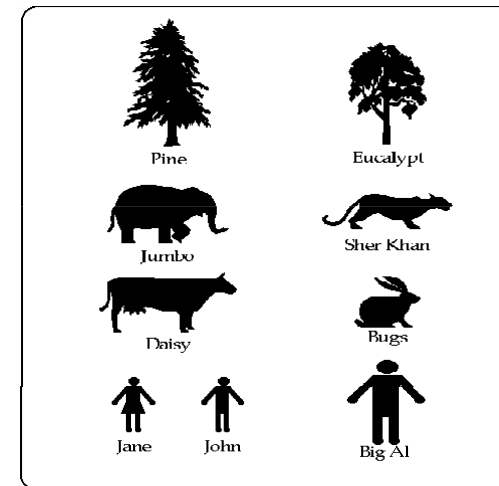
Các loại xe sport_car, luxury_car v.v. của đại lý Hùng Phát được phân cấp quản lý theo mô hình trên.

- **Tính bao gói:** Khả năng truy xuất vào các thành phần của một đối tượng trong khi vẫn đảm bảo che giấu các đặc tính riêng tư bên trong đối tượng được gọi là tính bao gói.



1.2.3. CÁC KHÁI NIỆM CƠ BẢN TRONG LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG (tiếp theo)

- **Tính trừu tượng:** Một đặc tả trừu tượng cho ta biết một đối tượng có thể làm gì mà không bận tâm vào việc nó làm như thế nào.



- **Tính đa hình:** Tính đa hình thể hiện khi với cùng một phương thức nhưng có thể có cách ứng xử khác nhau ở những lớp cùng giao diện.



CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 4 ▼

Điểm: 10

Một đối tượng là.....

- ☐ A. Các lớp được tạo thể hiện từ đó.
- ☐ B. Một thể hiện của lớp.
- ☐ C. Một tham chiếu đến một thuộc tính.
- ☐ D. Một biến

PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...

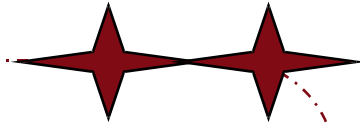


Edit in Quizmaker

CÂU HỎI THẢO LUẬN

Qua tìm hiểu giữa lớp và đối tượng, anh/chị nào có thể cho tôi biết sự khác nhau giữa lớp và đối tượng.

1.3. MÔ TẢ LỚP VÀ TẠO ĐỐI TƯỢNG TRONG JAVA



1. Cấu trúc cơ bản của Lớp.
2. Hàm tạo và vai trò của hàm tạo.
3. Tạo và sử dụng đối tượng.
4. Cách truy xuất vào các thành phần của đối tượng.
5. Vòng đời của một đối tượng.
6. Phân biệt Đối tượng và Lớp.
7. Mối tương quan giữa đối tượng thực tế và đối tượng phần mềm.

1.3.1. CẤU TRÚC CƠ BẢN CỦA LỚP

Cú pháp cơ bản

```
public class ClassName
{
    public ClassName()
    {

    }
}
```



Save



Class Name.java

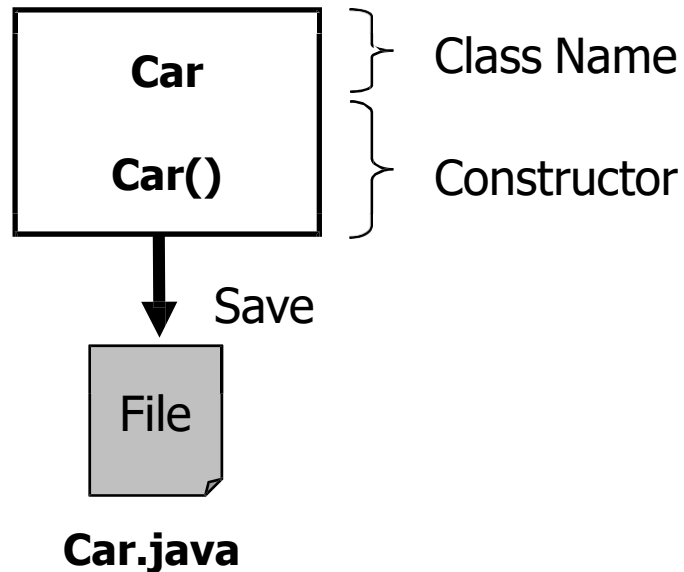
Kèm chú thích

```
/**
 * Write a description of class ClassName here.
 * @author (your name)
 * @version (a version number or a date)
 */
public class ClassName
{
    /**
     * Constructor for objects of class ClassName
     */
    public ClassName()
    {
        // To do:
    }
}
```

Lưu ý: Java là ngôn ngữ phân biệt chữ hoa chữ thường. Các **quy tắc đặt tên lớp** cần phải tuân thủ trong Java

1.3.1. CẤU TRÚC CƠ BẢN CỦA LỚP

Ví dụ mô tả lớp Car:



```
/**
 * Car.java
 * @author author name
 * @version 1/12/2010
 */
public class Car {
    /**
     * Hàm tạo đối tượng của lớp
     */
    public Car() {
        // Khởi tạo đối tượng:
    }
}
```

Hàm tạo (Constructor): Là hàm đặc biệt giúp tạo và khởi tạo các trạng thái ban đầu cho đối tượng.

1.3.1. CẤU TRÚC CƠ BẢN CỦA LỚP

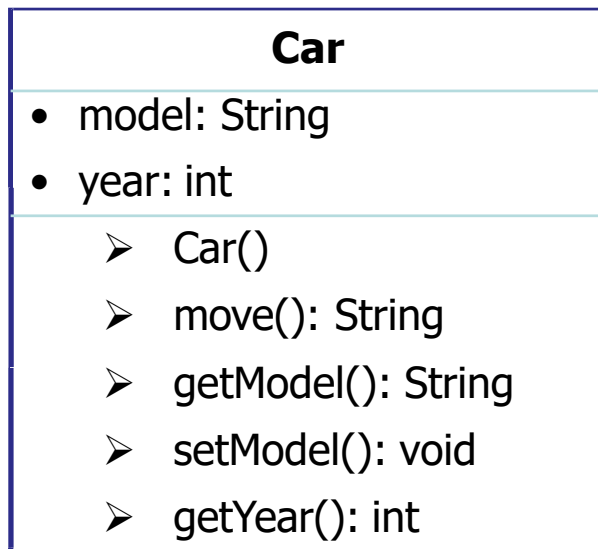
Bổ xung cho Car

- Các **thuộc tính**

- **model:** kiểu dữ liệu chuỗi
- **year:** kiểu dữ liệu số nguyên

- Các **phương thức**:

- move(): Có khả năng di chuyển "200km/h"
- getModel(), getYear(): Có khả năng cho biết kiểu dáng và năm sản xuất.



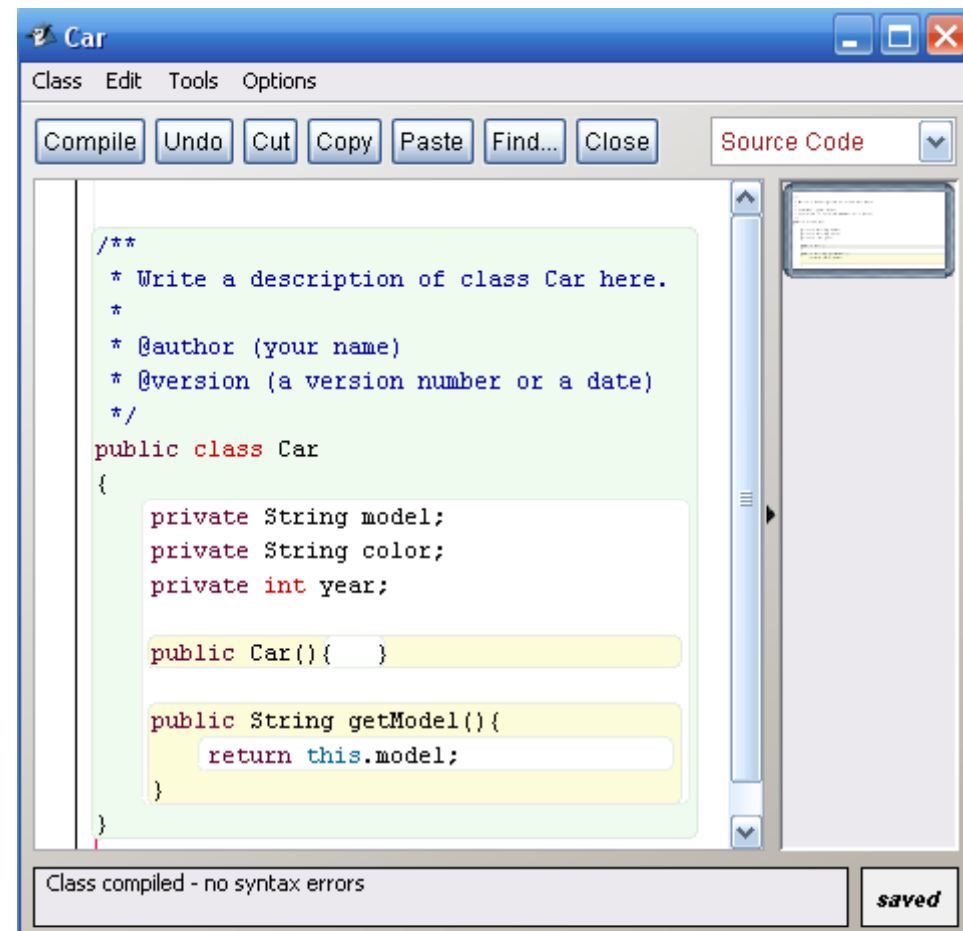
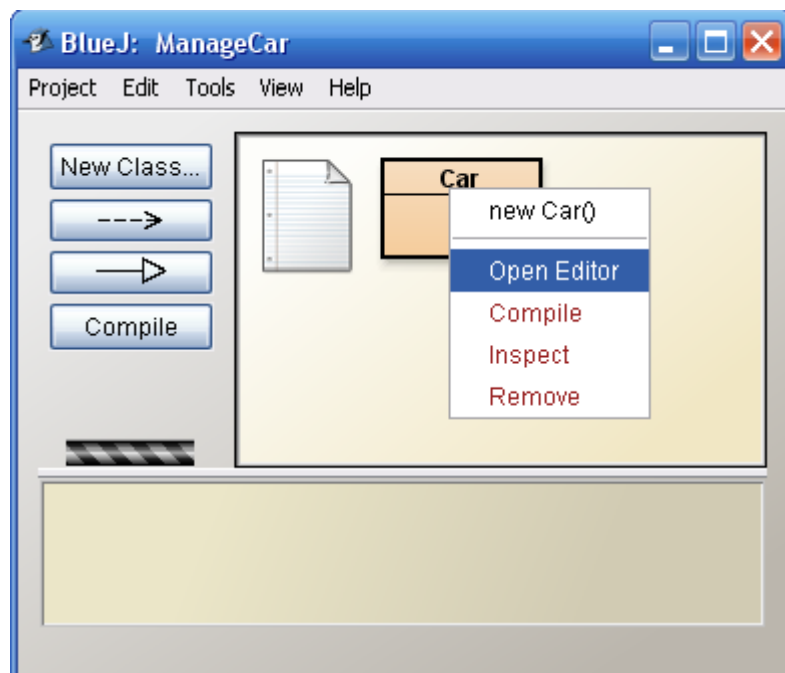
Java không cho phép mô tả thuộc tính trực tiếp mà gián tiếp thông qua **các trường dùng để lưu giá trị thuộc tính** của đối tượng:

```
public class Car {
    /**
     * Fields
     */
    private String model;
    private int year;
    /**
     * Constructors
     */
    public Car() {
    }
    /**
     * Methods
     */
    public String move () {
        return "200km/h";
    }
    public String getModel() {
        return this.model;
    }
    public int getYear() {
        return this.year;
    }
}
```

Java cho phép mô tả phương thức của đối tượng trong một lớp dưới dạng các hàm.

1.3.1. CẤU TRÚC CƠ BẢN CỦA LỚP (tiếp theo)

Tạo lớp Car trên BlueJ



1.3.2. HÀM TẠO VÀ VAI TRÒ CỦA HÀM TẠO

- Java đưa ra khái niệm Hàm tạo (Constructor) được mô tả trong mỗi lớp dùng để **tạo và khởi tạo** các trạng thái ban đầu của đối tượng;
- **Tên của hàm tạo luôn luôn trùng với tên của lớp;**
- Nếu lập trình viên không mô tả Hàm tạo nào thì mặc định sẽ có một **Hàm tạo không có đối số** được thêm vào cho lớp;
- Các giá trị ban đầu cho đối tượng có thể được truyền vào thông qua **Hàm tạo có đối số**;
- Ví dụ hàm khởi tạo trạng thái ban đầu của đối tượng:

```
public Car() {  
    model = "No model";  
    year = 2010;  
}
```

```
public Car(String m, int y) {  
    model = m;  
    year = y;  
}
```

1.3.3. HÀM TẠO VÀ SỬ DỤNG ĐỐI TƯỢNG

- Việc mô tả lớp chỉ có ý nghĩa đặc tả một nhóm đối tượng có chung đặc điểm thuộc tính và phương thức mà chưa thực sự tạo ra một đối tượng.
- Để tạo ra đối tượng ta sử dụng từ khóa **new** kết hợp với Hàm tạo.
- Ví dụ tạo một đối tượng Car thông qua Hàm tạo không có đối số như sau:

```
new Car ()
```

- Để truy xuất vào đối tượng vừa tạo cần phải khai báo một tham chiếu tới đối tượng thuộc lớp Car, cú pháp như sau:

```
Car c = new Car () ;
```

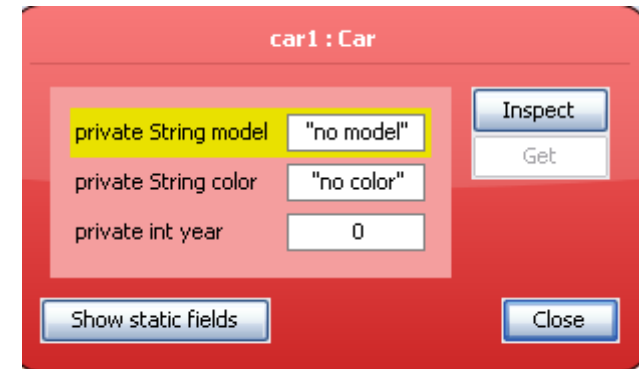
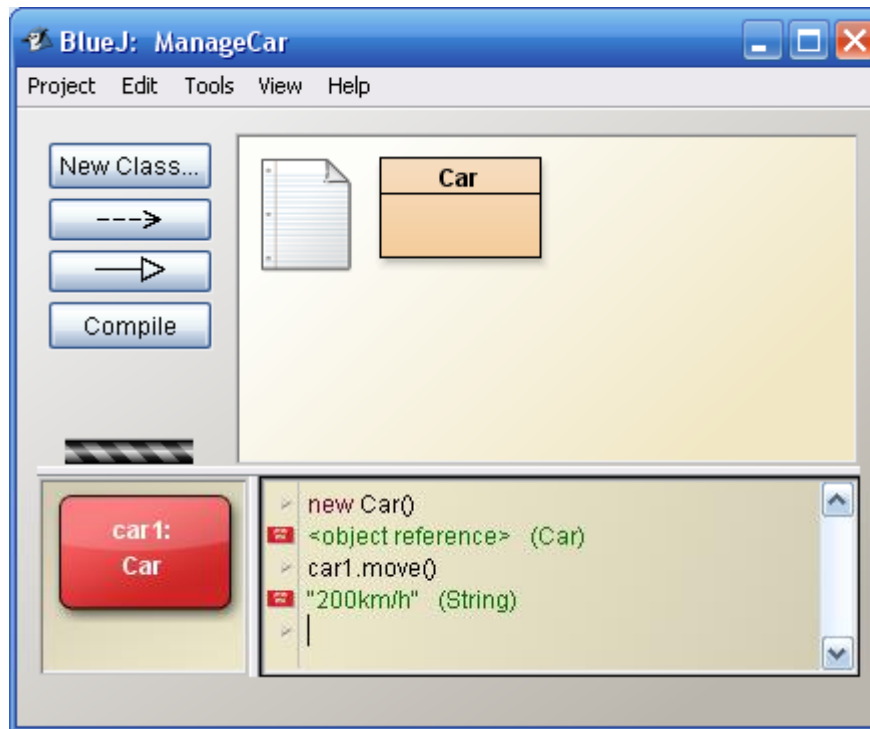

1.3.4. CÁCH TRUY XUẤT VÀO CÁC THÀNH PHẦN CỦA ĐỐI TƯỢNG

- Thông qua đối tượng tham chiếu ta có thể truy xuất vào các thành phần của đối tượng vừa được tạo (các phương thức và thuộc tính).
- Ví dụ muốn thực thi phương thức **move()** của đối tượng thuộc lớp **Car** ta khai báo và sử dụng thông qua đối tượng tham chiếu car1:

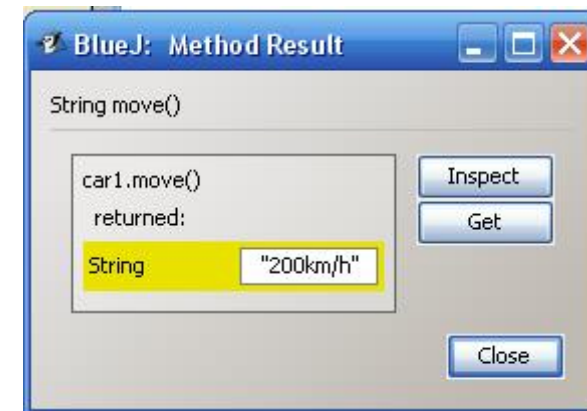
```
Car car1 = new Car ();  
car1.move ();
```

1.3.4. CÁCH TRUY XUẤT VÀO CÁC THÀNH PHẦN CỦA ĐỐI TƯỢNG (tiếp theo)

Kết quả chạy trên BlueJ



Trạng thái của đối tượng khi khởi tạo



Kết quả của lời gọi phương thức move()

1.3.5. VÒNG ĐỜI CỦA MỘT ĐỐI TƯỢNG

Khai báo một tham chiếu tới đối tượng:

```
Car c;
```



Sử dụng phương thức tạo để khởi tạo một đối tượng:

```
c = new Car();
```

Mỗi một object được sinh ra đều có thời gian sống, phạm vi hoạt động của riêng nó. Trong thời gian hoạt động ta có thể truy xuất vào các trường và phương thức có thể truy xuất của đối tượng.

```
c.move();  
c.getModel();
```



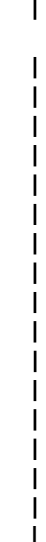
Đối tượng sẽ kết thúc vòng đời khi không còn được sử dụng nữa.

```
c = null;
```



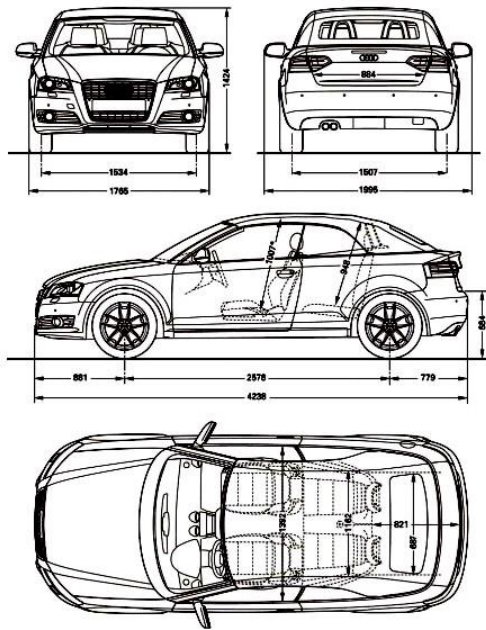
Garbage collector

c:Car



Life time

1.3.6. PHÂN BIỆT ĐỐI TƯỢNG VÀ LỚP



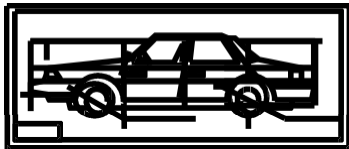
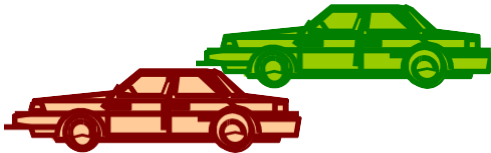
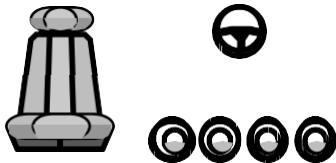

- **Lớp**

- Là đặc tả chi tiết về đối tượng (còn gọi là bản mẫu – blueprint).
- Một lớp có thể cho phép tạo ra nhiều đối tượng.

- **Đối tượng**

- Là thể hiện của một lớp với những đặc điểm, chức năng cụ thể.
- Các đối tượng từ cùng một lớp sẽ có cùng cấu trúc và chức năng như mô tả trong lớp.

1.3.7. MỐI TƯƠNG QUAN GIỮA ĐỐI TƯỢNG THỰC TẾ VÀ ĐỐI TƯỢNG PHẦN MỀM

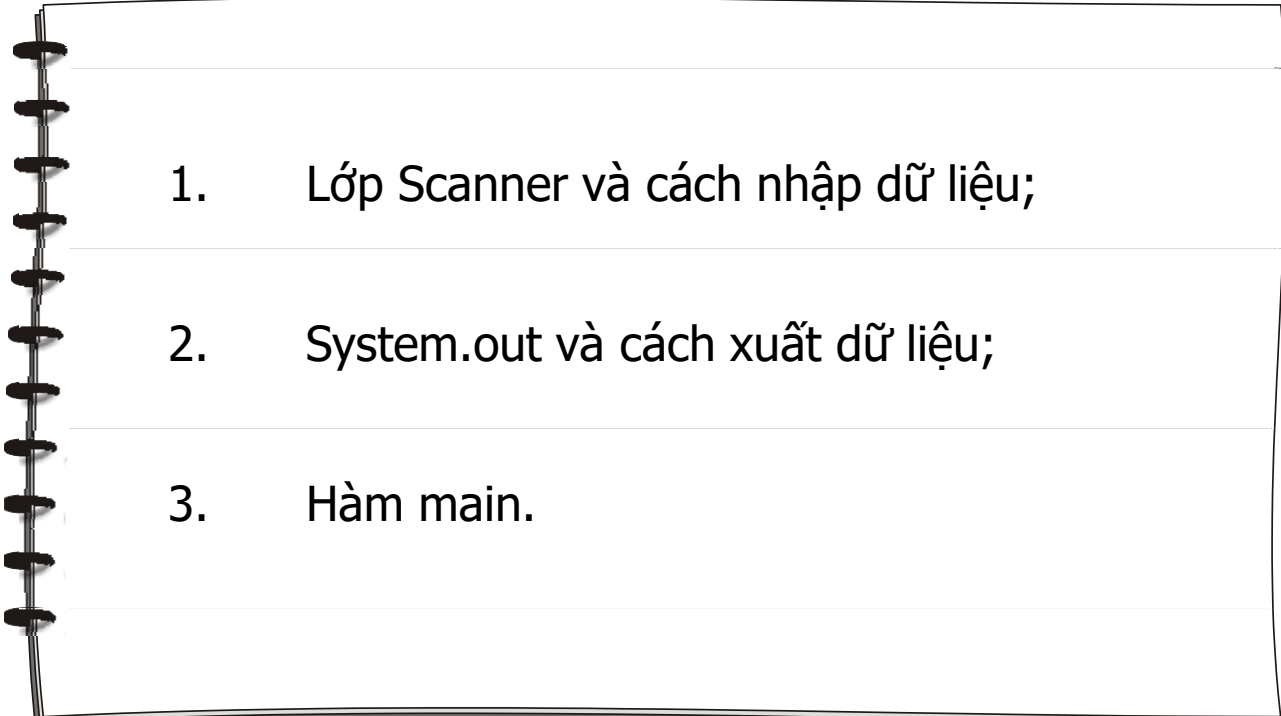
Khái niệm	Thực tế	Phần mềm
Lớp		<pre>public class Car { private int price; private String color; }</pre>
Đối tượng		<pre>Car myCar=new Car(); Car yourCar=new Car();</pre>
Thuộc tính		myCar <div> Price: 5.000\$ Color: "red" </div>
Phương thức		<pre>myCar.start(); myCar.setGear(1); myCar.accelerate(); myCar.turnLeft();</pre>

BÀI TẬP

Chương trình java đầu tiên cài đặt trên Bluej

- Chương trình in ra dòng chữ Hello World
- [Load code tham khảo](#)

1.4. LẬP TRÌNH ỨNG DỤNG CONSOLE VỚI JAVA

- 
1. Lớp Scanner và cách nhập dữ liệu;
 2. System.out và cách xuất dữ liệu;
 3. Hàm main.

1.4.1. LỚP SCANNER VÀ NHẬP DỮ LIỆU

- Lớp `java.util.Scanner` cho phép lập trình nhập dữ liệu cho các biến với các kiểu dữ liệu cơ bản khác nhau.
- Cách sử dụng:

```
Scanner input = new Scanner(System.in);  
  
int x = input.nextInt();  
double d = input.nextDouble();  
boolean b = input.nextBoolean();  
long l = input.nextLong();  
byte bi = input.nextByte();  
  
String s = input.nextLine();
```


1.4.2. SYSTEM.OUT VÀ XUẤT DỮ LIỆU

- Phương thức print(): In ra màn hình console một chuỗi.
- Phương thức printf() (có từ J2SE 5.0) có thể được sử dụng để xuất dữ liệu theo định dạng.

```
System.out.print("=====");  
System.out.println();  
System.out.printf("%s - %d", "Hello", 123);
```

Format Specifier	Description
%d	Result formatted as a decimal integer
%f	Result formatted as a real number
%o	Results formatted as an octal number
%e	Result formatted as a decimal number in scientific notation
%n	Result is displayed in a new line

1.4.3. HÀM MAIN

- Hàm main() là một hàm đặc biệt trong java. Một lớp bất kỳ đều có thể cài đặt hàm main, khi đó ta gọi lớp đó là Main Class.
- Một ứng dụng java thông thường sẽ được khởi chạy từ hàm main().
- Khi thực thi hàm main(), người dùng có thể khởi tạo các giá trị cho hàm thông qua các tham số truyền vào cho hàm main(), còn gọi là đối số dòng lệnh.

```
public class Demo {  
    public static void main(String[] args) {  
        System.out.print("Hello");  
        System.out.println(args[0]);  
        System.out.println(args[1]);  
    }  
}
```

```
$java Demo World Wide Web  
World  
Wide  
Web
```

BÀI TẬP

Dùng notepad để soạn thảo chương trình

- Chương trình in ra dòng chữ Hello World
- [Load code tham khảo](#)

BÀI TẬP 1: MÔ TẢ ĐỐI TƯỢNG CAR

- Viết một lớp mô các đối tượng xe ô tô lấy tên Car, với mỗi đối tượng xe ô tô sẽ có những đặc điểm sau:
 - Có thuộc tính tên mẫu xe;
 - Có thuộc tính mô tả màu;
 - Có thuộc tính tốc độ tối đa;
 - Có phương thức cho biết tên các loại xe;
 - Có phương thức cho biết màu xe;
 - Có phương thức cho biết tốc độ tối đa;
 - Có hàm tạo có tham số để khởi tạo các giá trị ban đầu như tên mẫu xe, màu xe, tốc độ tối đa.
- Sử dụng Bluej tạo và thực thi các phương thức của đối tượng thuộc lớp car.
- ***Đáp án tham khảo***

BÀI TẬP 2

Viết một lớp các đối tượng hình chữ nhật lấy tên là Rectangle, mỗi hình chữ nhật đều có những đặc tính sau:

- Có thông số chiều dài (int);
- Có thông số chiều rộng (int);
- Có tính năng tính diện tích;
- Có tính năng tính chu vi;
- Có tính năng hiển thị trực quan hình chữ nhật dưới dạng sau:

Ví dụ hình chữ nhật có chiều dài 7, chiều rộng 3 sẽ được hiển thị như sau lên màn hình console:

```
#####
```

```
#####
```

```
#####
```

- Sử dụng lớp Rectangle vừa định nghĩa để viết một chương trình java với kịch bản như sau:
- Khai báo và khởi tạo một đối tượng kiểu Rectangle;
- Hiển thị hình chữ nhật trên màn hình console;
- Thay đổi chiều dài và chiều rộng của hình chữ nhật;
- In ra màn hình diện tích và chu vi của hình chữ nhật đó.
- [Đáp án tham khảo](#)

Rectangle
- width: int - height: int
+ Rectangle() + Rectangle(int, int) + display(): void + getArea(): int + setHeight(int): void + getPerimeter(): int + getHeight(): int + getWidth(): int + setWidth(int): void

CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 3 ▼

Điểm: 10

Câu 1: Tất cả các đối tượng thuộc vào cùng một lớp có cùng thuộc tính và hành vi có thể có.

Câu 2: Một phương thức được gọi là constructor sử dụng để hủy một đối tượng.

- ☐ A. Cả hai câu đều đúng.
- ☐ B. Cả hai câu đều sai
- ☐ C. Câu 1 đúng, câu 2 sai.
- ☐ D. Câu 2 đúng, câu 1 sai.

PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

CÂU HỎI THẢO LUẬN

Qua tìm hiểu về hàm tạo anh/ chị nào có thể cho tôi biết sự khác nhau giữa hàm tạo và các hàm thông thường khác?

TÓM LƯỢC CUỐI BÀI

Sau khi học xong bài này chúng ta đã nắm được các kiến thức sau:

- Đặc trưng của công nghệ Java;
- Các khái niệm cơ bản trong lập trình hướng đối tượng;
- Viết ứng dụng theo hướng đối tượng với Java như:
 - Xây dựng lớp;
 - Tạo và sử dụng đối tượng;
 - Sử dụng lớp Scanner thực hiện việc nhập liệu.
- Cài đặt ứng dụng chạy trên BlueJ.

CÂU HỎI TRẮC NGHIỆM

Câu hỏi 1 trên 11 ▾

Điểm: 10

Đâu là không phải là đặc điểm của ngôn ngữ Java?

- ☐ A. Hướng đối tượng.
- ☐ B. An toàn.
- ☐ C. Mạnh mẽ.
- ☐ D. Đa dạng.

PROPERTIES

On passing, 'Finish' button:

On failing, 'Finish' button:

Allow user to leave quiz:

User may view slides after quiz:

User may attempt quiz:

Goes to Next Slide

Goes to Next Slide

At any time

At any time

Unlimited times



Properties...



Edit in Quizmaker

Máy ảo Java (Java Virtual Machine - JVM) là gì?

Máy ảo Java (Java Virtual Machine - JVM) là gì?

Gợi ý:

Máy ảo Java là một phần mềm dựa trên cơ sở máy tính ảo. Nó có tập hợp các lệnh logic để xác định các hoạt động của máy tính. Người at có thể xem nó như một hệ điều hành thu nhỏ. Nó thiết lập các lớp trừu tượng cho: phần cứng bên dưới, hệ điều hành, mã đã biên dịch.

Trình biên dịch chuyển mã nguồn thành tập các lệnh của máy ảo mà không phụ thuộc vào phần cứng cụ thể. Trình thông dịch trên mỗi máy sẽ chuyển tập lệnh này thành chương trình thực thi. Máy ảo tạo ra một môi trường bên trong để thực thi các lệnh bằng cách:

- Nạp các file .class
- Quản lý bộ nhớ
- Dọn "rác"

Java API là gì?

Bộ công cụ phát triển ứng dụng (JDK)?

Hàm print() và println() có gì khác nhau?

Các lệnh trong Java có phân biệt chữ hoa-thường không?

Trong khai báo biến, lớp, phương thức có quy định nào cho việc đặt tên không?

PROPERTIES

Allow user to leave interaction:

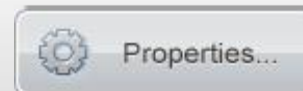
Show 'Next Slide' Button:

Completion Button Label:

[Anytime](#)

[Don't show](#)

[Next Slide](#)



A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Select a term:

API

Attribute

Class

CLASSPATH

Constructor

Garbage Collection

Hàm main()

IDE

JDK

JVM

Lớp Scanner

Object

API

Viết tắt của từ Application Programming Interface

PROPERTIES

Allow user to leave interaction:

Show 'Next Slide' Button:

Completion Button Label:

Anytime

Don't show

Next Slide



Properties...



Edit in Engage