

```
16     }
17
18     public int getYear() {
19         return year;
20     }
21
22     public void setYear(int y) {
23         year = y;
24     }
25
26     public String getMonth() {
27         return month;
28     }
29
30     public void setMonth(String m) {
31         month = m;
32     }
33     .... // 나머지 접근자와 설정자
34
35 }
36
37 public class DateTest {
38     public static void main(String[] args) {
39         Date date = new Date();
40         date.setDate(2009, "3월", 2);
41         date.printDate();
42         date.setYear(2010);
43         date.printDate();
44     }
45 }
```

#### 실행결과

2009년 3월 2일  
2010년 3월 2일

## EXERCISE

## INTRODUCTION TO JAVA PROGRAMMING

1. 다음 질문에 대하여 간단히 답하라.
  - ① 클래스와 객체의 차이점은 무엇인가?
  - ② 필드(field)와 지역 변수(local variable)의 차이점은 무엇인가?
  - ③ 메소드의 결과값은 어떤 문장에 의하여 반환되는가?
  - ④ 참조 변수의 선언과 객체의 생성은 어떻게 다른가?
  - ⑤ 지역 변수와 매개 변수(parameter)의 차이점은 무엇인가?
  - ⑥ 기초 변수와 참조 변수의 차이점은 무엇인가?
  
2. 다음 문장이 참인지 거짓인지를 말하라. 거짓이면 그 이유를 설명하라.
  - ① 메소드 이름 다음에 빈괄호가 오면 매개 변수는 없다.
  - ② 기초 타입 변수를 이용하여서 메소드를 호출할 수 있다.
  - ③ 메소드 안에서 정의된 변수는 필드라고 불리며 클래스 안의 모든 메소드가 사용할 수 있다.
  - ④ 기초 타입 지역 변수는 디폴트로 초기화가 된다.
  - ⑤ 참조 타입 인스턴스 변수는 디폴트로 null로 초기화가 된다.
  - ⑥ 참조 타입의 변수도 메소드의 매개 변수로 전달할 수 있다.
  - ⑦ 메소드는 하나 이상의 값을 리턴할 수 있다.
  
3. 필드에 직접 접근하여서 사용하는 것보다 접근자 메소드와 설정자 메소드를 통해서 사용하는 것이 좋은 이유는 무엇인가?
  
4. 다음 프로그램에서 잘못된 부분을 지적하고 올바르게 수정하시오. 그리고 수정된 후의 출력 결과를 쓰시오.

```
class Television {
    String brand;
```

```

    void setBrand(String b){
        brand = b;
    }
    void getBrand(){
        return brand;
    }
    String getManufacturer(Television t){
        return t.getBrand() + "Electronics";
    }
}
public class TelevisionTest {
    public static void main(String[] args){
        t = new television();
        t.setBrand("Samsung");
        String b = getBrand();
        t2 = new Television();
        System.out.println(t2.getManufacturer(t));
    }
}

```

5. 다음은 영화를 나타내는 Movie 클래스이다. 질문에 답하라.

```

public class Movie
{
    String title;
    String director;
    String actors;
}

```

- ① Movie 클래스를 UML로 그려보자. 메소드도 포함시켜라.
- ② 각 필드에 대한 접근자와 설정자를 작성하여 보라.
- ③ 추가할 수 있는 필드와 메소드를 생각하여 보자.
- ④ Movie 클래스를 자바로 작성하여 보라.
- ⑤ Movie 객체를 하나 생성하여 보자.
- ⑥ 생성된 객체의 title 속성을 "Transformer"로 변경하여 보자.
- ⑦ Movie 객체를 3개 생성하여 보자.

### 7. 다음의 은행 업무를 기술한 문장을 읽고 클래스의 후보를 생각하여 보자.

은행은 정기 예금과 보통 예금 업무를 제공한다. 고객들은 자신의 계좌에 돈을 입금할 수 있으며 계좌에서 돈을 인출할 수 있다. 그리고 각 계좌는 기간에 따라 이자를 지급한다. 계좌마다 이자는 달라진다.

- ① 이 문제 영역 기술 문서를 읽고 잠재적인 클래스 후보를 찾아보라. 참고로 업무 기술서의 문장에 등장하는 명사가 클래스의 후보가 된다.
- ② 이 문제를 해결하는 데 필요한 클래스만을 선별하여 보자.
- ③ 각 클래스에 필요한 필드와 메소드를 생각하여 보자.

## PROGRAMMING

### INTRODUCTION TO JAVA PROGRAMMING

1. 사각형을 나타내는 `Rectangle`이라고 이름 붙여진 클래스를 설계하여 보자. `Rectangle`은 사각형의 가로를 나타내는 `width`와 `length`를 필드로 가진다. 또한 사각형의 넓이를 계산하여서 반환하는 `calcArea()`를 메소드로 가진다. `Rectangle` 클래스를 UML로 그리고 접근자, 설정자를 추가하여서 자바로 구현하여 보라.

2. 책을 나타내는 `Book` 클래스를 정의하여 보자. `Book` 클래스는 제목(`title`)과 저자(`author`)를 나타내는 필드를 가진다.

① 각 필드를 직접 접근하여서 제목에 “Great Java”를 대입하고 저자에 “bob”을 대입하여 보자.

② 접근자와 설정자 메소드를 정의하고 이것을 통하여 제목과 저자를 설정하여 보자.

3. 주사위를 나타내는 클래스인 `Dice`를 작성하여 보자. `Dice` 클래스에 필요한 필드와 메소드를 생각하여 보자. 메소드에는 주사위를 굴리는 메소드인 `roll()`을 포함하라. `roll()` 메소드를 작성할 때 난수를 얻는 다음 문장을 참조하라.

```
face = (int) (Math.random() * 6) + 1;
```

`Dice` 클래스를 테스트하기 위한 별도의 클래스를 작성하여 테스트하라.

4. 상자를 나타내는 클래스인 `Box`를 작성하여 보자. `Box` 클래스는 가로, 세로, 높이를 나타내는 필드를 가진다. 또한 박스의 체적을 계산하는 `calcVolume()` 메소드를 가진다. `Box`의 객체를 생성하여서 테스트하여 보자.

5. 직원을 나타내는 `Employee` 클래스를 작성하여 보자. 직원은 이름, 전화 번호, 연봉을 필드로 가지고 있다. 각 필드에 대하여 접근자와 설정자를 작성하라. `EmployeeTest` 클래스를 작성하여서 `Employee` 객체를 생성하고 테스트하라.



6. 핸드폰을 나타내는 `CellPhone` 클래스를 작성하여 보자. 핸드폰의 상태는 모델명, 제조 회사, 색상, 전원, 카메라 장착 여부로 나타낸다. 이것들을 필드로 정의하고 각 필드에 대하여 접근자와 설정자를 작성한다. 전원을 끄고 켜는 메소드도 추가하여 보자. `CellPhoneTest` 클래스를 작성하여서 `CellPhone` 객체를 생성하고 테스트하라.
7. 복소수를 나타내는 `Complex` 클래스를 작성하라. 복소수는 실수부와 허수부로 이루어진다. 필요한 필드와 접근자와 설정자 메소드를 정의하라. 복소수를  $12.0 + 17.9i$ 와 같이 출력하는 `toString()` 메소드를 정의하라. 복소수에 대한 덧셈 연산과 뺄셈 연산을 정의하라.
8. 삼각형을 나타내는 `Triangle` 클래스를 정의하여 보자. `Triangle` 클래스는 밑변과 높이, 면적을 나타내는 필드를 가진다. 각 필드에 대하여 접근자와 설정자를 작성한다. 삼각형의 면적을 구하는 메소드 `getArea()`를 추가하라. `TriangleTest` 클래스를 작성하여서 `Triangle` 객체를 생성하고 테스트하라.
9. 본문에 등장하는 은행 계좌를 나타내는 `BankAccount` 클래스에 다음과 같은 기능을 하는 메소드를 추가하고 테스트하라.

```
// 현재 객체의 잔액에서 amount만큼을 otherAccount 계좌로 송금한다
public int transfer(int amount, BankAccount otherAccount)
{
    ...
}
```

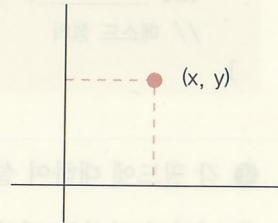
## LAB

## INTRODUCTION TO JAVA PROGRAMMING

1. 2차원 공간에서 하나의 점을 나타내는 Point 클래스를 작성하여 보자.

```
class Point {
    int x;
    int y;
    public int getX()           // 접근자
    {
        _____;
    }
    public void setX(int xCoord) // 설정자
    {
        _____;
    }
}

public class PointTest{
    public static void main(String[] args)
    {
        Point p1 = _____; // 첫번째 객체 생성
    }
}
```



- ① 위의 빈칸을 채우고 실행하여 보자.
- ② 필드 y에 대해서도 접근자와 설정자를 추가하여 보자.
- ③ p1을 통하여 설정자를 호출하여 p1의 x 좌표를 100, y 좌표를 200으로 설정하여 보자.
- ④ 두 번째 객체를 생성하고 (100, 300)으로 설정하라.

2. 상자를 나타내는 Box 클래스를 작성하자.

- ① 먼저 상자의 속성을 나열하여 보자.
- ② 상자의 속성 중에서 가로, 세로, 높이를 나타내는 속성만을 선택하고 width, length, height라는 이름을 부여하여 필드로 만들어라.



- ③ 위의 필드들을 포함하는 Box 클래스를 정의하라.

```
class Box {
    // 필드 정의
    int _____;
    int _____;
    int _____;
    // 메소드 정의
}
```

- ④ 각 필드에 대하여 설정자와 접근자 메소드를 작성하여 Box 클래스에 추가하라.
- ⑤ 상자의 부피를 계산하는 메소드 `getVolume()`을 작성하여 Box 클래스에 추가하라.
- ⑥ 현재 상자의 속성값을 문자열로 만들어서 반환하는 `toString()` 메소드를 작성하여 Box 클래스에 추가하라.
- ⑦ `BoxTest`라는 클래스를 작성하고 `main()` 메소드를 추가한다. `main()` 메소드 안에서 상자 객체를 가리킬 수 있는 참조 변수 `box1`을 정의하라.
- ⑧ 상자 객체를 생성하여서 참조 변수 `box1`이 가리키게 하라.
- ⑨ 설정자 메소드를 호출하여서 상자 `box1`의 가로, 세로, 높이를 100, 100, 100으로 설정하여 보자.
- ⑩ 참조 변수 `box1`을 통하여 `getVolume()` 메소드를 호출하여 보자. 어떤 값이 반환되는가?
- ⑪ `System.out.println(box1);`하면 무엇이 출력되는가?
- ⑫ 두 번째 객체인 `box2`를 생성하고 `box2`의 가로, 세로, 높이를 200, 200, 200으로 설정하여 보자.
- ⑬ 참조 변수 `box2`의 값을 `box1`에 대입한 후에 접근자를 통하여 `box1`의 속성값을 출력하여 보자. 무엇을 알 수 있는가?
3. 본문에 등장하는 은행 계좌를 나타내는 `BankAccount` 클래스를 입력하고 다양한 실험을 하여 보자. `BankAccount`는 잔고를 나타내는 정수형 멤버 변수(`balance`)를 가지고 있고 예금 인출 메소드(`draw`)와 예입 메소드(`deposit`), 현재의 잔고를 반환하는 메소드(`getBalance`)를 가지고 있다.



```

class BankAccount {           // 은행 계좌
    int balance;               // 잔액을 표시하는 변수
    void deposit(int amount) { // 저금
        _____;
    }
    void withdraw(int amount) { // 인출
        _____;
    }
    int getBalance() {
        _____;
    }
}

public class BankAccountTest {
    public static void main(String[] args) {
        _____;
    }
}

```



- ① 설정자 메소드와 접근자 메소드를 정의하라.
- ② main() 메소드 안에서 a1, a2 두 개의 BankAccount 객체를 생성하고 다음과 같은 순서로 메소드를 호출한다.
  - a1의 잔고를 100으로 한다.
  - a2의 잔고를 50으로 한다.
  - a1에서 60을 인출한다.
  - a2에서 30을 인출한다.
  - a1의 현재 잔고를 얻어서 화면에 출력한다.
  - a2의 현재 잔고를 얻어서 화면에 출력한다.
- ③ 현재 잔액을 출력하는 printBalance() 메소드를 구현하고 테스트하라.
- ④ 현재 잔액에 대하여 연 7.5%의 이자를 계산하여 추가하는 addInterest() 메소드를 구현하고 테스트하라.
- ⑤ BankAccount 클래스의 예금 인출 메소드인 withdraw()를 현재 잔고가 음수이면 예금 인출이 일어나지 않도록 변경하라. 이러한 BankAccount 클래스의 변경으로 BankAccountTest 클래스를 변경하여야 하는가?

- ⑥ BankAccount 클래스의 잔고를 나타내는 balance 멤버 변수의 자료형을 int 형에서 double 형으로 변경하라. BankAccount 클래스의 변경으로 BankAccountTest 클래스를 변경하여야 하는가?
- ⑦ BankAccount 클래스 앞에 public을 추가하고 프로그램을 컴파일해보자. 어떤 오류가 발생하는가? public을 붙이는 것과 붙이지 않는 것의 차이는 무엇인가?
- ⑧ BankAccount 클래스 안에도 public static void main(String args[]) { }을 추가하여 보라. 컴파일 오류가 발생하는가? 만약 컴파일 오류가 발생하지 않으면 BankAccount 클래스와 BankAccountTest 클래스의 main() 메소드에 각각 println() 문장을 넣어서 어떤 main() 메소드가 수행되는지를 확인하라.
- ⑨ 7번 문제와 8번 문제의 실험 결과를 토대로 public 클래스와 main() 메소드에 대하여 어떤 결론을 내릴 수 있는가?