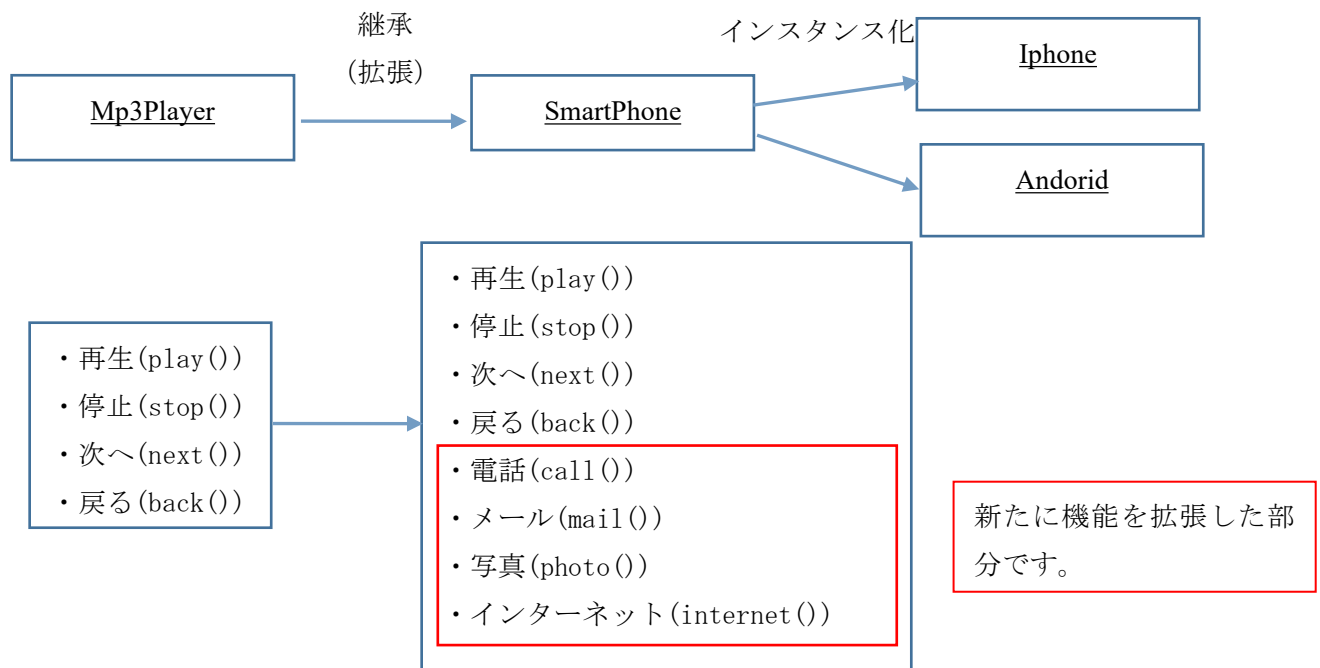


### <継承>

Java で大きなプログラムを書き始めると、以前作ったクラスと似かよったクラスを作る場合があります。このような時にすでにあるクラスを元に、必要な機能だけを追加する形で新しいクラスを作ること継承といいます。次の例を見てみましょう。



Mp3Player が元々持っている機能を SmartPhone にそのまま継承をして、新たに 4 つの機能を追加しました。そこから Iphone や Andorid などのクラスを作成することができます。

### (演習①)

Java プロジェクト「Iphone」を作成しましょう。

以下の Mp3Player クラスを作成後、プログラミングしましょう。

```
public class Mp3Player{
    public void play() {
        System.out.println("再生");
    }
    public void stop() {
        System.out.println("停止");
    }
    public void next() {
        System.out.println("次へ");
    }
}
```

メソッド名はそれぞれ  
play(),stop(),next()

```
}  
public void back() {  
    System.out.println("戻る");  
}  
}
```

---

### (演習②)

ここから **Mp3Player** クラスを元にして **SmartPhone** クラスへ継承します。

以下の **SmartPhone** クラスを作成後、プログラミングしましょう。

---

```
public class SmartPhone extends Mp3Player {  
    public void call() {  
        System.out.println("電話");  
    }  
    public void mail() {  
        System.out.println("メール");  
    }  
    public void photo() {  
        System.out.println("写真");  
    }  
    public void internet() {  
        System.out.println("インターネット");  
    }  
}
```

`extends` を使うことによって、  
**Mp3Player** から **SmartPhone** へ継  
承するという意味。

---

今回は **Mp3Player** を元にして作っていきました。この元のクラスをスーパークラス（親クラス）と呼び、継承してできた **SmartPhone** のことをサブクラス（子クラス）と呼びます。

```
public class SmartPhone extends Mp3Player
```

サブクラス  
(子クラス)

継承(拡張)

スーパークラス  
(親クラス)

---

### (演習③)

以下の Iphone クラスを作成後、プログラミングして実行してみましょう。

```
public class Iphone{  
    public static void main(String[] args){  
        SmartPhone iphone = new SmartPhone();  
        iphone.play();  
        iphone.stop();  
        iphone.next();  
        iphone.back();  
        iphone.call();  
        iphone.mail();  
        iphone.photo();  
        iphone.internet();  
    }  
}
```

変数名.(ドット) メソッド名：  
各変数に格納されたインスタンスが持つメソッドを呼び出す

また、スーパークラス（親クラス）の機能をサブクラス（子クラス）で上書くこともできます。この上書きのことを**オーバーライド**と言います。オーバーロードとは違うので注意。

### (演習④)

オーバーライドをしてプログラミングを実行してみましょう。

オーバーライドする場合は、サブクラスにスーパークラスのメソッドも書きます。

```
public class SmartPhone extends Mp3Player{  
    public void play() {  
        System.out.println("再生 (ハイレゾ機能)");  
    }  
    public void stop() {  
        System.out.println("停止 (ハイレゾ機能)");  
    }  
    public void next() {  
        System.out.println("次へ (ハイレゾ機能)");  
    }  
    public void back() {  
        System.out.println("戻る (ハイレゾ機能)");  
    }  
}
```

オーバーライドの際

メソッド名は変えてはいけない  
(スーパークラスと同じ)

処理だけを変える

オーバーライドした  
プログラム

→Mp3Player の同名メ  
ソッドは無視されて、  
こちらが実行される。

```
}  
public void call() {  
    System.out.println("電話");  
}  
public void mail() {  
    System.out.println("メール");  
}  
public void photo() {  
    System.out.println("写真");  
}  
public void internet() {  
    System.out.println("インターネット");  
}  
}
```

---

#### (演習⑤)

Iphone プログラムをもとにして Android プログラムを作ってみましょう。  
構成は以下の通りとします。

