

# 익명 객체

# 익명 객체

---

## ❖ 익명 객체

- 이름이 없는 객체

```
class Child extends Parent { } ----- 자식 클래스 선언
```

```
class A {  
    Parent field = new Child(); ----- 필드에 자식 객체를 대입  
    void method() {  
        Parent localVar = new Child(); ----- 로컬 변수에 자식 객체를 대입  
    }  
}
```

# 익명 객체

---

## ❖ 익명 객체: 이름이 없는 객체

```
부모클래스 [필드|변수] = new 부모클래스(매개값, ... ) {  
    // 필드  
    // 메서드  
};
```

- 익명 객체는 단독 생성 불가
  - 클래스 상속하거나 인터페이스 구현해야만 생성 가능
- 사용 위치
  - 필드의 초기값, 로컬 변수의 초기값, 매개변수의 매개값으로 주로 대입
  - UI 이벤트 처리 객체나, 스레드 객체를 간편하게 생성할 목적으로 주로 활용

## 익명 객체

### ❖ 익명 자식 객체 생성 - 초기값 설정에 주목

```
class A {  
    Parent field = new Parent() {  
        int childField;  
        void childMethod() { }  
        @Override  
        void parentMethod() { }  
    };  
}
```

A 클래스의 필드 선언

Parent 의 메소드를 오버라이딩

## 익명 객체

### ❖ 익명 자식 객체 생성 - 초기값 설정에 주목

```
class A {  
    void method1(Parent parent) { }  
  
    void method2() {
```

```
        method1(  
            new Parent() {  
                int childField;  
                void childMethod() { }  
                @Override  
                void parentMethod() { }  
            }  
        );  
    }  
}
```

method1() 메소드 호출

method1()의 매개값으로  
익명 자식 객체를 대입

## 익명 객체

### ❖ 익명 객체에 새롭게 정의된 필드와 메소드

- 익명 객체 내부에서만 사용
- 외부에서는 익명 객체의 필드와 메소드에 접근할 수 없음
  - 이유: 익명 객체는 부모 타입 변수에 대입되므로 부모 타입에 선언된 것만 사용 가능

```
class A {  
    Parent field = new Parent() {  
        int childField; ←-----  
        void childMethod() { } ←-----  
        @Override  
        void parentMethod() { ←-----  
            childField = 3;  
            childMethod();  
        }  
    };  
  
    void method() {  
        field.childField = 3; -----  
        field.childMethod(); -----  
        field.parentMethod(); -----  
    }  
}
```

---

## ❖ 부모 클래스: Person.java

```
public class Person {  
    void wake() {  
        System.out.println("7시에 일어납니다.");  
    }  
}
```

## 익명 객체

---

### ❖ 익명 자식 객체 생성: Anonymous.java

```
public class Anonymous {  
    // 필드 초기값으로 대입  
    Person field = new Person() {  
        void work() {  
            System.out.println("출근합니다.");  
        }  
  
        @Override  
        void wake() {  
            System.out.println("6시에 일어납니다.");  
            work();  
        }  
    };  
};
```



# 익명 객체

---

## ❖ 익명 자식 객체 생성: Anonymous.java

```
void method1() {  
    // 로컬변수값으로 대입  
    Person localVar = new Person() {  
        void walk() {  
            System.out.println("산책합니다.");  
        }  
  
        @Override  
        void wake() {  
            System.out.println("7시에 일어납니다.");  
            walk();  
        }  
    };  
    // 로컬변수 사용  
    localVar.wake();  
}  
  
void method2(Person person) {  
    person.wake();  
}  
}
```

## ❖ 익명 자식 객체 생성: AnonymousExample.java

```
public class AnonymousExample {
    public static void main(String[] args) {
        Anonymous anony = new Anonymous();
        // 익명 객체 필드 사용
        anony.field.wake();
        // 익명 객체 로컬변수 사용
        anony.method1();
        // 익명 객체 매개값 사용
        anony.method2(new Person() {
            void study() {
                System.out.println("공부합니다.");
            }

            @Override
            void wake() {
                System.out.println("8시에 일어납니다.");
                study();
            }
        });
    }
}
```

## 익명 객체

### ❖ 익명 구현 객체 생성

```
class TV implements RemoteControl { }
```

```
class A {
```

```
    RemoteControl field = new TV(); ----- 필드에 구현 객체를 대입
```

```
    void method() {
```

```
        RemoteControl localVar = new TV(); --- 로컬 변수에 구현 객체를 대입
```

```
    }
```

```
}
```

```
인터페이스 [필드|변수] = new 인터페이스() {
```

```
    //인터페이스에 선언된 추상 메소드의 실제 메소드 선언
```

```
    //필드
```

```
    //메소드
```

```
};
```

# 익명 객체

## ❖ 익명 구현 객체 생성

```
class A {  
    RemoteControl field = new RemoteControl() {           클래스 A의 필드 선언  
        @Override                                         RemoteControl 인터페이스의  
        void turnOn() { }                                추상 메소드에 대한 실제 메소드  
    };  
}  
  
void method() {  
    RemoteControl localVar = new RemoteControl() {       로컬 변수 선언  
        @Override                                         RemoteControl 인터페이스의  
        void turnOn() { }                                추상 메소드에 대한 실제 메소드  
    };  
}
```

# 익명 객체

## ❖ 익명 구현 객체 생성

```
class A {  
    void method1(RemoteControl rc) { }
```

```
    void method2() {
```

```
        method1(  
            new RemoteControl() {  
                @Override  
                void turnOn() { }            }  
        );  
    }  
}
```

method1() 메소드 호출

method1()의 매개값으로  
익명 구현 객체를 대입

### ❖ 인터페이스: RemoteControl.java

```
public interface RemoteControl {  
    public void turnOn();  
    public void turnOff();  
}
```

## 익명 객체

---

### ❖ 익명 구현 클래스와 객체 생성: Anonymous.java

```
public class Anonymous {  
    // 필드 초기값으로 대입  
    RemoteControl field = new RemoteControl() {  
        @Override  
        public void turnOn() {  
            System.out.println("TV를 켭니다.");  
        }  
  
        @Override  
        public void turnOff() {  
            System.out.println("TV를 끕니다.");  
        }  
    };  
}
```

## 익명 객체

### ❖ 익명 구현 클래스와 객체 생성: Anonymous.java

```
void method1() {  
    // 로컬변수값으로 대입  
    RemoteControl localVar = new RemoteControl() {  
        @Override  
        public void turnOn() {  
            System.out.println("Audio를 켭니다.");  
        }  
  
        @Override  
        public void turnOff() {  
            System.out.println("Audio를 끕니다.");  
        }  
    };  
    // 로컬변수 사용  
    localVar.turnOn();  
}  
  
void method2(RemoteControl rc) {  
    rc.turnOn();  
}  
}
```



## 익명 객체

### ❖ 익명 구현 클래스와 객체 생성: RemoteControl.java

```
public class AnonymousExample {
    public static void main(String[] args) {
        Anonymous anony = new Anonymous();
        // 익명 객체 필드 사용
        anony.field.turnOn();
        // 익명 객체 로컬변수 사용
        anony.method1();
        // 익명 객체 매개값 사용
        anony.method2(new RemoteControl() {
            @Override
            public void turnOn() {
                System.out.println("SmartTV를 켭니다.");
            }

            @Override
            public void turnOff() {
                System.out.println("SmartTV를 끕니다.");
            }
        });
    }
}
```

## 익명 객체

---

### ❖ UI 클래스: Button.java

```
public class Button {
    OnClickListener listener;    // 인터페이스 타입 필드

    // 매개변수의 다형성
    void setOnClickListener(OnClickListener listener) {
        this.listener = listener;
    }

    void touch() {
        listener.onClick();    // 구현 클래스의 onClick() 메서드 호출
    }

    // 중첩 인터페이스
    interface OnClickListener {
        void onClick();
    }
}
```

# 익명 객체

---

## ❖ UI 클래스: Window.java

```
public class Window {
    Button button1 = new Button();
    Button button2 = new Button();

    // 필드 초기값으로 대입
    Button.OnClickListener listener = new Button.OnClickListener() {
        @Override
        public void onClick() {
            System.out.println("전화를 겁니다.");
        }
    };

    Window() {
        button1.setOnClickListener(listener);

        // 매개값으로 대입
        button2.setOnClickListener(new Button.OnClickListener() {
            @Override
            public void onClick() {
                System.out.println("메시지를 보냅니다.");
            }
        });
    }
}
```

# 익명 객체

---

## ❖ 실행 클래스: Main.java

```
public class Main {  
    public static void main(String[] args) {  
        Window w = new Window();  
        w.button1.touch();  
        w.button2.touch();  
    }  
}
```

## 익명 객체

### ❖ 익명 객체의 로컬 변수 사용

```
void outMethod(final int arg1, int arg2) {  
    final int var1 = 1;  
    int var2 = 2;
```

```
    인터페이스 변수 = new 인터페이스() {  
        void method() {  
            int result = arg1+arg2+var1+var2;  
        }  
    };  
}
```



```
인터페이스 변수 = new 인터페이스() {  
    int arg2 = 매개값;  
    int var2 = 2;  
    void method() {  
        int arg1 = 매개값;  
        int var1 = 1;  
        int result = arg1+arg2+var1+var2;  
    }  
};
```

## 익명 객체

---

### ❖ 인터페이스: Main.java

```
public interface Calculatable {  
    public int sum();  
}
```

## 익명 객체

### ❖ 익명 객체의 로컬 변수 사용: Anonymous.java

```
public class Anonymous {
    private int field;

    public void method(final int arg1, int arg2) {
        final int var1 = 0;
        int var2 = 0;

        field = 10;
        // arg1 = 20; (x)
        // arg2 = 20; (x)
        // var1 = 30; (x)
        // var2 = 30; (x)

        Calculatable calc = new Calculatable() {
            @Override
            public int sum() {
                int result = field + arg1 + arg2 + var1 + var2;
                return result;
            }
        };
        System.out.println(calc.sum());
    }
}
```

## 익명 객체

---

### ❖ 실행 클래스: AnonymousExample.java

```
public class AnonymousExample {  
    public static void main(String[] args) {  
        Anonymous anony = new Anonymous();  
        anony.method(0, 0);  
    }  
}
```