Object Oriented Programming

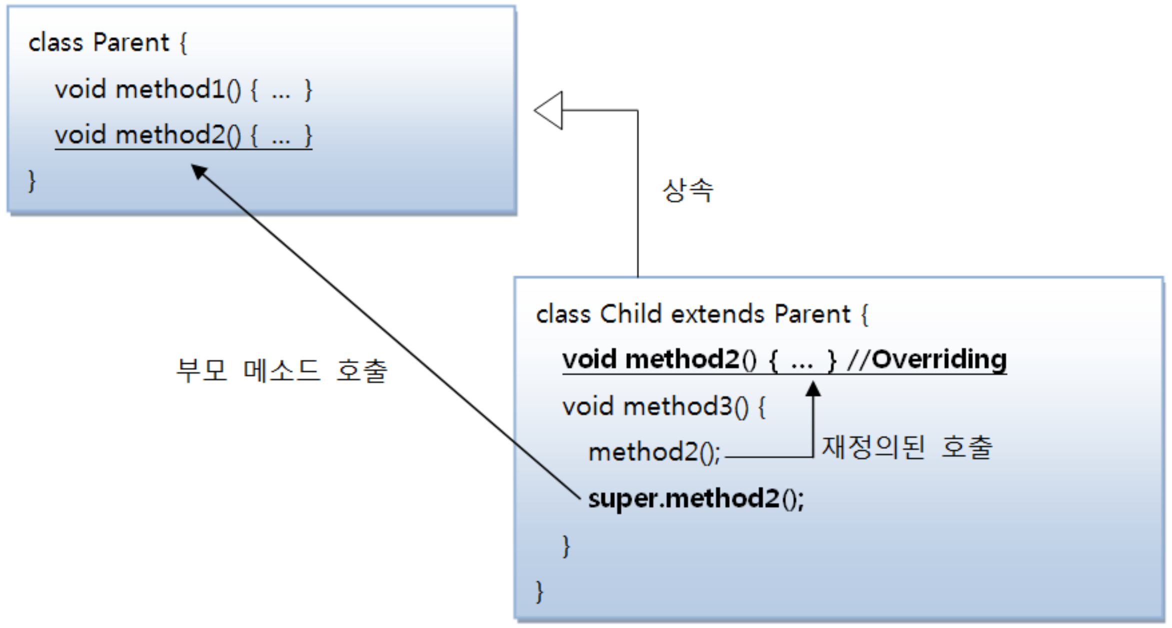
Kookmin University
Department of Computer Science

Method Overriding

- A child class can can use/access parent class method (protected methods)
- Overriding allows rewriting a method that is inherited from a parent class
- If a overridden method is called from an object of child class, the method declaration by a child class is executed

Calling Parent Method After Override

- After overriding a method from a child class, to access the parent's method, we can use super



```
class Parent {
  void method1() { … }
  void method2() { … }
}
```

```
class Child extends Parent {
  void method2() { … } //Overriding
  void method3() {
    method2();
    super.method2();
  }
}
```

이것이자바다 P. 296

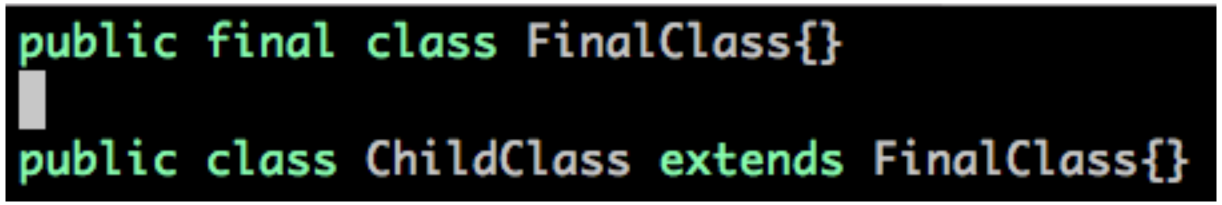
- Example : SmartWatch.java

Overload VS. Override

- Overload allows to define a new method with a same name but with different arguments
- Override allows to redefine a method body that is already defined in a parent class

final Class

- Remember what final keywords mean for a class field
  - The field cannot be modified after initialization in a constructor
- Then, what final means for a class and method?
- final class cannot be inherited
  - EX: public final class ClassName{}
- When to declare a class as final
  - Unless you take extra care to design a class to allow for extension and document how methods may be overridden, …
  - It is a defensive method to prohibit unexpected behavior

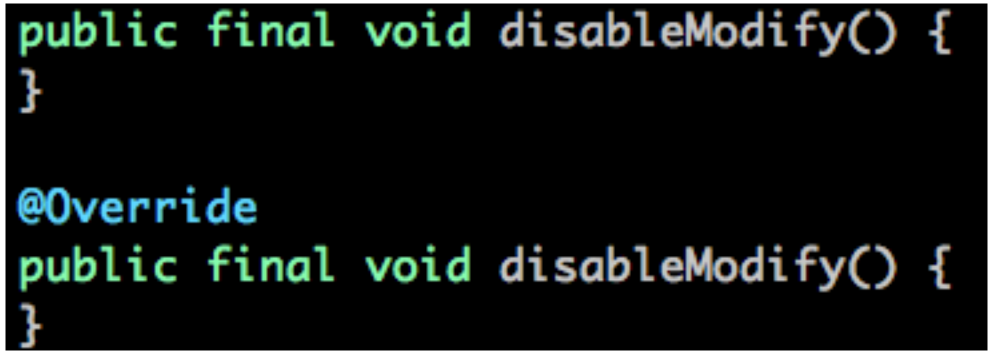


```
public final class FinalClass{}

public class ChildClass extends FinalClass{}
```

Incorrect usage

final Method

- final method cannot be overriden
  - A child class cannot override a method implemented in the parent class



```
public final void disableModify() {
}

@Override
public final void disableModify() {
}
```

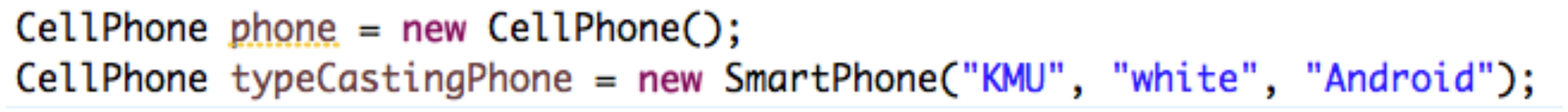
Incorrect usage

Polymorphism and Type Casting

- Polymorphism (다형성) allows allocation of multiple types in another type
  - It is tightly connected with inheritance and parent-child relationship
- Type casting for a class
  - A child class object can be automatically converted (type casting) to a parent object type
  - A parent class object cannot be converted to a child class type
  - Remember CellPhone class and SmartPhone class.

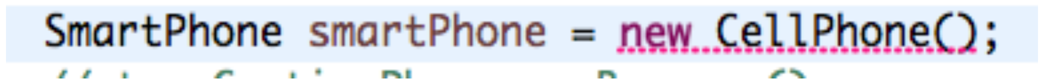
Class Type Casting Example

- A child class can use a type of the parent class



```java
CellPhone phone = new CellPhone();
CellPhone typeCastingPhone = new SmartPhone("KMU", "white", "Android");
```

- As SmartPhone class inherits from a CellPhone class
  - SmartPhone object is also a CellPhone object
  - However, not all CellPhone objects are not SmartPhone objects

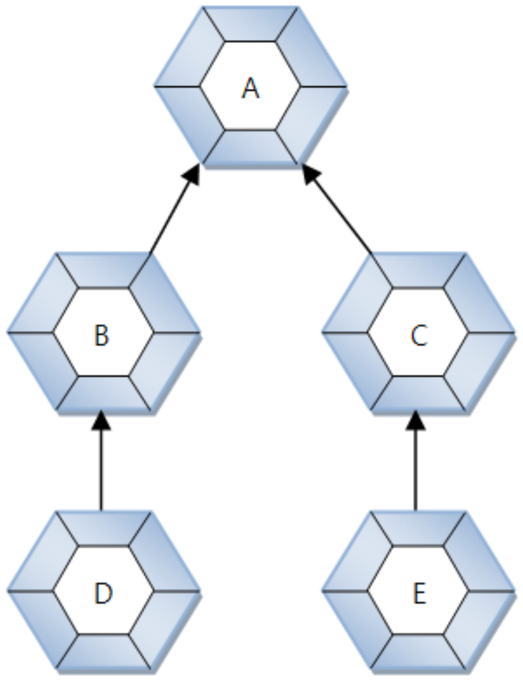


```java
SmartPhone smartPhone = new CellPhone();
```

    - Type Mismatch error: Cannot convert from CellPhone to SmartPhone
- Similar principles applies to
  - Animal and Cat/Dog

Automatic Type Conversion

- After a class type conversion to a parent class type, only the members from the parent class can be accessed



```
B b = new B();
C c = new C();
D d = new D();
E e = new E();
```

```
A a1 = b;  (가능)
A a2 = c;  (가능)
A a3 = d;  (가능)
A a4 = e;  (가능)

B b1 = d;  (가능)
C c1 = e;  (가능)

B b3 = e;  (불가능)
C c2 = d;  (불가능)
```

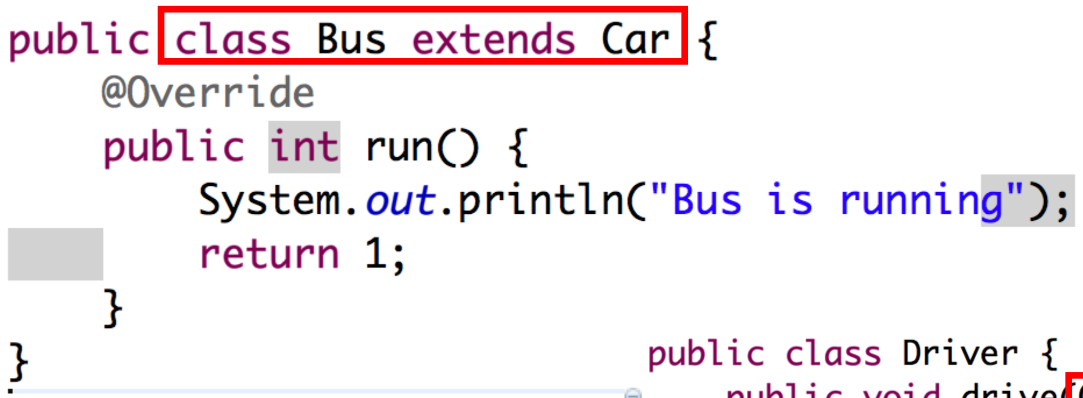
이것이 자바다 P. 308

Field Polymorphism

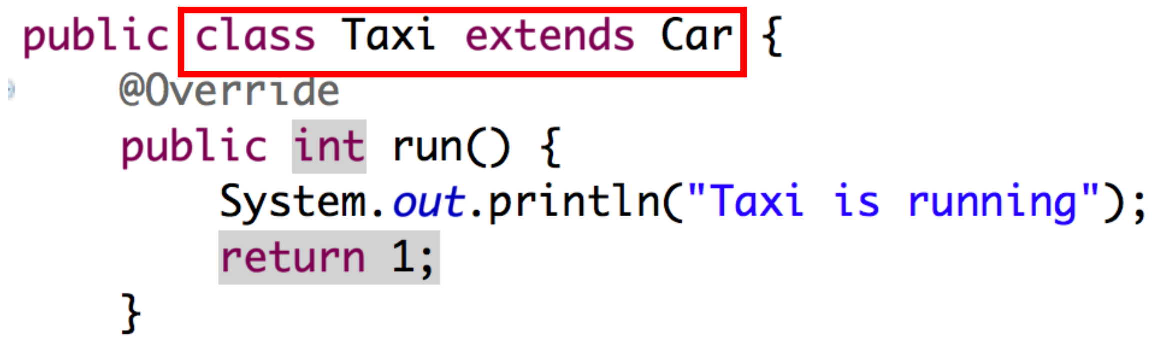
- Field object can have multiple forms if they share a same parent class type
- Imagine a car that has four tires. The tires do not have to manufactures from a same class (or company) assuming all tires meet standards
- Example : Car

Polymorphism for Method Arguments

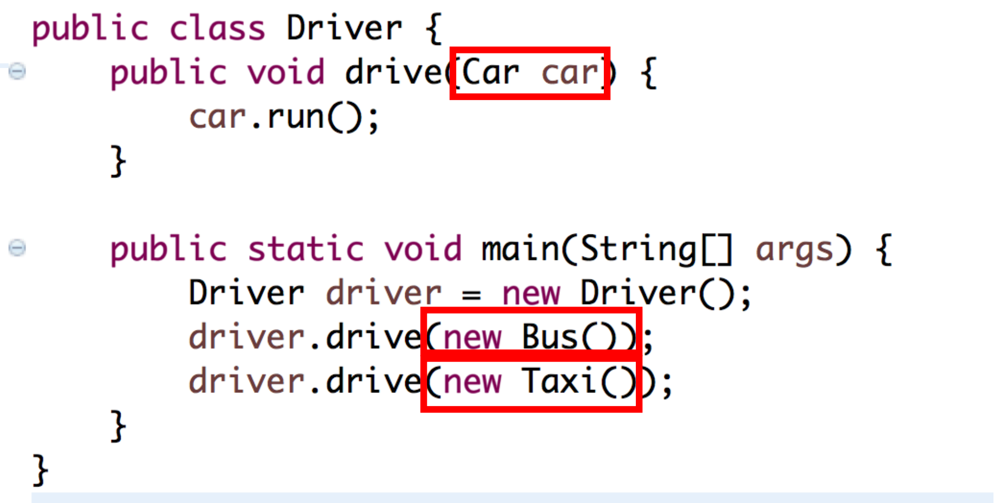
- Polymorphism applies well to class fields
- Polymorphism can also applies to method arguments



```
public class Bus extends Car {
    @Override
    public int run() {
        System.out.println("Bus is running");
        return 1;
    }
}
```



```
public class Taxi extends Car {
    @Override
    public int run() {
        System.out.println("Taxi is running");
        return 1;
    }
```



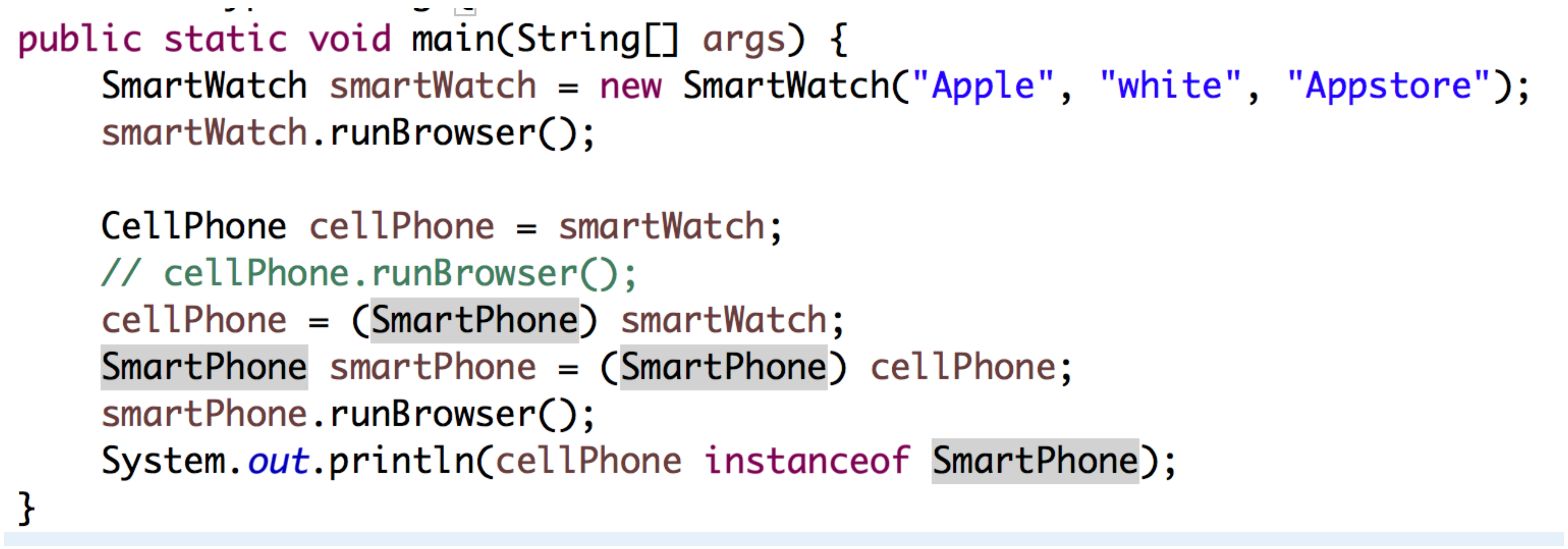
```
public class Driver {
    public void drive(Car car) {
        car.run();
    }

    public static void main(String[] args) {
        Driver driver = new Driver();
        driver.drive(new Bus());
        driver.drive(new Taxi());
    }
}
```

Type Casting from Parent to Child Class

- In case an object created from a child class definition is declared with a parent class type, it can be converted to a child class type that generally has more fields/methods
  - EX: CellPhone cellPhone = new SmartPhone();
  - The cellPhone can be converted to SmartPhone() object
- To type casting to a child class, use parenthesis with a child class name
  - EX: ParentClass parentClass = new ChildClass();
  - ChildClass childClass = (ChildClass)parentClass ;
- isinstanceof allows to check if a given class is instance of an object

TypeCasting Example

- TypeCasting.java



```java
public static void main(String[] args) {
    SmartWatch smartWatch = new SmartWatch("Apple", "white", "Appstore");
    smartWatch.runBrowser();

    CellPhone cellPhone = smartWatch;
    // cellPhone.runBrowser();
    cellPhone = (SmartPhone) smartWatch;
    SmartPhone smartPhone = (SmartPhone) cellPhone;
    smartPhone.runBrowser();
    System.out.println(cellPhone instanceof SmartPhone);
}
```

References

- 이것이자바다 – 한빛미디어 2015