

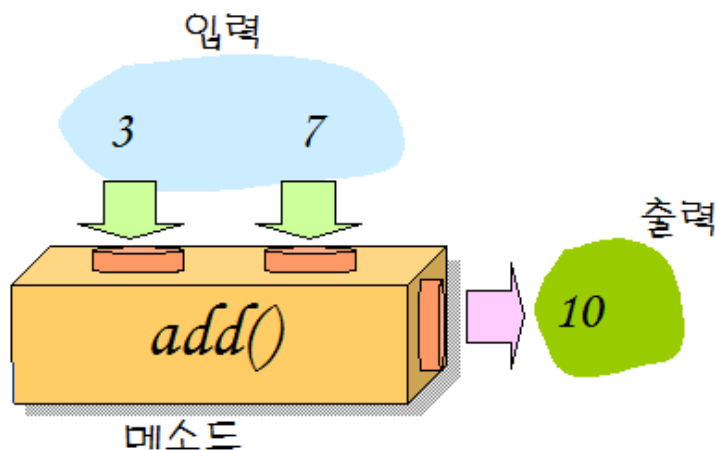
# 클래스와 객체 II

01 클래스와 객체

02 메소드와 필드



### 메소드

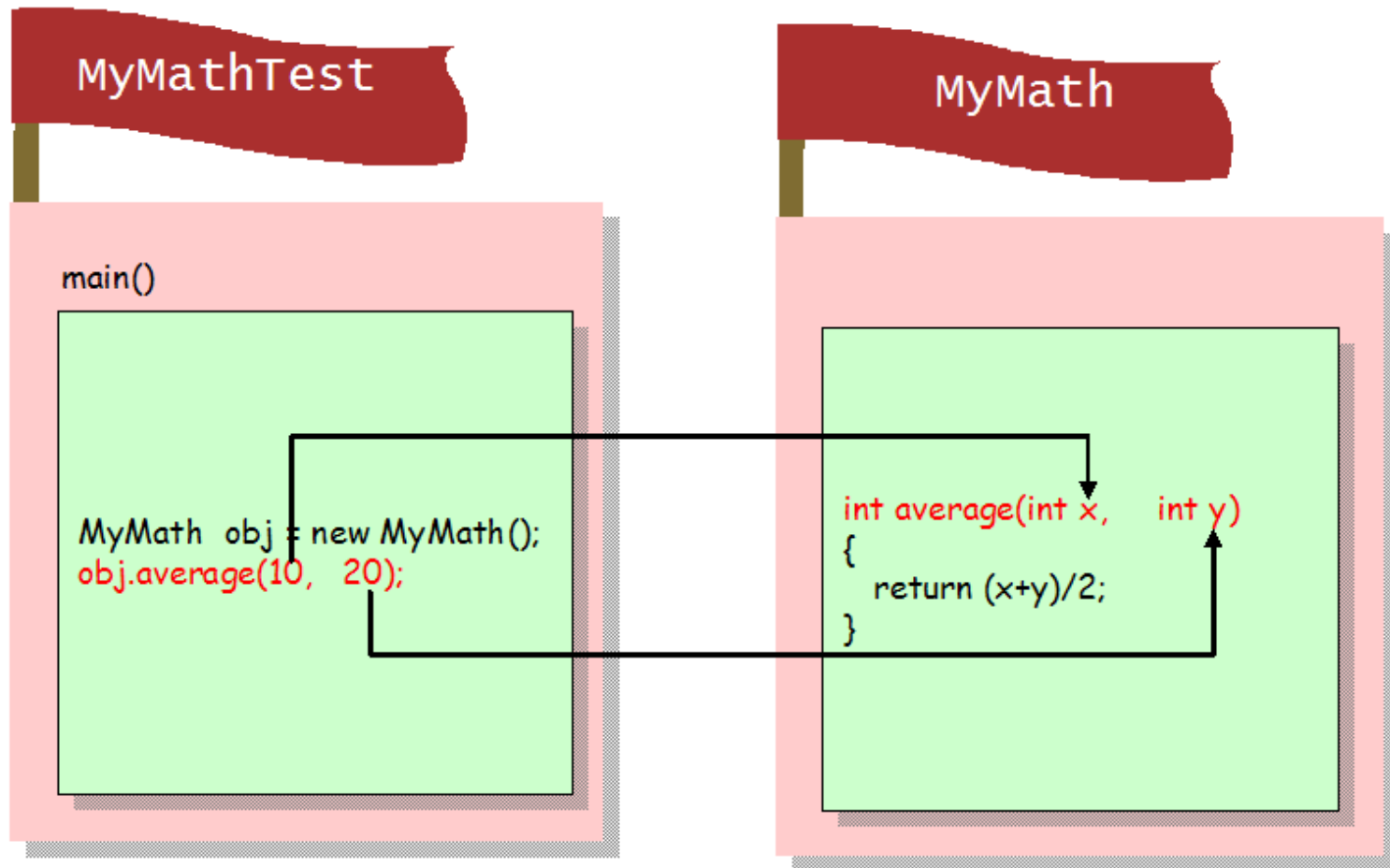


메소드는 입력을 받아서 출력을 내보내는 작은 기계와 같다.

메소드 선언은 다음과 같은 형식을 가진다.

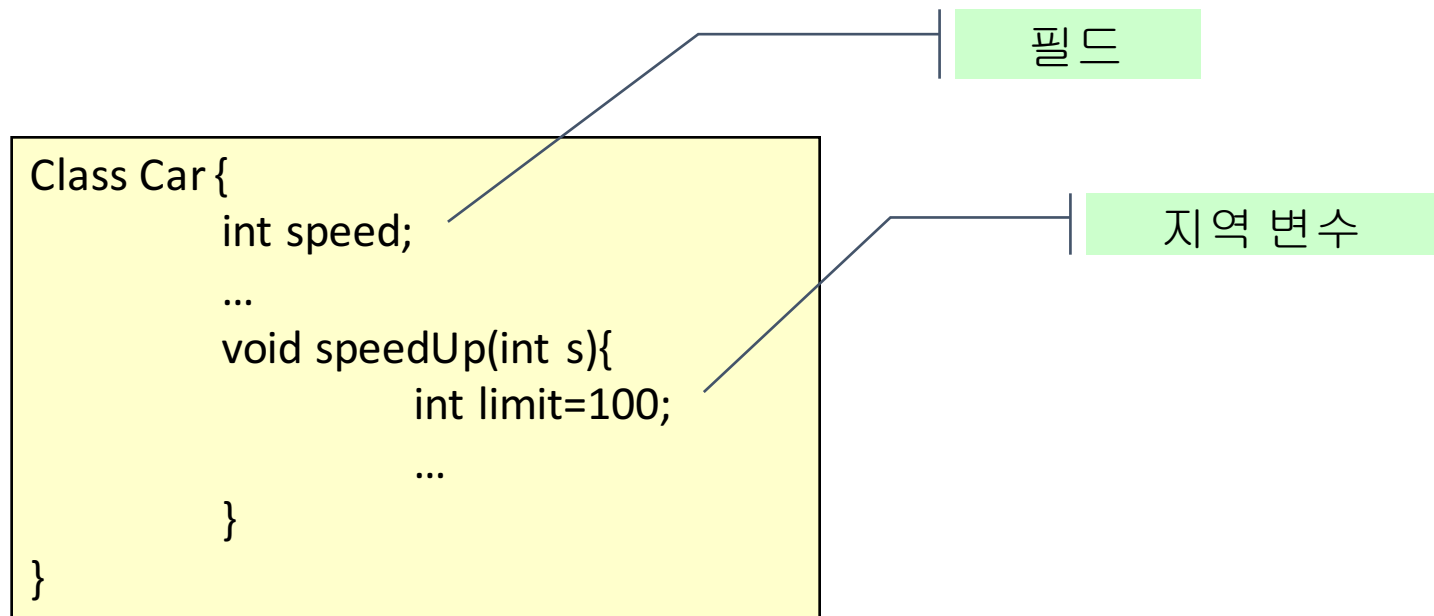
```
[수식자] 반환형 myMethod ( 매개변수 목록 ) {  
    // 문장들  
    ...  
}
```

### 매개 변수

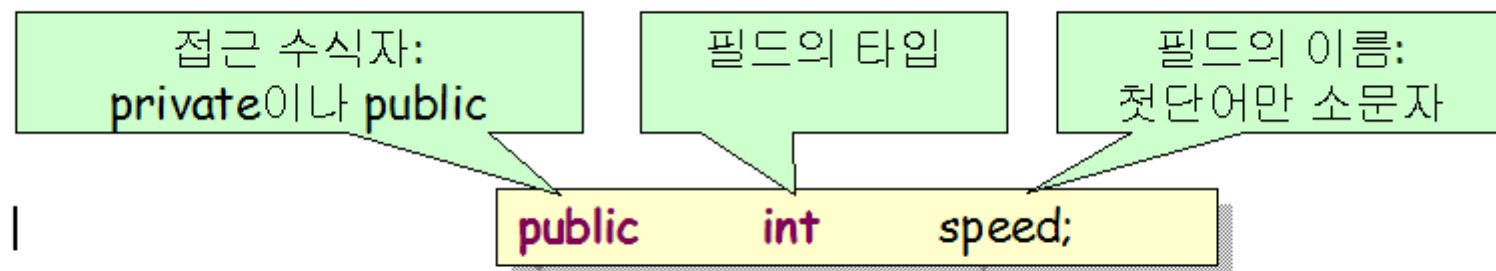


## 자바에서의 변수의 종류

- 필드(field) 또는 인스턴스 변수: 클래스 안에서 선언되는 멤버 변수
- 지역 변수(local variable): 메소드나 블록 안에서 선언되는 변수



# 필드의 선언



필드의 접근 수식자는 어떤 클래스가 필드에 접근할 수 있는지를 표시한다.

- `public` : 이 필드는 모든 클래스로부터 접근가능하다.
- `private` : 클래스 내부에서만 접근이 가능하다.

## 필드의 사용 범위

Date.java

```
public class Date {  
  
    public void printDate() {  
        System.out.println(year + "." + month + "." + day);  
    }  
  
    public int getDay() {  
        return day;  
    }  
  
    // 필드 선언  
    public int year;  
    public String month;  
    public int day;  
}
```

선언 위치와는 상관없이 어디서나 사용이 가능하다.

## 필드의 초기화

- 필드 값은 선언과 동시에 초기화 될 수 있다.

```
public class Classroom {  
    public int capacity = 60;  
    private boolean use = false;  
}
```

- 생성자를 사용하는 방법

```
public class Classroom {  
    public int capacity;  
    private boolean use;  
  
    public Classroom() {  
        capacity = 60;  
        use = false;  
    }  
}
```

## 주의

지역 변수는 사용하기 전에 반드시 초기화를 하여야 한다. 자바에서는 지역 변수를 선언하고 초기화하지 않으면 오류가 발생한다.

```
class BugProgram {  
    ...  
    public int getAverage(int x, int y)  
    {  
        int sum;  
        sum += x;           // 초기화 되지 않은 지역 변수를 사용하면 오류!  
        sum += y;  
        return sum/2;  
    }  
}
```

### 실행결과

```
Exception in thread "main" java.lang.Error: Unresolved compilation problem:  
    The local variable sum may not have been initialized  
    at BugProgram.getAverage(BugProgram.java:6)  
    at Test.main(Test.java:5)
```



## 실습2 클래스와 메소드 정의

- 다음은 주사위 게임 예제입니다.

```
public class DiceGameTest {  
    public static void main(String[] args) {  
        System.out.println("예상 값을 입력하시오:");  
        Scanner s = new Scanner(System.in);  
        int userGuess = s.nextInt();  
  
        int diceFace = (int)(Math.random() * 6) + 1;  
  
        if( diceFace == userGuess )  
            System.out.println("맞았습니다");  
        else  
            System.out.println("틀렸습니다");  
    }  
}
```

## 실습2 클래스와 메소드 정의

- 주사위 게임 예제를 다음과 같은 DiceGame 클래스 구조로 변경하시오.

```
public class DiceGameTest {  
    public static void main(String[] args) {  
        DiceGame game = new DiceGame();  
        game.startPlaying()  
    }  
}  
  
class DiceGame {  
    int diceFace;  
    int userGuess;  
  
    void RollDice() { ...}  
    int getUserInput(String prompt) {... }  
    void checkUserGuess() { ... }  
  
    void startPlaying() {  
        int userGuess = getUserInput("예상값을 입력하시오:");  
        RollDice();  
        checkUserGuess();  
    }  
}
```