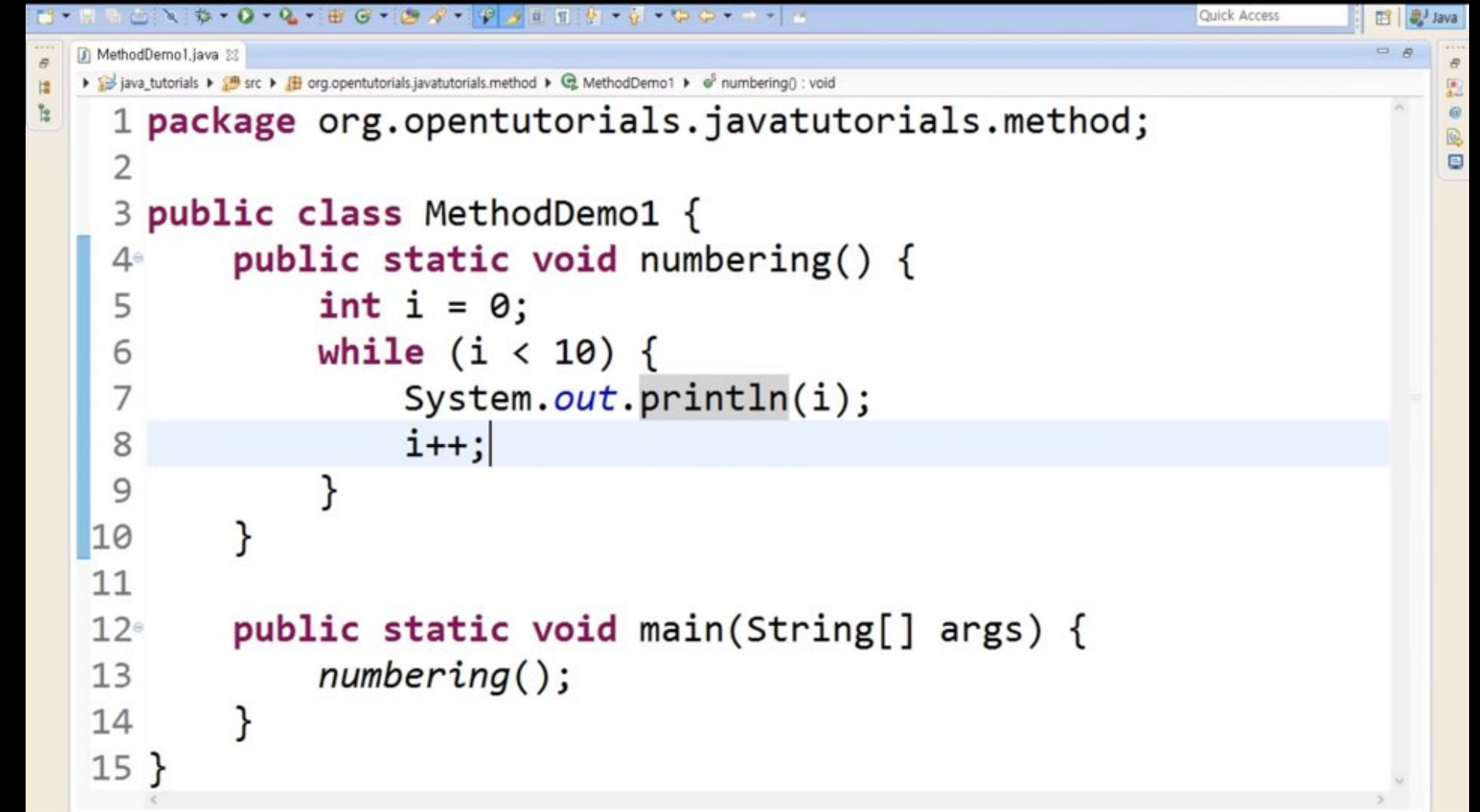


디지털컨버전스 기반 UI/UX Front 전문 개발자양성 과정(비트캠프)

강사 : 이승훈
수강생 : 오진욱

Method

- 수학의 함수와 비슷한 개념
- 예) 자동차 이름,색->(필드) / 전진 후진->(메소드)
- 어떤 입력값(매개변수,인자)를 어떠한 함수를 통하여 출력값(리턴값)을 얻는 것
- 클래스가 가지고 있는 기능을 뜻함 ->클래스 내부에 선언이 됨
- 정의 / 호출 ; 정의는 메서드안 함수를 만드는 것을 말함 / 호출 은그 메서드를 사용하는 것
- Method는 재활용 가능성이 높음 -> 코드량 낮아짐, 유지 보수 유리



```
1 package org.opentutorials.javatutorials.method;
2
3 public class MethodDemo1 {
4     public static void numbering() {
5         int i = 0;
6         while (i < 10) {
7             System.out.println(i);
8             i++;
9         }
10    }
11
12    public static void main(String[] args) {
13        numbering();
14    }
15 }
```

Method

- 입력 방법 :
- `Public static void 이름(매겨 변수){ }` -> 정의
- `이름(인자)` -> 호출

```
1 package org.opentutorials.javatutorials.method;
2
3 public class MethodDemo5 {
4
5     public static void numbering(int init, int limit) {
6         int i = init;
7         while (i < limit) {
8             System.out.println(i);
9             i++;
10        }
11    }
12
13    public static void main(String[] args) {
14        numbering(3, 5);
15    }
16 }
```

The screenshot shows a Java IDE with a file named `MethodDemo5.java`. The code defines a `numbering` method with two parameters, `init` and `limit`, and a `main` method that calls `numbering(3, 5)`. Red annotations highlight the parameter-passing mechanism: `init` and `limit` in the `numbering` method signature are circled in red. In the `main` method, the arguments `3` and `5` in the `numbering(3, 5);` call are also circled in red. Red arrows point from these arguments to their corresponding parameters: one arrow from `3` to `init`, and another from `5` to `limit`. This visualizes how the values are passed from the caller to the callee.

Method

- 출력 (return)방법
- 메소드 외부로 반환하기 위해 사용
- `Public static 데이터 타입 메소드 이름() { } return 이름 ;`
- (리턴 타입은 메소드의 데이터 타입과 일치해야함)
- 메서드는 기계의 부품이라고 생각! 리턴하는 이유는 부품으로서의 가치를 높이기 위해
- 리턴이 선언되면 무조건 종료가 된다.
- 여러개 값의 리턴을 위해서는 -> 배열이용(복수의 리턴 사용은 컴파일 조차 안됨)