

EXERCISE

INTRODUCTION TO JAVA PROGRAMMING

1. 다음 질문에 간단히 답하라.

- ① 서브 클래스에서 재정의된 수퍼 클래스의 메소드를 호출하려면 어떻게 하여야 하는가?
- ② 만약 서브 클래스에 수퍼 클래스의 메소드와 동일한 시그니처를 가진 메소드가 있다고 하면 이것은 재정의인가 아니면 중복 정의인가?
- ③ 메소드가 재정의되는 것을 방지하려면 어떻게 하여야 하는가?
- ④ 메소드 재정의는 어떤 용도로 사용하는가?
- ⑤ 만약 어떤 멤버가 `protected`로 선언되었다면 어떤 코드가 접근할 수 있는가?
- ⑥ 전용 멤버와 패키지 멤버의 차이점은 무엇인가?
- ⑦ 상속 체인의 맨 위에 있는 클래스는 무엇인가?
- ⑧ 우리가 어떤 클래스를 정의하든지 간에 `toString()`과 `equals()` 메소드는 기본으로 가지고 있다. 왜 그럴까?

2. 다음 문장의 참과 거짓을 말하라.

- ① 서브 클래스의 생성자가 수퍼 클래스의 생성자를 명시적으로 호출하지 않으면 자바는 수퍼 클래스의 어떤 생성자도 호출하지 않는다.
- ② 생성자는 상속되지 않는다.
- ③ 수퍼 클래스의 객체는 서브 클래스의 필드에 접근할 수 있다.
- ④ 메소드가 `final`로 선언되면 반드시 재정의하여야 한다.
- ⑤ 수퍼 클래스의 생성자는 서브 클래스의 생성자보다 먼저 호출된다.

3. 다음 소스에서 오류를 최대한 많이 찾으려 하라.

```
public class Bike
{
    private int gear;
```

```

    public int speed;
}
public class MountainBike expands Bike
{
    public int seatHeight;
    public void MountainBike(int g)
    {
        super();
        gear=g;
    }
}

```

4. 다음과 같은 클래스 정의에서 질문에 답하라.

```

class Student {
    private int number;
    public String name;
}

public class GraduateStudent extends Student {
    public String lab;
}

```

- ① 위의 코드에서 수퍼 클래스는 _____이고 서브 클래스는 _____이다.
- ② 위의 클래스 관계를 나타내는 UML 도형을 그려라.
- ③ 각 필드의 접근자와 설정자를 작성하라.
- ④ 생성자를 추가하여 보시오. 서브 클래스의 생성자에서 수퍼 클래스의 생성자를 명시적으로 호출하게 하라.
- ⑤ `GraduateStudent s = new GraduateStudent();`와 같이 객체를 생성하였다고 하자. 다음 중 필드를 잘못 접근한 것은?
 - ① `s.number = 10;`
 - ② `s.lab = "Image Processing Lab";`
 - ③ `s.name = "홍길동";`

5. 다음 프로그램의 출력을 쓰시오.

```

class Animal
{
    String name;
    public Animal()    {
        name = "UNKNOWN";
        System.out.println("동물입니다." + name);
    }
    public Animal(String name) {
        this.name = name;
        System.out.println("동물입니다." + name);
    }
}
class Lion extends Animal
{
    public Lion() {
        System.out.println("사자입니다.");
    }
    public Lion(String name) {
        super(name);
        System.out.println("사자입니다.");
    }
}
public class Test
{
    public static void main(String[] args) {
        Lion lion = new Lion("Brave");
        Lion lion2 = new Lion();
    }
}

```

PROGRAMMING

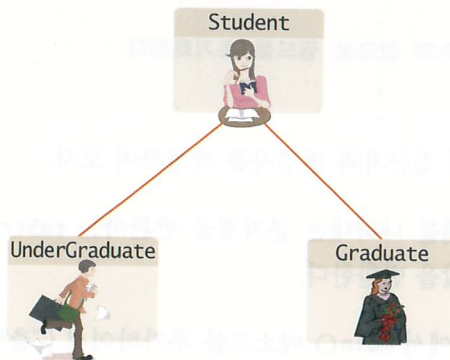
INTRODUCTION TO JAVA PROGRAMMING

1. 회사에서 근무하는 직원들을 나타내는 클래스들을 상속을 이용하여 작성하여 보자.
 - ① Employee 클래스를 설계하라. Employee 클래스는 이름, 사번 등의 정보를 필드로 가져야 한다. 생성자를 정의하고 접근자와 설정자 메소드도 작성하라. 월급을 계산하는 메소드 `computeSalary()`를 구현하라.
 - ② Employee 클래스에서 상속받아서 `SalariedEmployee`라는 클래스를 정의하여 보자. 이 클래스는 월급이라는 필드를 추가로 가진다. 역시 생성자를 정의하고 접근자와 설정자 메소드도 작성하라. 슈퍼 클래스의 `computeSalary()`를 재정의하라.
 - ③ Employee 클래스에서 상속받아서 시간제 직원을 나타내는 `HourlyEmployee` 클래스를 정의하라. 시간당 임금과 일한 시간을 필드로 추가하라. 역시 생성자를 정의하고 접근자와 설정자 메소드도 작성하라. 슈퍼 클래스의 `computeSalary()`를 재정의하라.
2. Person 클래스를 설계하라. Person 클래스는 이름, 주소, 전화 번호를 필드로 가진다. 하나 이상의 생성자를 정의하고 각 필드에 대하여 접근자와 설정자 메소드를 작성하라. 이어서 Person을 상속받아서 Customer를 작성하여 보자. Customer는 고객 번호와 마일리지를 필드로 가지고 있다. 한 개 이상의 생성자를 작성하고 적절한 접근자 메소드와 설정자 메소드를 작성하라.
3. 일반적인 책을 나타내는 Book 클래스를 상속받아서 잡지를 나타내는 Magazine 클래스를 작성하여 보자. Book 클래스는 제목, 페이지수, 저자 등의 정보를 가진다. Magazine 클래스는 추가로 발매일 정보를 가진다. 먼저 UML을 그리고 생성자를 포함하여서 자바로 구현하여 보라.
4. 일반적인 음식을 나타내는 Food 클래스를 상속받아서 멜론을 나타내는 Melon 클래스를 작성하여 보자. Food 클래스는 칼로리, 가격, 중량 등의 정보를 가진다. Melon 클래스는 추가로 경작 농원 정보를 가진다. 먼저 UML을 그리고, 생성자를 포함하여서 자바로 구현하여 보라.

5. Coffee라는 슈퍼 클래스를 작성하고 이 클래스를 상속받아서 Latte, Mocha, Cappuccino, Americano 커피를 정의해보라.

6. 2차원 도형들을 나타내는 클래스들을 작성하여 보자. 슈퍼 클래스인 Shape에는 도형의 위치, 크기 등의 정보가 들어 간다. 각각의 필드에 대하여 접근자와 설정자 메소드를 정의하라. 추가로 도형의 둘레, 면적을 계산하는 메소드도 제공한다. 각 도형은 Shape에서 상속받아서 특정 도형에 맞도록 둘레와 면적을 계산하는 메소드를 수정한다. 테스트 클래스를 작성하여서 여러 도형들을 생성하고 테스트한다.

7. 다음 그림에 해당하는 클래스를 작성하여 보자.



모든 학생은 이름, 학번, 소속 학과, 학년, 이수 학점 수를 가진다. 추가적으로 학부생은 소속 동아리명을 가지고 있고 대학원생은 조교 유형과 장학금 비율을 가진다. 조교 유형에는 교육 조교와 연구 조교가 있으며 장학금 비율은 0과 1사이의 값이다. 각 클래스는 적절한 생성자 메소드, 접근자 메소드, 변경자 메소드를 가진다. 이러한 클래스들의 객체를 만들고 각 객체에 대응된 모든 정보를 출력하는 테스트 클래스를 작성하라..

LAB

INTRODUCTION TO JAVA PROGRAMMING

1. 상속에 대하여 실습하여 보자.

- ① 클래스 Human을 정의하여 보자. 다음과 같은 필드를 정의하라.

```
String name;
int age;
```

- ② 생성자를 작성하여 보자.

```
public Human(String name, int age)
{
    ...// 매개 변수의 값으로 필드를 초기화한다
}
```

- ③ 각 필드에 대하여 접근자와 변경자를 작성하여 보자.

- ④ 객체의 현재 상태를 나타내는 문자열을 반환하는 toString()을 작성하여 보자. 객체의 현재값을 반환한다.

- ⑤ 클래스 Human 안에서 main() 메소드를 추가하여서 (“준향”, 18세), (“몽룡”, 21세), (“사또”, 50)에 해당하는 객체를 생성하여 보라. 각 객체의 toString을 호출해서 출력하여 보라.

- ⑥ 클래스 Human을 상속하여서 Student 클래스를 작성하여 보자. 다음과 같은 필드가 추가된다.

```
String major;
;
```

- ⑦ 다음과 같은 생성자를 정의하여 보자. 부모 클래스의 생성자를 호출하여 보자.

```
public Student(String name, int age, String major)
{
    ...// 매개 변수의 값으로 필드를 초기화한다
}
```

- ⑧ 각 필드에 대하여 접근자와 변경자를 작성하여 보자.

- ⑨ 객체의 현재 상태를 나타내는 문자열을 반환하는 `toString()`을 작성하여 보자. 부모의 `toString()`을 호출하여 문자열을 만들어라.
 - ⑩ 클래스 `Student` 안에서 `main()` 메소드를 추가하여서 (“명진”, 21세, “컴퓨터”), (“미현”, 22세, “경영”), (“용준”, 24세, “경제”)에 해당하는 객체를 생성하여 보라. 각 객체의 `toString`을 호출해서 출력하여 보라.
 - ⑪ `Human`과 `Student` 클래스의 생성자들이 어떤 순서로 호출되는지를 화면에 출력하여 보라.
-
- ⑫ `name`과 `age`의 접근 지정자를 `private`, `protected`, `public`으로 변화시키면서 `Student` 클래스에서 접근하여 보라.
 - ⑬ 현재의 직업을 반환하는 `getProfession()` 메소드를 `Human` 클래스 안에 정의하고 이것을 다시 `Student` 클래스에서 재정의하여 보라.

