

# Lớp (Class) và Đối Tượng (Object)

## Phần 1

# Hướng dẫn sử dụng ví dụ minh họa



Thứ tự:

Circle1 → Circle2 →  
.....→ Circle6 → Circle

```
1  public class Circle3 {
2
3      public static double SOPI=3.14;
4      private double x;
5      private double y;
6      private double R;
7      private double CV=0;
8      Circle3() {CV=x=y=R=0;}
9      Circle3(double toadoX, double toadoY, double bankinhR)
10         {x=toadoX; y=toadoY; R=bankinhR>=0?bankinhR:0;CV=2*R*SOPI;}
11
12      public double DT(){ return SOPI*R*R;}
13      public String toString()
14      {
15          return "Vong tron co tam o (" +x+", "+y+") va ban kinh R= "+R;
16      }
17  }
18
19  /*
20  Circle3 C3= new Circle3(1.13,-0.76,6.75);
21  System.out.println(C3);
22  //System.out.println(C3.CV);
23  System.out.println(C3.DT()+" voi gia tri PI la "+ Circle3.SOPI);
24  Circle3.SOPI=3.1;
25  System.out.println(C3.DT()+" voi gia tri PI la "+ Circle3.SOPI);
26
27  */
```

# Hướng dẫn sử dụng ví dụ minh họa

```
9 public class Circle3 {
10     public static double SOPI=3.14;
11     private double x;
12     private double y;
13     private double R;
14     private double CV=0;
15     Circle3() {CV=x=y=R=0;}
16     Circle3(double toadoX, double toadoY, double bankinhR)
17     {x=toadoX; y=toadoY; R=bankinhR>=0?bankinhR:0;CV=2*R*SOPI;}
18
19     public double DT(){ return SOPI*R*R;}
20     public String toString()
21     {
22         return "Vong tron co tam o (" +x+", "+y+") va ban kinh R= " +R;
23     }
24 }
25 /*
26 Circle3 C3= new Circle3(1.13,-0.76,6.75);
27 System.out.println(C3);
28 //System.out.println(C3.CV);
29 System.out.println(C3.DT()+" voi gia tri PI la "+ Circle3.SOPI);
30 Circle3.SOPI=3.1;
31 System.out.println(C3.DT()+" voi gia tri PI la "+ Circle3.SOPI);
32
33 */
```

Xem phần định nghĩa lớp

Một số lệnh thử nghiệm

Sẽ Copy/Paste vào Main  
(Main.java)

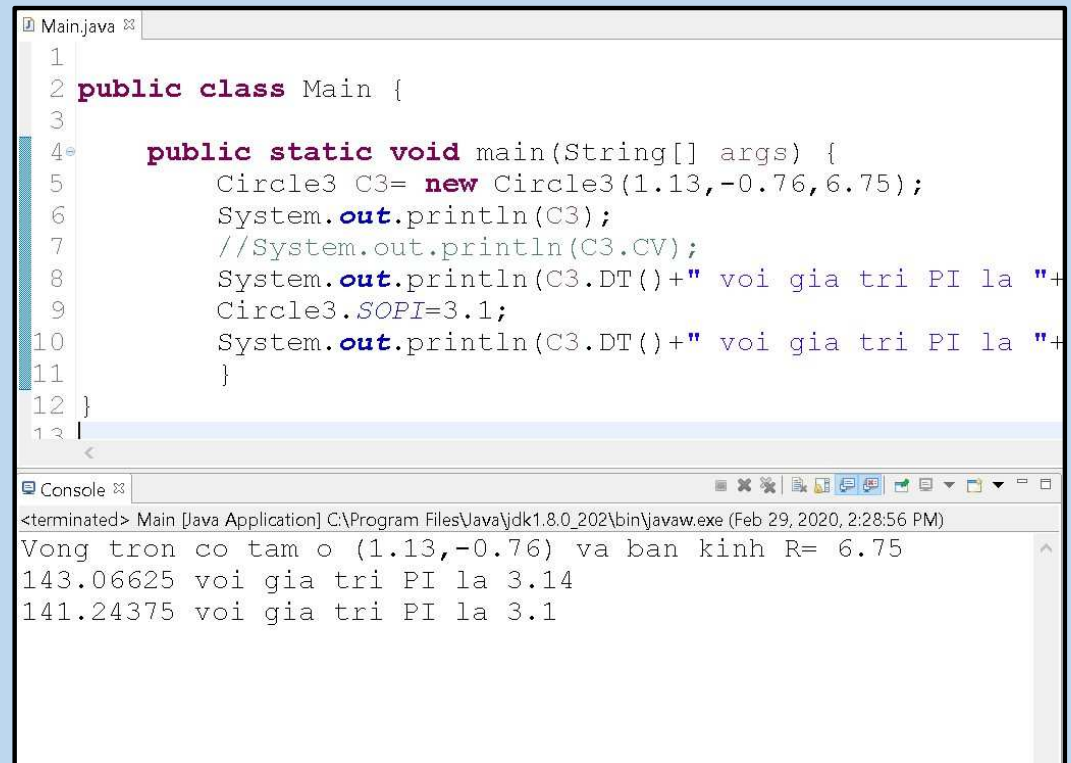
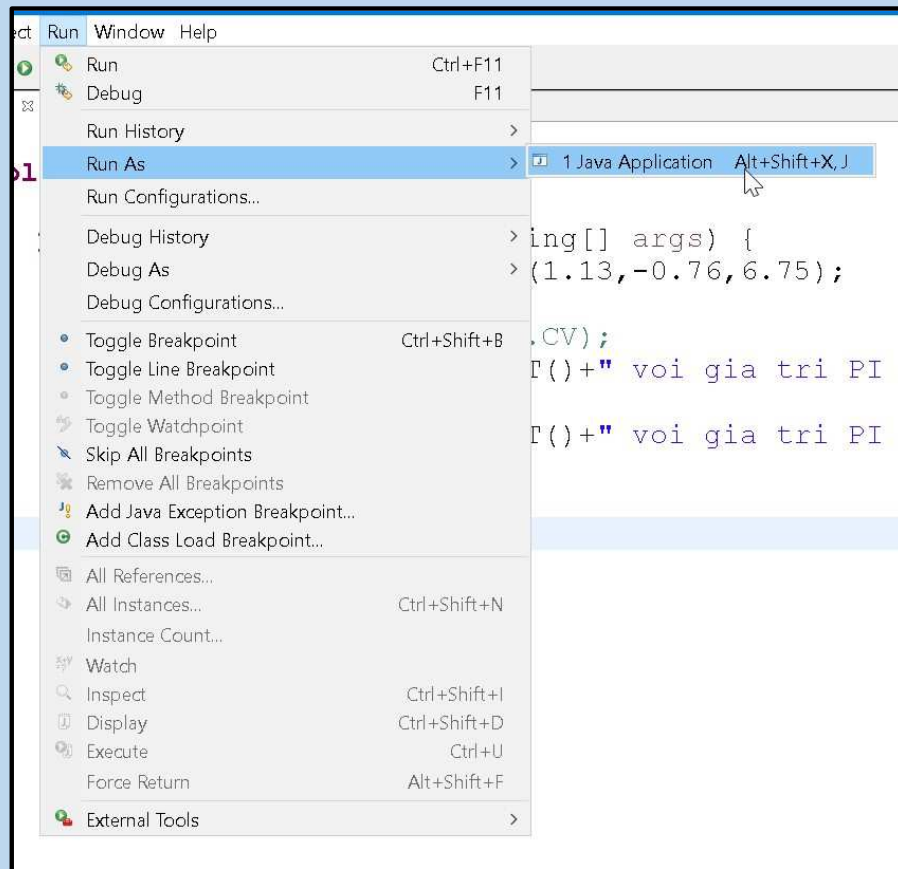
```
public class Main {
    public static void main(String[] args) {
    }
}
```

# Hướng dẫn sử dụng ví dụ minh họa

```
9 public class Circle3 {
10     public static double SOPI=3.14;
11     private double x;
12     private double y;
13     private double R;
14     private double CV=0;
15     Circle3() {CV=x=y=R=0;}
16     Circle3(double toadoX, double toadoY, double bankinhR)
17         {x=toadoX; y=toadoY; R=bankinhR>=0?bankinhR:0;CV=2*R*SOPI;}
18
19     public double DT(){ return SOPI*R*R;}
20     public String toString()
21     {
22         return "Vong tron co
23     }
24 }
25 /*
26 Circle3 C3= new Circle3(1.13,-0.76
27 System.out.println(C3);
28 //System.out.println(C3.CV
29 System.out.println(C3.DT()
30 Circle3.SOPI=3.1;
31 System.out.println(C3.DT()
32 */
33 */
```

```
*Main.java
1
2 public class Main {
3
4     public static void main(String[] args) {
5         Circle3 C3= new Circle3(1.13,-0.76,6.75);
6         System.out.println(C3);
7         //System.out.println(C3.CV);
8         System.out.println(C3.DT()+" voi gia tri PI la "+ Circle3.SOPI);
9         Circle3.SOPI=3.1;
10        System.out.println(C3.DT()+" voi gia tri PI la "+ Circle3.SOPI);
11    }
12 }
13
```

# Hướng dẫn sử dụng ví dụ minh họa



# Tạo lớp đơn giản Circle1

```
public class Circle1  
{  
    public double x; //hoành độ tâm O  
    public double y; //tung độ tâm O  
    public double R; //bán kính  
}
```

Nhận xét:

Giống như struct trong C/C++: lưu trữ dữ liệu.

public: công khai/công cộng/truy cập tự do!!

# Tạo lớp đơn giản Circle1

Thử nghiệm Circle1 trong Main

```
public class Main {  
    public static void main(String[] args) {  
        Circle1 C1= new Circle1();  
        C1.x=0;  
        C1.y=0;  
        C1.R=-2;
```

```
        System.out.println("Vong tron C1 co tam O("+C1.x+", "+C1.y+") va ban kinh  
                               R= "+C1.R);  
        Circle1 C2=C1;  
        System.out.println("Vong tron C2 co tam O("+C2.x+", "+C2.y+") va ban kinh  
                               R= "+C2.R);
```

Nhận xét:

Khai báo 1 đối tượng của 1 lớp

<Tên lớp> <tên đối tượng>=new <Tên Lớp>()

Liên tưởng kiểu dữ liệu cơ bản:

long L = 12;

double D=34.1;

Gán trực tiếp dữ liệu vào vùng tin (thuộc tính/properties)!!

Gán 1 biến đối tượng A cho đối tượng B: theo nghĩa cùng chỉ tới cùng 1 nơi!!

Khác với

long L1=3;

long L2=L1;

# Tạo lớp đơn giản Circle1

Thay đổi giá trị các thuộc tính trong đối tượng C2  
Sau đó in ra các vùng của C1 và C2 → như nhau!!

```
C2.x=1; C2.y=1; C2.R=2;
```

```
System.out.println("Vong tron C1 co tam O("+C1.x+", "+C1.y+") va ban kinh R= "+C1.R);
```

```
System.out.println("Vong tron C2 co tam O("+C2.x+", "+C2.y+") va ban kinh R= "+C2.R);
```

```
Circle1 C3=new Circle1();
```

```
C3.x=C1.x; C3.y=C1.y; C3.R=C1.R;
```

```
C1.x=-1;
```

Khai báo đối tượng C3 và cấp vùng nhớ riêng!!!

Gán các giá trị trong C3 tương ứng từ C1

Thay đổi x của C1

Sau đó in ra các vùng của C1 và C3 → khác nhau!!

```
System.out.println("Vong tron C1 co tam O("+C1.x+", "+C1.y+") va ban kinh R= "+C1.R);
```

```
System.out.println("Vong tron C3 co tam O("+C3.x+", "+C3.y+") va ban kinh R= "+C3.R);
```

```
Circle1 CircleList[]= new Circle1[]{ new Circle1(), new Circle1(), new Circle1()};
```

```
} //Main
```

```
}// class
```

Khai báo mảng các đối tượng

<tên lớp> <tên mảng>[]= new <tên lớp>[

Nhớ lại

double arrav[]={1.2, 4.5, 6.7};



# Tạo lớp đơn giản Circle1

## Lớp

- Lớp (class) là **khuôn mẫu** (template) định nghĩa các **thuộc tính** (properties) và **hành vi** (behaviors) của các thực thể.
- Cú pháp khai báo lớp trong Java

```
[public] class TênLớp {  
    [Các thuộc tính]  
    [Các phương thức]  
}
```

Xem slide 3-7 bài 3 Lớp và Đối tượng trong Giáo Trình

## Thuộc tính

- Các thuộc tính của lớp dùng **lưu trữ** thông tin của thực thể.
- Cú pháp khai báo thuộc tính trong Java

```
[AccessModifier] KiểuDữLiệu tênThuộcTính;
```

```
graph TD; A["[AccessModifier]"] --> B["private"]; A --> C["public"]; A --> D["protected"]; A --> E["default"];
```

**private**   public   protected   default

```
public class SinhVien {  
    private String maSo;  
    private String hoTen;  
    private boolean daTotNghiep;  
}
```

# Tạo lớp đơn giản Circle1: Tổng kết

- Mức độ đơn giản: hoàn toàn giống struct
- public: truy cập được từ bên ngoài “gói” (package)
- Không an toàn ví dụ **C1.R=-2;**
- Phân biệt khai báo và cấp phát vùng nhớ lưu trữ
- Gán 2 biến đối tượng (phép =) : cùng chỉ đến 1 vùng lưu trữ
- Khác nhau giữa?

Circle1 listA[ ]= new Circle1[2];

Circle1 listB[ ]= new Circle1[]{new Circle1(), new Circle1()};

- Cách gọi:

Kiểu dữ liệu (Data Type)/Biến(Variable)  $\longleftrightarrow$  Lớp  
(Class)/Đối tượng(Object)

# Tạo lớp đơn giản Circle2

```
public class Circle1 {  
    public double x;  
    public double y;  
    public double R;  
}
```

- Ý tưởng cải tiến

Cần kiểm soát giá trị bán kính: không thể là giá trị  $< 0$ .

Có thể khai báo+cấp phát+gán giá trị ban đầu cùng lúc

tương tự như `long l=34; double a[] {4.5, 6.2, 8.1};`

→ Bổ sung:

*Cơ chế kiểm soát việc gán giá trị cho R → lệnh → phương thức (methods)*

*Hàm khởi tạo/hàm xây dựng (constructors) đảm nhiệm việc cấp phát và khởi tạo.*

*Hàm khởi tạo: thống nhất trùng tên với tên lớp.*

*Cho phép có nhiều hàm khởi tạo (nhưng danh sách tham số khác nhau để đáp ứng nhiều yêu cầu gán giá trị ban đầu khác nhau)*

# Tạo lớp đơn giản Circle2

```
public class Circle2 {  
    private double x;  
    private double y;  
    private double R;  
    Circle2(double toadoX, double toadoY, double bankinhR)  
        {x=toadoX; y=toadoY; R=bankinhR>=0?bankinhR:0;}  
    public String toString()  
    {  
        return "Vong tron co tam o (" + x + ", " + y + ") va ban kinh R= " + R;  
    }  
}
```

- Chú ý:

- +từ khóa private

- +hàm xây dựng/khởi tạo: cùng tên lớp và có 3 tham số

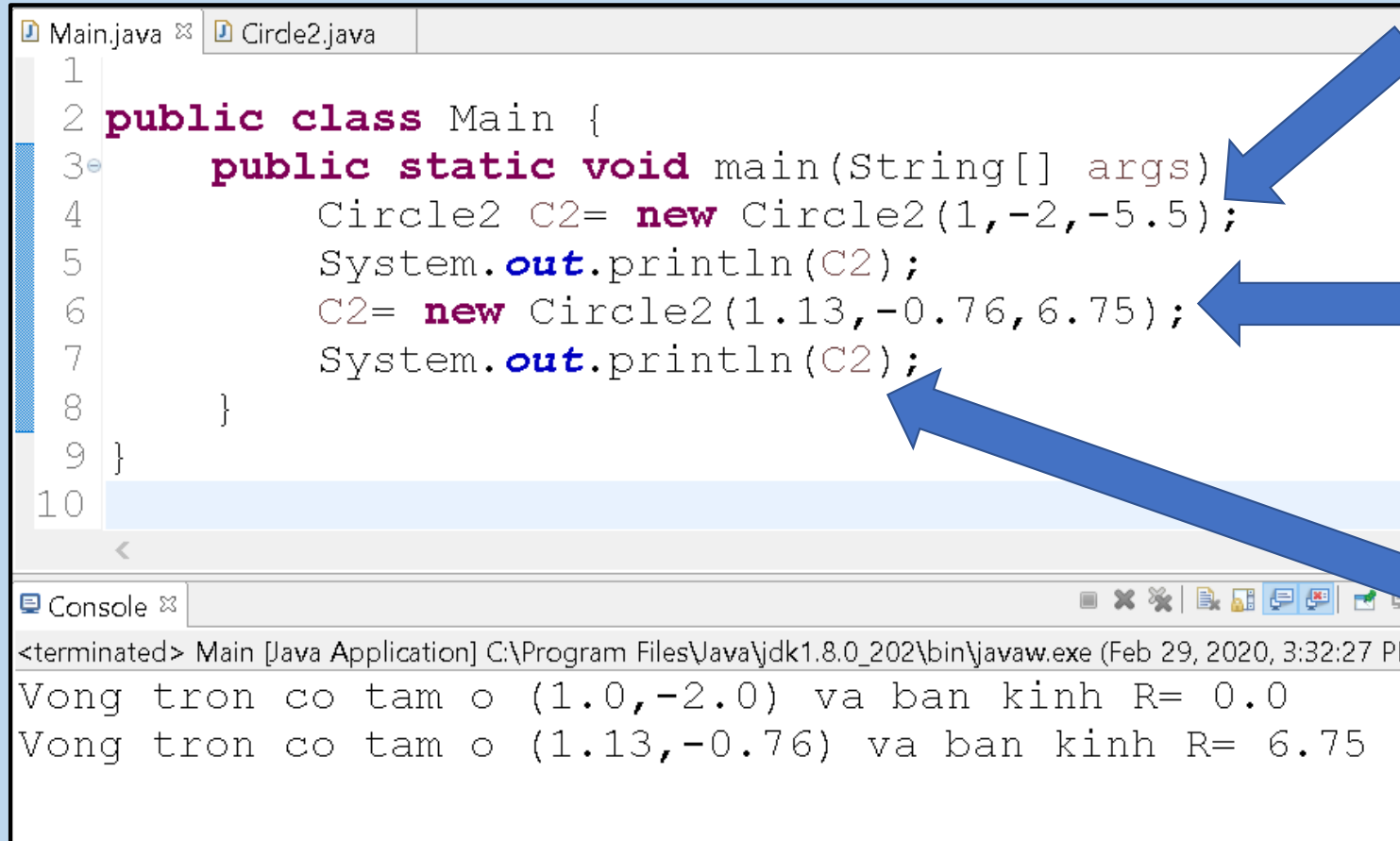
- +phương thức toString() kết quả trả về là 1 chuỗi

# Tạo lớp đơn giản Circle2

```
public class Main {  
    public static void main(String[] args) {  
        Circle2 C2= new Circle2(1,-2,-5.5);  
        //Circle2 C3= new Circle2(); dòng này báo lỗi  
        //System.out.println("Vong tron co tam o (" +C2.x+", "+C2.y+") " +  
        // "va ban kinh R= " +C2.R); dòng này báo lỗi  
        System.out.println(C2);  
        C2= new Circle2(1.13,-0.76,6.75);  
        System.out.println(C2);  
    }  
}
```

- Chú ý:
- Báo lỗi Circle2 C3 = new Circle2( ); → ?
- Báo lỗi dòng System.out.println → ?
- Câu lệnh System.out.println(C2);

# Tạo lớp đơn giản Circle2



```
1
2 public class Main {
3     public static void main(String[] args)
4         Circle2 C2= new Circle2(1,-2,-5.5);
5         System.out.println(C2);
6         C2= new Circle2(1.13,-0.76,6.75);
7         System.out.println(C2);
8     }
9 }
10
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 29, 2020, 3:32:27 PM)  
Vong tron co tam o (1.0,-2.0) va ban kinh R= 0.0  
Vong tron co tam o (1.13,-0.76) va ban kinh R= 6.75

## Nhận xét

- Gán 1 bán kính có giá trị < 0
- Gán 1 giá trị hợp lệ cho R
- phương thức print/println(đối tượng) → gọi toString() của lớp/đối tượng

# Tạo lớp đơn giản Circle2

- Xem slide 14 bài 3 Giáo trình Lớp và Đối tượng.
- Phạm vi [AccessModifier]

public

protected

private

## Phương thức khởi tạo

- Phương thức khởi tạo (constructor) được gọi đầu tiên khi tạo đối tượng.
- Đặc điểm phương thức khởi tạo trong Java
  - Tên phương thức **trùng tên lớp**.
  - **Không có** kiểu dữ liệu trả về.
- Cú pháp

```
[AccessModifier] TênLớp ([ThamSố]);
```

# Tạo lớp đơn giản Circle2: Tổng kết

- Thiết kế các hàm xây dựng/khởi tạo: cấp phát vùng nhớ, gán giá trị ban đầu và có sự kiểm soát
- Có thể thiết kế nhiều hàm khởi tạo!!
- `Circle2(double x) {...}`   `Circle2(double x, double y) {...}`
- Nếu lớp hoàn toàn không có hàm xây dựng → Java cung cấp 1 hàm xây dựng mặc định không tham số. Nếu có ít nhất 1 hàm xây dựng thì Java không cung cấp hàm mặc định này (→ người lập trình nếu muốn có thì tự viết thêm)
- Nguyên tắc: tránh bên ngoài có thể thay đổi giá trị các vùng thông tin mà không có sự kiểm soát (từ khóa `private`)
- Phương thức `print/println` và 1 số phương thức khác sẽ in ra 1 chuỗi trả về khi gọi phương thức `toString()` của 1 lớp/đối tượng → có thể thiết kế hàm `toString()` để in các kết quả cần thiết (mà không cần mô tả cụ thể trong chương trình sử dụng nó)



# Tạo lớp đơn giản Circle3

```
public class Circle2 {  
    private double x;  
    private double y;  
    private double R;  
    Circle2(double toadoX, double toadoY, double bankinhR)  
        {x=toadoX; y=toadoY; R=bankinhR>=0?bankinhR:0;}  
    public String toString()  
    {  
        return "Vong tron co tam o (" + x + ", " + y + ") va ban kinh R= " + R;  
    }  
}
```

- Vấn đề
- R gán ban đầu nhưng giữa chừng muốn đổi giá trị R-giữ nguyên tọa độ tâm O?
- ➔ có thể new đối tượng mới C2 = new Circle2(C2.x, C2.y, 1.23)
- ➔ có thể cung cấp 1 phương thức cho phép đổi R có kiểm soát

# Tạo lớp đơn giản Circle3

```
public class Circle3 {  
    private double x;  
    private double y;  
    private double R;  
    Circle3() {x=y=R=0;}  
    Circle3(double toadoX, double toadoY, double bankinhR)  
        {x=toadoX; y=toadoY; setR(bankinhR);}  
    public double getR(){return R;};  
    public void setR(double bk){R=bk>=0?bk:R;}  
    public String toString()  
    {  
        return "Vong tron co tam o (" + x + ", " + y + ") va ban kinh R= " + R;  
    }  
}
```

- Phương thức getR và setR: getter/setter.
- vùng R: private + kiểm tra bằng setR

# Tạo lớp đơn giản Circle3

```
2 public class Main {  
3 public static void main(String[] args) {  
4     Circle3 C3= new Circle3();  
5     System.out.println(C3);  
6     C3.setR(1);  
7     System.out.println(C3);  
8     C3.setR(-2);  
9     System.out.println(C3);  
10    System.out.println("Chu vi= "+3.14*C3.getR()+  
11        "    DT="+3.14*C3.getR()*C3.getR());  
12 }  
13 }
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 29, 2020, 4:09:52 PM)

Vong tron co tam o (0.0,0.0) va ban kinh R= 0.0  
Vong tron co tam o (0.0,0.0) va ban kinh R= 1.0  
Vong tron co tam o (0.0,0.0) va ban kinh R= 1.0  
Chu vi= 3.14 DT=3.14

Nhận xét:

Ban đầu gán 0  
cho mọi vùng.

Thay đổi giá trị R  
(bằng phương  
thức)

Lấy giá trị R để  
tính chu vi, diện  
tích

# Tạo lớp đơn giản Circle3: Tổng kết

- Qui định đặt tên phương thức (bắt buộc) để thiết lập/trả về giá trị của thuộc tính: xem slide 12 của giáo trình (bài 3-Lớp và đối tượng)

- Ví dụ: x: hoành độ tâm O, kiểu double, private

→ thiết lập giá trị cho x

```
public setx(.....) {.....}
```

→ lấy giá trị của x

```
public double getx() {.....return giá trị;}
```

- Nhờ các phương thức getter/setter: vẫn đảm bảo an toàn dữ liệu và có thể thay đổi dữ liệu có kiểm soát.
- Nên viết thêm hàm xây dựng không tham số (thuận lợi cho NSD).

## Phương thức getter và setter

- Phương thức cho phép **đọc** dữ liệu của một thuộc tính gọi là phương thức **getter**. Quý ước đặt tên
  - Thuộc tính kiểu bool: **isTênThuộcTính()**
  - Thuộc tính kiểu khác: **getTênThuộcTính()**
- Phương thức cho phép **ghi** dữ liệu vào một thuộc tính gọi là phương thức **setter**. Quý ước đặt tên:
  - **setTênThuộcTính([ThamSố])**
  - [ThamSố] thường là một tham số cùng kiểu dữ liệu với thuộc tính.

# Tạo lớp đơn giản Circle4

```
System.out.println(C3);  
System.out.println("Chu vi= "+3.14*C3.getR()+  
    "    DT="+3.14*C3.getR()*C3.getR());  
,
```

- Vấn đề:
- NSD phải biết giá trị Pi? phải biết công thức tính diện tích và chu vi hình tròn!!

➔ Lớp cung cấp đầy đủ các phương thức cần thiết, NSD chỉ có dùng!!

Cải tiến:

Cung cấp thêm thông tin về Diện tích và Chu vi hình tròn.

# Tạo lớp đơn giản Circle4

```
public class Circle4 {  
    private static double SOPI=3.14;  
    private double x;    private double y;    private double R;  
    Circle4() {x=y=R=0;}  
    Circle4(double toadoX, double toadoY, double bankinhR)  
        {x=toadoX; y=toadoY; setR(bankinhR);}  
    public double getR(){return R;};  
    public void setR(double bk){R=bk>=0?bk:R;}  
    public String toString()  
    {  
        String s="O("+x+", "+y+")\n";  
        s+="R= "+R+"\n";  
        s+="Chu vi= "+2*SOPI*R+"    Dien tich= "+R*R*SOPI+" (Pi="+SOPI+")\n";  
        return s;  
    }  
}
```

Giá trị cho số PI: private + static

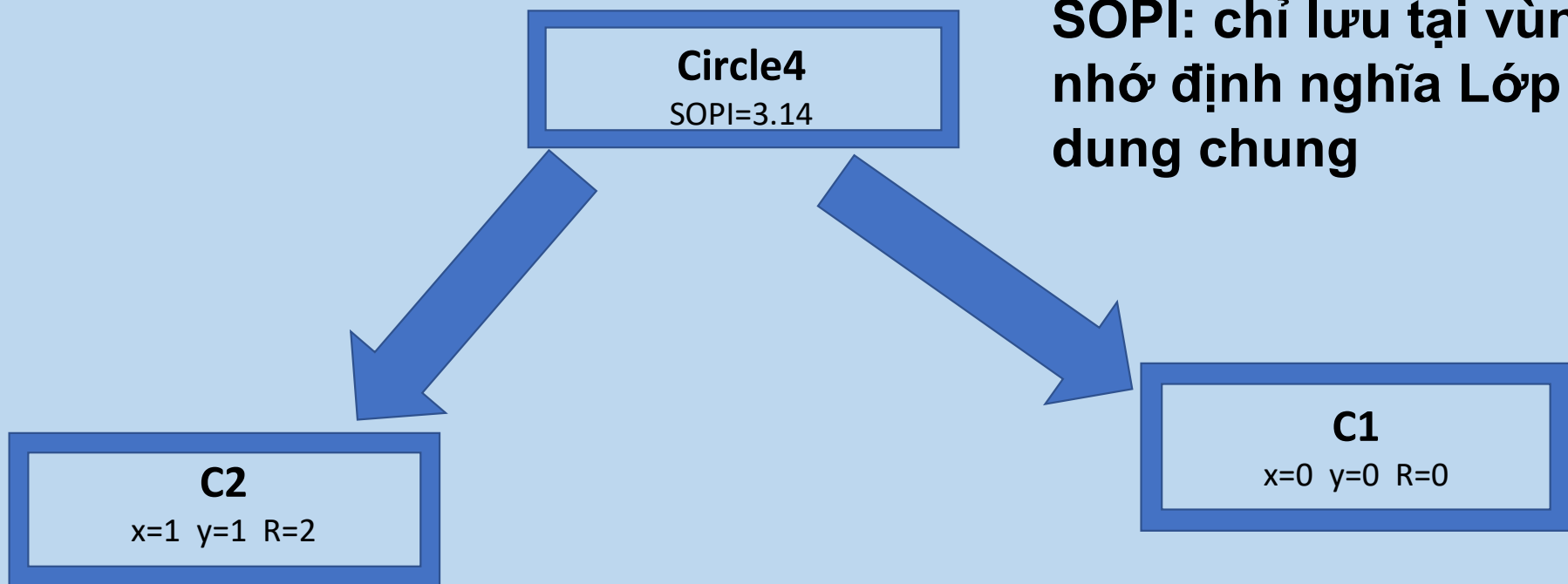
Bổ sung thêm thông tin chu vi và diện tích hình tròn

# Tạo lớp đơn giản Circle4

- Circle4 C1= new Circle4();
- Circle4 C2= new Circle4(1,1,2)

**C1, C2 được cấp vùng nhớ riêng để lưu trữ các giá trị x, y và R**

**SOPI: chỉ lưu tại vùng nhớ định nghĩa Lớp và dung chung**



# Tạo lớp đơn giản Circle4

```
2 public class Main {  
3     public static void main(String[] args) {  
4         Circle4 C4= new Circle4();  
5         System.out.println(C4);  
6         C4.setR(10);  
7         System.out.println(C4);  
8     }  
9 }  
10
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 29, 2020, 4:55:32 PM)

O(0.0,0.0)

R= 0.0

Chu vi= 0.0 Dien tich= 0.0 (Pi=3.14)

O(0.0,0.0)

R= 10.0

Chu vi= 62.80000000000000004 Dien tich= 314.0 (Pi=3.14)

Nhận xét:

Tính toán diện  
tích và chu vi  
dựa trên  
SOPI=3.14



# Tạo lớp đơn giản Circle4: Tổng kết

- static là gì: tĩnh, gắn với định nghĩa lớp → luôn tồn tại sau khi “ nạp lớp”
- non-static là gì: “động” do gắn với đối tượng → mỗi đối tượng có riêng các vùng non-static
- private là gì: phạm vi truy cập giới hạn trong Lớp
- public là gì: phạm vi truy cập là mọi nơi.
- → protected, final: sẽ tìm hiểu sau.
- Làm sao xác định thuộc tính XYZ là static hay non-static?
- Làm sao xác định Lớp, thuộc tính, phương thức là static hay non-static?

# Tạo lớp đơn giản Circle5

```
2 public class Main {  
3 public static void main(String[] args) {  
4     Circle4 C4= new Circle4(1,1,2);  
5     System.out.println(C4);  
6     C4.setR(2);  
7     System.out.println(C4);  
8 }  
9 }  
10
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 2

O(1.0,1.0)

R= 2.0

Chu vi= 12.56 Dien tich= 12.56 (Pi=3.14)

O(1.0,1.0)

R= 2.0

Chu vi= 12.56 Dien tich= 12.56 (Pi=3.14)

```
public class Circle4 {  
    private static double SOPI=3.14;  
    private double x;    private double y;  
    Circle4() {x=y=R=0;}  
}
```

- Vấn đề đặt ra

Vài ứng dụng muốn thay đổi giá trị số PI!! cần chính xác hơn (ví dụ 3.14159) hoặc gần đúng là đủ (3.1)

→ thay đổi số PI

→ Bỏ private!!

# Tạo lớp đơn giản Circle5

```
public class Circle5 {
    static double SOPI=3.14;
    private double x;
    private double y;
    private double R;
    Circle5() {x=y=R=0;}
    Circle5(double toadoX, double toadoY, double bankinhR)
        {x=toadoX; y=toadoY; setR(bankinhR);}
    public double getR(){return R;};
    public void setR(double bk){R=bk>=0?bk:R;}
    public String toString()
    {
        String s="O("+x+", "+y+")\n";
        s+="R= "+R+"\n";
        s+="Chu vi= "+2*R*SOPI+"   Dien tich= "+R*R*SOPI+" (Pi="+SOPI+")\n";
        return s;
    }
}
```

- Cho phép thay đổi và lưu ý vẫn là **static** → cách truy xuất

# Tạo lớp đơn giản Circle5

```
1
2 public class Main {
3     public static void main(String[] args) {
4         Circle5 C5= new Circle5(0,1,1);
5         System.out.println(C5); //in ra thông tin C5
6         Circle5.SOPI=3.14159; //thay đổi số PI
7         System.out.println(C5); //in ra thông tin C5
8         Circle5.SOPI=-3.1; //thay đổi số PI
9         System.out.println(C5);
10    }
11 }
12
```

Console

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 29, 2020, 5:10:42 PM)

O(0.0,1.0)  
R= 1.0  
Chu vi= 6.28 Dien tich= 3.14 (Pi=3.14)

O(0.0,1.0)  
R= 1.0  
Chu vi= 6.28318 Dien tich= 3.14159 (Pi=3.14159)

O(0.0,1.0)  
R= 1.0  
Chu vi= -6.2 Dien tich= -3.1 (Pi=-3.1)

- Truy xuất vùng static  
<tên lớp>.<tên vùng>
- Vì sao/Khi nào khai báo vùng/phương thức là static
- Lưu ý: Số PI bị gán sai!!!
- Cần cải tiến!!!  
➔ phải thay đổi thông qua phương thức!!!

# Tạo lớp đơn giản Circle6

```
public class Circle6 {
    private static double SOPI=3.14;
    private double x;
    private double y;
    private double R;
    Circle6() {x=y=R=0;}
    Circle6(double toadoX, double toadoY, double bankinhR)
        {x=toadoX; y=toadoY; setR(bankinhR);}
    public double getR(){return R;};
    public void setR(double bk){R=bk>=0?bk:R;}
    public static void setSOPI(double newvalue)
    {
        if (newvalue>3.16||newvalue<3.14) return;
        SOPI=newvalue;
        return;
    }
}
```

SOPI: static+private và sử dụng setSOPI để thay đổi giá trị có kiểm soát

# Tạo lớp đơn giản Circle6

```
2 public class Main {
3     public static void main(String[] args) {
4         Circle6 C= new Circle6(0,1,1);
5         System.out.println(C); //in ra thông tin C
6         Circle6.setSOPI(-1); //thay đổi số PI
7         System.out.println(C); //in ra thông tin C
8         Circle6.setSOPI(3.17); //thay đổi số PI
9         System.out.println(C);
10        Circle6.setSOPI(3.141592); //thay đổi số PI
11        System.out.println(C);
12    }
13 }
```

Console Problems

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 29, 2020, 5:20:5)

O(0.0,1.0)  
R= 1.0  
Chu vi= 6.28 Dien tich= 3.14 (Pi=3.14)  
|  
O(0.0,1.0)  
R= 1.0  
Chu vi= 6.28 Dien tich= 3.14 (Pi=3.14)  
|  
O(0.0,1.0)  
R= 1.0  
Chu vi= 6.283184 Dien tich= 3.141592 (Pi=3.141592)

```
public static void setSOPI(double newvalue)
{
    if (newvalue>3.16||newvalue<3.14) return;
    SOPI=newvalue;
    return;
}
```

Nhận xét:

Các lời gọi phương thức setSOPI với giá trị không hợp lệ (-1 và 3.17) → vẫn giữ giá trị Pi đang có

Lời gọi setSOPI với giá trị hợp lệ (3.141592) → thay đổi thực sự

# Circle: Lớp hoàn chỉnh!!

```
public class Circle {  
    private static double SOPI=3.14;  
    private double x;  
    private double y;  
    private double R;  
    Circle() {x=y=R=0;}  
    Circle(double toadoX, double toadoY, double bankinhR)  
        {x=toadoX; y=toadoY; setR(bankinhR);}  
    public double getR(){return R;};  
    public void setR(double bk){R=bk>=0?bk:R;}  
    public static void setSOPI(double newvalue)  
    {    if (newvalue>3.16||newvalue<3.14) return;  
        SOPI=newvalue;  
        return;  
    }  
    public static double getSOPI(){return SOPI;}  
    public double ChuVi(){ return 2*R*SOPI;}  
    public double DienTich(){ return R*R*SOPI;}  
    public String toString()  
    {  
        String s="O("+x+", "+y+")\n";  
        s+="R= "+R+"\n";  
        s+="Chu vi= "+ChuVi()+"   Dien tich= "+DienTich()+"   (Pi="+SOPI+")\n";  
        return s;  
    }  
}
```

SOPI: static → (của) lớp

x,y,R: non-static → (của) đối tượng

# Circle: Lớp hoàn chỉnh!!

```
1
2 public class Main {
3     public static void main(String[] args) {
4         Circle C= new Circle (1,1,1);
5         System.out.println(C);
6         C.setR(10);
7         System.out.println(C);
8         Circle.setSOPI(3.14159);
9         System.out.println("Chu vi= "+C.ChuVi());
10        System.out.println("Dien tich= "+C.DienTich());
11        System.out.println("So Pi dang dung= "+Circle.getSOPI());
12    }
13 }
```

Console Problems

<terminated> Main [Java Application] C:\Program Files\Java\jdk1.8.0\_202\bin\javaw.exe (Feb 29, 2020, 5:40:17 PM)

O(1.0,1.0)

R= 1.0

Chu vi= 6.28 Dien tich= 3.14 (Pi=3.14)

O(1.0,1.0)

R= 10.0

Chu vi= 62.800000000000004 Dien tich= 314.0 (Pi=3.14)

Chu vi= 62.8318

Dien tich= 314.159

So Pi dang dung= 3.14159

```
private static double SOPI=3.14;
public static double getSOPI(){return SOPI;}
public static void setSOPI(double newvalue)
{    if (newvalue>3.16||newvalue<3.14) return;
    SOPI=newvalue;
    return;
}
```

- Lưu ý cách gọi các thuộc tính/phương thức kiểu static

C.setR: non-static → (của) đối tượng

Circle.setSOPI: static → (của) lớp



# Tổng kết

- Lớp: Class
  - Thuộc tính/Vùng tin: Properties
  - Phương thức/Hàm: Methods
  - Hàm xây dựng/khởi tạo
  - Các phương thức getter/setter
- Kiểu dữ liệu (Data Type)/Biến(Variable)/Hàm(Function)/Vùng tin(Field)
- Lớp (Class)/Đối tượng (Object)/Phương thức(Method)/Thuộc tính(Property)
- Phạm vi truy cập: public/private/protected
- Vùng nhớ cấp phát: static/non-static
- Khai báo đối tượng – objects (tức là khai báo biến)
- Cách gọi/sử dụng/truy xuất các vùng/phương thức
  - <tên đối tượng>.<vùng> <tên đối tượng>.<phương thức>(...) nếu **non-static** và public
  - <tên lớp>.<vùng> <tên lớp>.<phương thức>(...) nếu là **static** và public (hoặc như cách trên)

## Phương thức

- Các phương thức thể hiện **hành vi** thực thể.
- Các loại phương thức

