

A collection of various geometric shapes and patterns. The shapes include circles, squares, triangles, and polygons, some filled with solid colors like pink and others with patterns like dots or stripes. There are also abstract shapes like a cross and a wavy line. The shapes are scattered across the page, with some overlapping. The colors are primarily black, white, and pink. The patterns include dots, stripes, and wavy lines. The shapes are drawn with black outlines. The overall style is minimalist and geometric.

Part1. 익명 객체

1) 익명 자식 객체 생성

- 부모 클래스, 익명 자식 객체 생성
- 실행 결과

2-1) 익명 구현 객체 생성

- 인터페이스, 익명 구현 클래스와 객체 생성
- 실행 결과

2-2) 익명 구현 객체 생성

- UI 클래스, 실행 클래스
- 실행 결과

3) 익명 객체의 로컬 변수 사용

- 인터페이스, 익명 객체의 로컬 변수 사용
- 실행 결과

4) 문제 풀어보기

1) 익명 자식 객체 생성 - 부모 클래스

```
Person.java Anonymous.java AnonymousExample.java
1 package week1_2;
2
3 public class Person {
4     void wake() {
5         System.out.println("7시에 일어납니다.");
6     }
7 }
```

```

1 package week1_2;
2
3 public class Anonymous {
4
5     //필드 초기값으로 대입
6     Person field = new Person() {
7         void work() {
8             System.out.println("출근합니다.");
9         }
10        @Override
11        void wake() {
12            System.out.println("6시에 일어납니다.");
13            work();
14        }
15    };
16
17    void method1() {
18        //로컬 변수값으로 대입
19        Person localVar = new Person() {
20            void walk() {
21                System.out.println("산책합니다.");
22            }
23            @Override
24            void wake() {
25                System.out.println("7시에 일어납니다.");
26                walk();
27            }
28        };

```

```

29
30        //로컬 변수 사용
31        localVar.wake();
32    }
33
34    void method2(Person person) {
35        person.wake();
36    }
37 }

```

1) 익명 자식 객체 생성

Person.java Anonymous.java AnonymousExample.java ✕

```
1 package week1_2;
2
3 public class AnonymousExample {
4     public static void main(String[] args) {
5
6         Anonymous anony = new Anonymous();
7         //익명 객체 필드 사용
8         anony.field.wake();
9         //익명 객체 로컬 변수 사용
10        anony.method1();
11        //익명 객체 매개값 사용
12        anony.method2(
13            new Person() {
14                void study() {
15                    System.out.println("공부합니다.");
16                }
17                @Override
18                void wake() {
19                    System.out.println("8시에 일어납니다.");
20                    study();
21                }
22            }
23        );
24    }
25 }
```

1) 익명 자식 객체 생성

실행 결과

Console ✕ Pro

<terminated> Ano

6시에 일어납니다.

출근합니다.

7시에 일어납니다.

산책합니다.

8시에 일어납니다.

공부합니다.

2-1) 익명 구현 객체 생성 - 인터페이스

RemoteControl.java Anonymous2.java AnonymousExample2.java

```
1 package week1_2;  
2  
3 public interface RemoteControl {  
4     public void turnOn();  
5     public void turnOff();  
6 }
```

```

1 package week1_2;
2
3 public class Anonymous2 {
4
5     //필드 초기값으로 대입
6     RemoteControl field = new RemoteControl() {
7         @Override
8         public void turnOn() {
9             System.out.println("TV를 켭니다.");
10        }
11        @Override
12        public void turnOff() {
13            System.out.println("TV를 끕니다.");
14        }
15    };

```

```

16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36

```

2-1) 익명 구현 객체 생성

- 익명 구현 클래스와 객체 생성

```

void method1() {
    //로컬 변수값으로 대입
    RemoteControl localVar = new RemoteControl() {
        @Override
        public void turnOn() {
            System.out.println("Audio를 켭니다.");
        }
        @Override
        public void turnOff() {
            System.out.println("Audio를 끕니다.");
        }
    };
    //로컬 변수 사용
    localVar.turnOn();
}

void method2(RemoteControl rc) {
    rc.turnOn();
}
}

```

2-1) 익명 구현 객체 생성

- 익명 구현 클래스와 객체 생성

```
RemoteControl.java Anonymous2.java AnonymousExample2.java ✕
1 package week1_2;
2
3 public class AnonymousExample2 {
4     public static void main(String[] args) {
5
6         Anonymous2 anony = new Anonymous2();
7         //익명 객체 필드 사용
8         anony.field.turnOn();
9         //익명 객체 로컬 변수 사용
10        anony.method1();
11        //익명 객체 매개값 사용
12        anony.method2(
13            new RemoteControl() {
14                @Override
15                public void turnOn() {
16                    System.out.println("SmartTV를 켭니다.");
17                }
18                @Override
19                public void turnOff() {
20                    System.out.println("SmartTV를 끕니다.");
21                }
22            }
23        );
24    }
25 }
```

실행 결과

```
Console ✕ Problem
<terminated> Anonymo
TV를 켭니다.
Audio를 켭니다.
SmartTV를 켭니다.
```


2-2) 익명 구현 객체 생성 - UI 클래스

```
Button.java Window.java Main.java
1 package week1_2;
2
3 public class Button {
4     OnClickListener listener;
5
6     void setOnClickListener(OnClickListener listener) {
7         this.listener = listener;
8     }
9
10    void touch() {
11        listener.onClick();
12    }
13
14    interface OnClickListener {
15        void onClick();
16    }
17 }
```

2-2) 익명 구현 객체 생성 - UI 클래스

```
Button.java Window.java Main.java
1 package week1_2;
2
3 public class Window {
4     Button button1 = new Button();
5     Button button2 = new Button();
6
7     //필드 초기값으로 대입
8     Button.OnClickListener listener = new Button.OnClickListener() {
9         @Override
10         public void onClick() {
11             System.out.println("전화를 겁니다.");
12         }
13     };
14
15     Window() {
16         button1.setOnClickListener(listener);
17         button2.setOnClickListener(new Button.OnClickListener() {
18             @Override
19             public void onClick() {
20                 System.out.println("메시지를 보냅니다.");
21             }
22         });
23     }
24 }
```

2-2) 익명 구현 객체 생성 - 실행 클래스

```
Button.java Window.java Main.java ✕  
1 package week1_2;  
2  
3 public class Main {  
4     public static void main(String[] args) {  
5         Window w = new Window();  
6         w.button1.touch();  
7         w.button2.touch();  
8     }  
9 }  
10
```

실행 결과

```
Console ✕ Proc  
<terminated> Main  
전화를 겁니다.  
메시지를 보냅니다.
```

3) 익명 객체의 로컬 변수 사용 - 인터페이스

Calculatable.java ✕ Anonymous3.java AnonymousExample3.java

```
1 package week1_2;  
2  
3 public interface Calculatable {  
4     public int sum();  
5 }  
-
```

```

1 package week1_2;
2
3 public class Anonymous3 {
4     private int field;
5
6     public void method(final int arg1, int arg2) {
7         final int var1 = 0;
8         int var2 = 0;
9
10        field = 10;
11
12        //arg1 = 20; (x)
13        //arg2 = 20; (x)
14
15        //var1 = 30; (x)
16        //var2 = 30; (x)
17
18        Calculatable calc = new Calculatable() {
19            @Override
20            public int sum() {
21                int result = field + arg1 + arg2 + var1 + var2;
22                return result;
23            }
24        };
25
26        System.out.println(calc.sum());
27    }
28 }

```

3) 익명 객체의 로컬 변수 사용

3) 익명 객체의 로컬 변수 사용

```
Calculatable.java Anonymous3.java AnonymousExample3.java ✖
1 package week1_2;
2
3 public class AnonymousExample3 {
4     public static void main(String[] args) {
5
6         Anonymous3 anony = new Anonymous3();
7         anony.method(0, 0);
8     }
9 }
```

실행 결과

```
Console ✖ P
<terminated> Anc
10
```

4) 문제 풀어보기

Q. AnonymousExample 클래스의 실행 결과를 보고 Vehicle 인터페이스의 익명 구현 객체를 이용해서 필드, 로컬 변수의 초기값과 메소드의 매개값을 대입해보세요.

```
Vehicle.java Anonymous.java
1 package week1_2;
2
3 public interface Vehicle {
4
5     public void run();
6 }
7
```

자동차가 달립니다.
승용차가 달립니다.
트럭이 달립니다.

```
Vehicle.java Anonymous.java
1 package week1_2;
2
3 public class Anonymous {
4
5     Vehicle field = .....
6
7     void method1() {
8         Vehicle localVar = .....
9
10        localVar.run();
11    }
12
13    void method2(Vehicle v) {
14        v.run();
15    }
16 }
17
```

4) 문제 풀어보기

Q. AnonymousExample 클래스의 실행 결과를 보고 Vehicle 인터페이스의 익명 구현 객체를 이용해서 필드, 로컬 변수의 초기값과 메소드의 매개값을 대입해보세요.

```
Vehicle.java  Anonymous.java  *AnonymousEx.java ✖
1 package week1_2;
2
3 public class AnonymousEx {
4
5     public static void main(String[] args) {
6         Anonymous an = new Anonymous();
7
8         an.field.run();
9         an.method1();
10        an.method(.....);
11    }
12 }
13
```


출석 과제 (3/18 목 오후 11:55 마감)

Q. 익명 객체에 대한 설명으로 틀린 것은 무엇입니까?

- 1) 익명 객체는 클래스를 상속하거나 인터페이스를 구현해야만 생성될 수 있다.
- 2) 익명 객체는 필드, 매개 변수, 로컬 변수의 초기값으로 주로 사용된다.
- 3) 익명 객체에는 생성자를 선언할 수 있다.
- 4) 부모 클래스나 인터페이스에 선언된 필드와 메소드 이외에 다른 필드와 메소드를 선언할 수 있지만, 익명 객체 내부에서만 사용이 가능하다.