



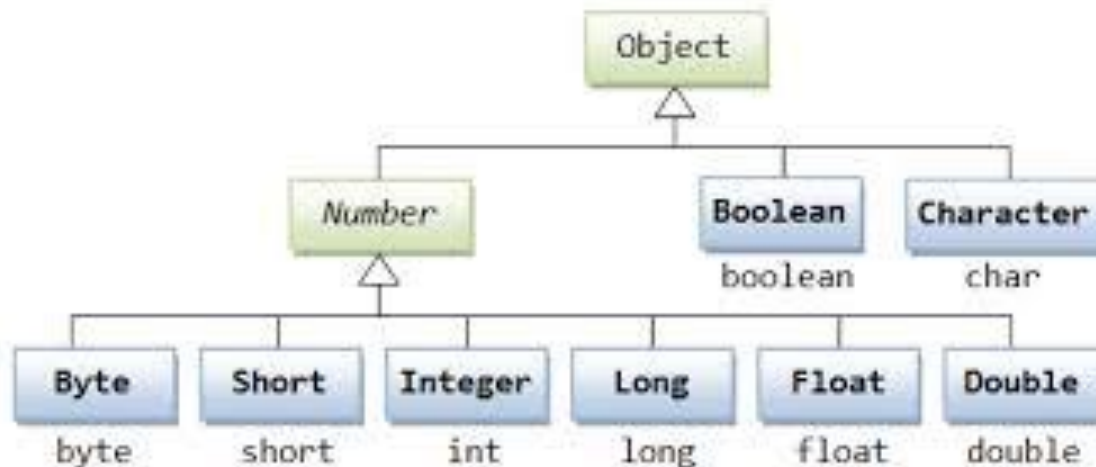
LẬP TRÌNH JAVA 2

BÀI 2: LỚP NÂNG CAO

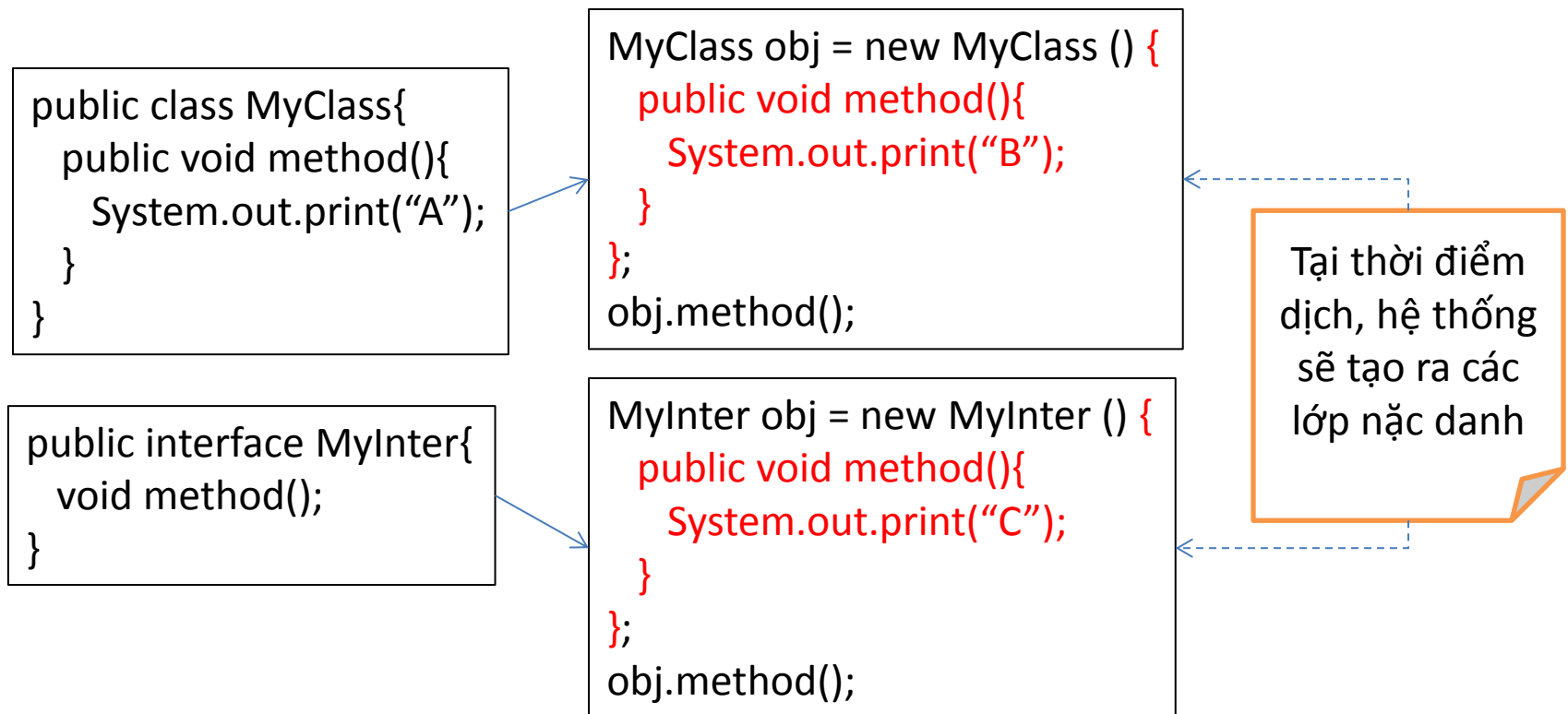
PHẦN 1

- ☐ Tìm hiểu lớp Object
- ☐ Giải thích sâu hơn về hàm tạo
- ☐ Sử dụng static để khai báo field, method và class
- ☐ Sử dụng final khai báo field, method và class
- ☐ Thiết kế giao diện và xử lý sự kiện click

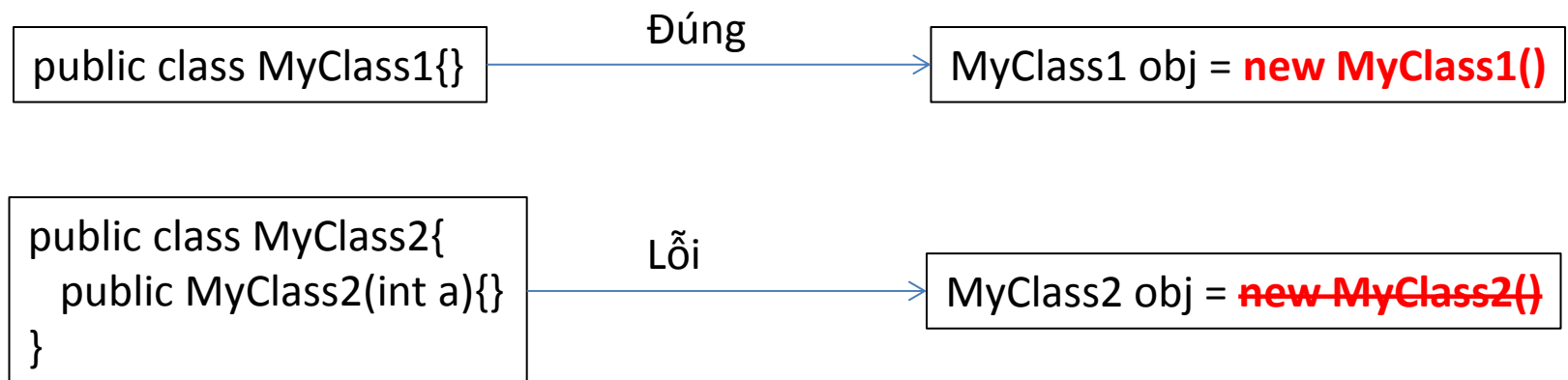
- ❑ Lớp object là lớp ở mức cao nhất trong phân cấp kế thừa của Java.
- ❑ Khi khai báo một lớp mà không kế thừa từ một lớp khác thì mặc định kế thừa lớp Object
- ❑ Như vậy mọi lớp đều có lớp cha, chỉ duy nhất một lớp không có cha là Object



- ❑ Lớp nặc danh là lớp không tên
- ❑ Lớp nặc danh được tạo ra khi tạo một đối tượng từ một lớp hoặc một interface nào đó mà bạn muốn override một vài phương thức của lớp đó



- ❑ Nếu một lớp không khai báo hàm tạo thì Java tự động cung cấp hàm tạo mặc định (không tham số)
- ❑ Khi đã khai báo các hàm tạo cho một lớp thì chỉ được phép sử dụng các hàm tạo này để tạo đối tượng



- ❑ Trong hàm tạo muốn gọi hàm tạo khác cùng lớp thì sử dụng this()
- ❑ Lời gọi hàm tạo phải là lệnh đầu tiên

Đúng

```
public class MyClass{  
    public MyClass(int a, int b){}  
    public MyClass(){  
        this(5, 7);  
        System.out.print("Hello");  
    }  
}
```

Lỗi

```
public class MyClass{  
    public MyClass(int a, int b){}  
    public MyClass(){  
        System.out.print("Hello");  
        this(5, 7);  
    }  
}
```

- ❑ Trong hàm tạo muốn gọi hàm tạo của lớp cha thì sử dụng `super()`
- ❑ Nếu trong hàm tạo không gọi hàm tạo khác thì nó tự gọi hàm tạo không tham số của lớp cha `super()`

```
public class Parent{  
    public Parent(int a, int b){}  
}
```

```
public class Child extends Parent{  
    public Child(){  
        super(5, 7);  
    }  
}
```

```
public class Parent{  
    public Parent(){}  
}
```

```
public class Child extends Parent{  
    public Child(){  
        super();  
    }  
}
```

Nếu chúng ta không gọi `super()` thì hệ thống cũng tự gọi

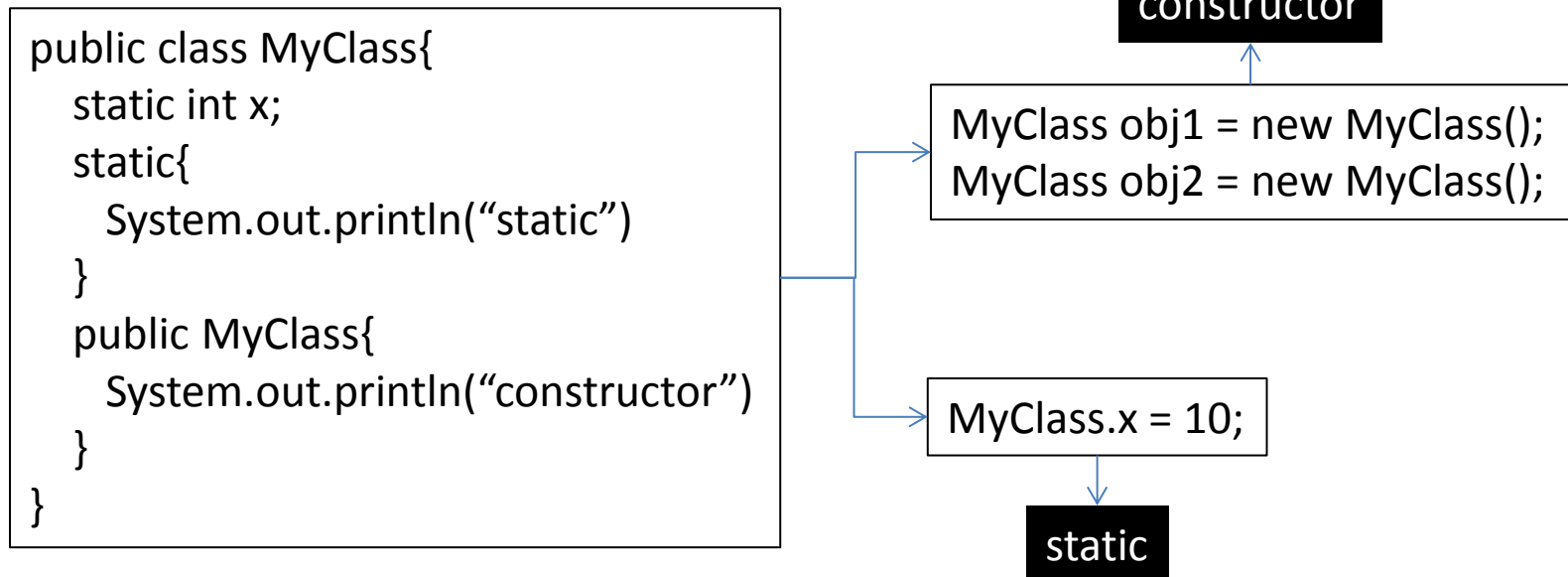
□ Từ khóa **static** được sử dụng để khai báo cho

- ❖ Khối
- ❖ Field
- ❖ Method
- ❖ Inner Class

```
public class MyClass{  
    static int x;  
    static{  
        System.out.println("block")  
    }  
    public static void method(){  
        System.out.println("method")  
    }  
    static class MyInnerClass{  
    }  
}
```


❑ Khối static {} sẽ **chạy trước** khi

- ❖ Tạo đối tượng
- ❖ Truy xuất bất kỳ thành viên tĩnh



Chú ý: khối static chỉ chạy một lần đầu tiên duy nhất

- ❑ Thành viên tĩnh của lớp được sử dụng **độc lập** với các đối tượng được tạo ra từ lớp đó.
- ❑ Có thể truy cập đến một thành viên tĩnh thông qua **tên lớp** mà không cần tham chiếu đến một đối tượng cụ thể
- ❑ Trường static là dữ liệu **dùng chung** cho tất cả các đối tượng được tạo ra từ lớp đó.

```
public class MyClass{  
    static int field;  
    public static void method()  
}
```

```
MyClass.field = 10;  
MyClass.method();  
  
MyClass obj = new MyClass();  
obj.field = 100;
```

MyClass.field và obj.field là 1

- ❑ Trong khối và phương thức tĩnh **chỉ được truy cập đến các thành viên tĩnh** khác mà không được phép truy cập đến thành viên thông thường của class

```
public class MyClass{
    static int x;
    int y;
    static{
        x = 5;
    }
    public static void method{
        x = 5
    }
}
```

Đúng

```
public class MyClass{
    static int x;
    int y;
    static{
        y = 5;
    }
    public static void method{
        y = 5
    }
}
```

Sai

```
public class MyClass{  
    static public int X = 100;  
    static{  
        X+=100;  
    }  
    static public void method(){  
        X+=200;  
    }  
}
```

```
MyClass o = new MyClass();  
o.X += 300;  
MyClass.X += 500;  
MyClass.method()
```

MyClass.X, o.X có
giá trị là bao nhiêu



DEMO

Hiện thực hóa slide trước.
Giải thích kết quả



- ❑ Lớp nội là lớp được khai báo bên trong một lớp khác
- ❑ Có hai loại
 - ❖ Lớp nội tĩnh
 - ❖ Lớp nội thông thường

```
public class MyClass{  
    static public class MyInner1{}  
    public class MyInner2{}  
}
```

Sử dụng lớp nội

```
MyClass.MyInner1 x = new MyClass.MyInner1();  
MyClass.MyInner2 y = new MyClass().new MyInner2();
```

❑ Trong Java có 3 loại hằng

- ❖ Lớp hằng là lớp không cho phép thừa kế
- ❖ Phương thức hằng là phương thức không cho phép ghi đè
- ❖ Biến hằng là biến không cho phép thay đổi giá trị

❑ Sử dụng từ khóa **final** để định nghĩa hằng

```
final public class MyFinalClass{...}
```

```
public class MyClass{  
    final public double PI = 3.14  
    final public void method(){...}  
}
```

A

```
final public class Parent{...}
```

```
public class Child extends Parent{  
...  
}
```

B

```
public class MyClass{  
    final int PI = 3.14;  
    public void method(){  
        PI = 3.1475;  
    }  
}
```

C

```
public class Parent{  
    final public void method(){...}  
}
```

```
public class Child extends Parent{  
    public void method(){...}  
}
```

D

```
public class Parent{  
    public void method(){...}  
}
```

```
public class Child extends Parent{  
    public void method(){...}  
}
```



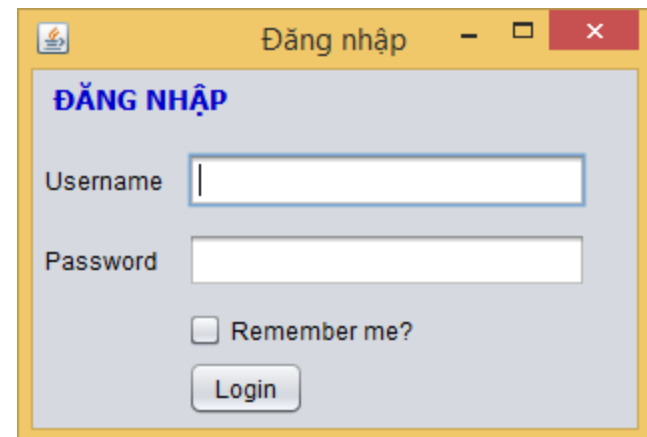

Java™

LẬP TRÌNH JAVA 2

BÀI 2: LỚP NÂNG CAO

PHẦN 2

- ❑ Với ứng dụng console bạn sử dụng Scanner để nhập dữ liệu từ người dùng và System.out.print() để xuất thông tin cho người dùng xem
- ❑ Giao diện giúp ứng dụng giao tiếp với người dùng trở nên thân thiện hơn.
- ❑ Ví dụ sau là một cửa sổ ứng dụng. Thông qua giao diện này bạn có thể đọc dữ liệu và hiển thị thông tin cho người dùng.



Đăng nhập

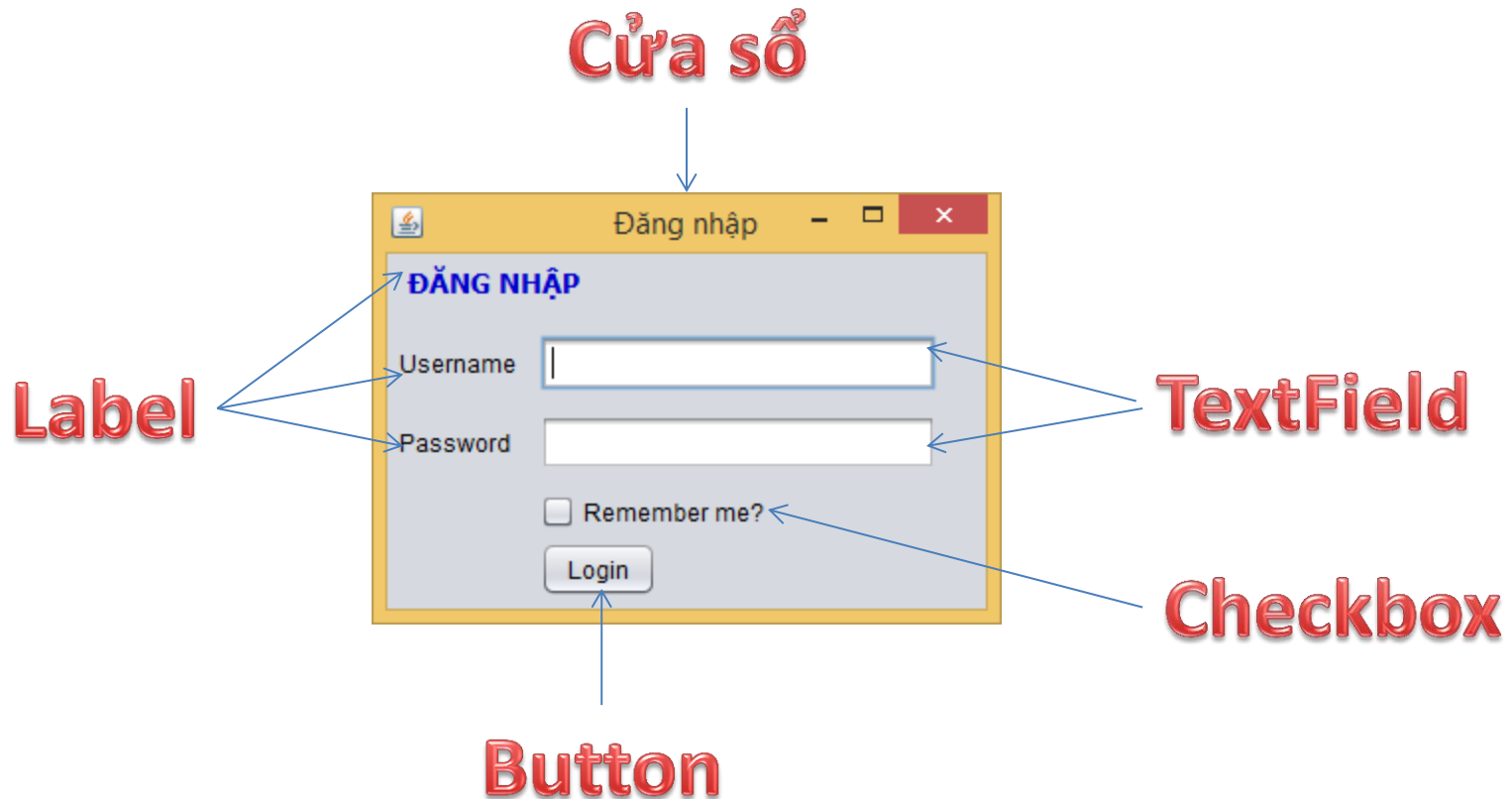
ĐĂNG NHẬP


Username

Password

☐ Remember me?

Login





Đăng nhập

ĐĂNG NHẬP

Username

Password

☐ Remember me?

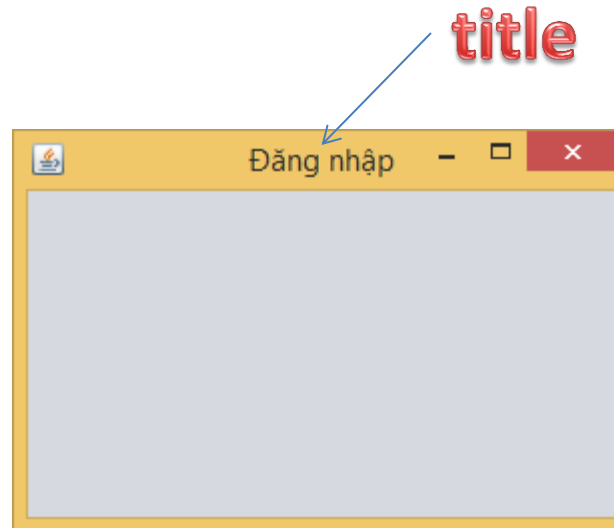
Login

DEMO

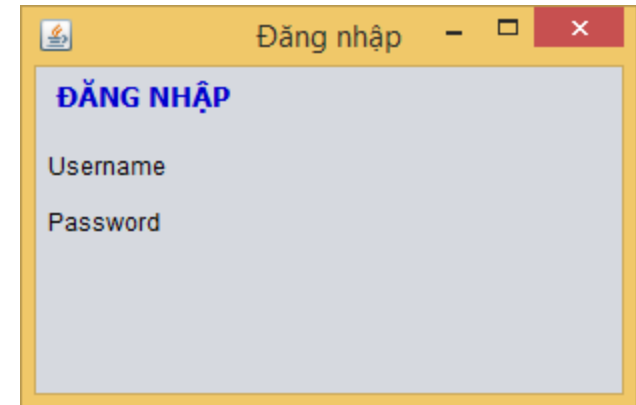
Thiết kế form đăng nhập



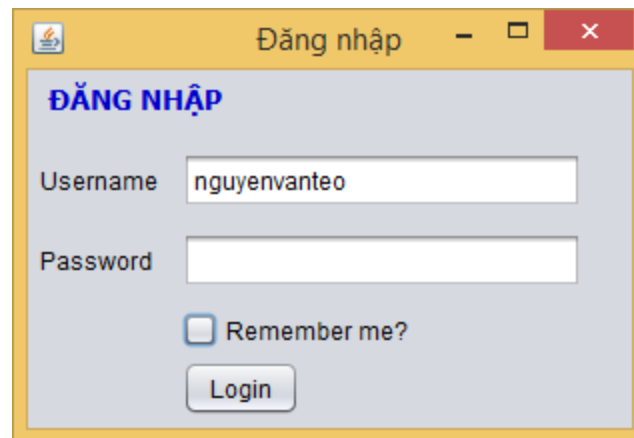
- ❑ Cửa sổ được tạo ra từ lớp JFrame
- ❑ Tiêu đề của cửa sổ được thiết lập bởi thuộc tính title.
- ❑ Để đọc/ghi title sử dụng getTitle()/setTitle()



- ❑ Nhãn được sử dụng để trình bày thông tin cho người dùng.
- ❑ Thuộc tính thường được sử dụng
 - ❖ Text: nội dung
 - ❖ Font: font chữ
 - ❖ Foreground: màu chữ
 - ❖ Background: màu nền
- ❑ Sử dụng các phương thức getter/setter để đọc ghi các thuộc tính



- ❑ TextField là thành phần giao diện tạo ô nhập.
- ❑ Có thể sử dụng TextField để tiếp nhận hoặc trình bày dữ liệu với người dùng
- ❑ Thuộc tính thường dùng
 - ❖ Text: nội dung ô nhập
 - ❖ Editable: trạng thái cho phép nhập hay không



Đăng nhập

ĐĂNG NHẬP

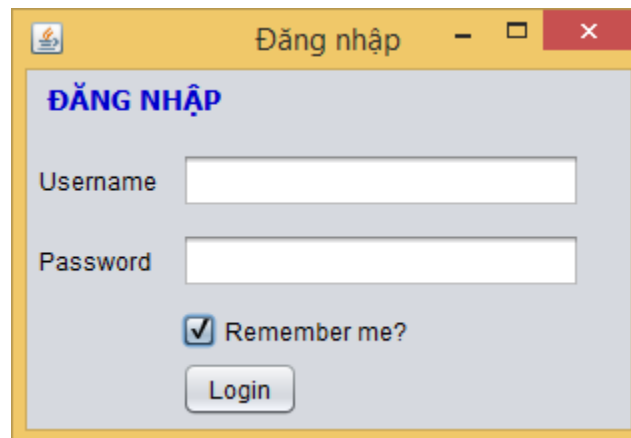
Username: nguyenvanteo

Password:

☐ Remember me?

Login

- ❑ Checkbox là thành phần giao diện được sử dụng để nhập và trình bày trạng thái (true hoặc false)
- ❑ Thuộc tính `selected` cho biết checkbox có được chọn hay không
- ❑ Sử dụng `isSelected()` và `setSelected()` để đọc ghi trạng thái của checkbox
- ❑ Sử dụng thuộc tính `text` để làm việc với nội dung checkbox.



The image shows a Java Swing window titled "Đăng nhập" (Login). Inside the window, there is a label "ĐĂNG NHẬP" in blue. Below it are two text input fields: "Username" and "Password". Under the "Password" field is a checked checkbox with the label "Remember me?". At the bottom of the form is a "Login" button.

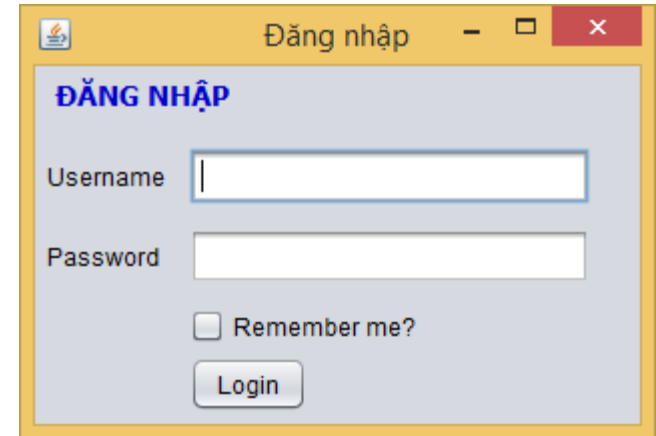
- ❑ Button là thành phần giao diện tạo ra nút bấm để người dùng click vào khi muốn thực hiện một công việc nào đó.
- ❑ Thuộc tính thường dùng là text để làm việc với nội dung nút.
- ❑ Sự kiện quan trọng nhất của nút là Action. Để viết mã cho nút bạn chỉ cần nhấp đúp vào nút và viết mã thích hợp.

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    // đặt mã ở đây  
}
```

- ❑ Các thành phần giao diện cần được đặt tên phù hợp để dễ nhớ, dễ sử dụng trong quá trình viết mã.
- ❑ Sau đây là vài quy ước đặt tên:
 - ❖ Label: **lbl**Xxx
 - ❖ TextField: **txt**Xxx
 - ❖ Button: **btn**Xxx
 - ❖ CheckBox: **chk**Xxx
 - ❖ RadioButton: **rdo**Xxx
 - ❖ ComboBox: **cbo**Xxx
 - ❖ Table: **tbl**Xxx

❑ Đặt tên cho các thành phần giao diện

- ❖ lblTitle
- ❖ lblUsername
- ❖ lblPassword
- ❖ txtUsername
- ❖ txtPassword
- ❖ chkRemem
- ❖ btnLogin



❑ Viết mã cho nút đăng nhập

```
private void btnLoginActionPerformed(java.awt.event.ActionEvent evt) {  
    String fullname = txtUsername.getText();  
    JOptionPane.showMessageDialog(this, "Chào " + fullname);  
    txtPassword.setText("I love you");  
    chkRemem.setSelected(true);  
}
```

- ❑ Tìm hiểu lớp Object
- ❑ Giải thích sâu hơn về hàm tạo
- ❑ Sử dụng static để khai báo field, method và class
- ❑ Sử dụng final khai báo field, method và class
- ❑ Thiết kế giao diện và xử lý sự kiện click

