연습문제

EXERCISE

이론문제

• 홀수 문제는 정답이 공개됩니다.

1. 다음 클래스에 대해 물음에 답하라.

```
class A {
   private int a;
   public void set(int a) { this.a = a; }
}
class B extends A {
   protected int b, c;
}
class C extends B {
   public int d, e;
}
```

(1) A objA = new objA();에 의해 생성되는 객체 objA의 멤버들을 모두 나열하라.

(2) B objB = new objB();에 의해 생성되는 객체 objB의 멤버들을 모두 나열하라.

(3) C objC = new objC();에 의해 생성되는 객체 objC의 멤버들을 모두 나열하라.

(4) 클래스 D를 다음과 같이 작성하였을 때, 오류가 발생하는 라인을 모두 찾아라.

```
class D extends C {
   public void f() {
      a = 1;     // ①
      set(10);   // ②
      b = 20;    // ③
      d = 30;    // ④
   }
}
```

2. 자바의 모든 클래스가 반드시 상속받게 되어 있는 클래스는?

① Object ② Java ③ Class ④ Super

3. 상속을 이용하여 다음 클래스들을 간결한 구조로 재작성하라.

```
class SharpPencil { // 샤프펜슬
   private int width; // 펜의 굵기
   private int amount; // 남은 량
   public int getAmount() { return amount; }
   public void setAmount(int amount) { this.amount = amount; }
}
class BallPen { // 볼펜
   private int amount; // 남은 량
   private String color; // 볼펜의 색
   public int getAmount() { return amount; }
   public void setAmount(int amount) { this.amount = amount; }
   public String getColor() { return color; }
   public void setColor(String color ) { this.color = color; }
}
class FountainPen  { // 만년필
   private int amount; // 남은 량
   private String color; // 만년필의 색
   public int getAmount() { return amount; }
   public void setAmount(int amount) { this.amount = amount; }
   public String getColor() { return color; }
   public void setColor(String color ) { this.color = color; }
   public void refill(int n) { amount = n; } // 남은 량 보충
}
```

4. 다음 중 설명에 적절한 단어를 기입하라.

자바에서 상속받는 클래스를 _____라고 부르며, _____ 키워드를 이용하여 상속을 선언한다. 상속받은 클래스에서 상속해준 클래스의 멤버를 접근할 때 _____ 키워드를 이용한다. 한편, 객체가 어떤 클래스의 타입인지 알아내기 위해서는 _____ 연산자를 이용하며, 인터페이스는 클래스와 달리 _____ 키워드를 이용하여 선언한다.

5. 상속에 관련된 접근 지정자에 대한 설명이다. 틀린 것은?
   ① 슈퍼 클래스의 private 멤버는 서브 클래스에서 접근할 수 없다.
   ② 슈퍼 클래스의 protected 멤버는 같은 패키지에 있는 서브 클래스에서만 접근할 수 있다.
   ③ 슈퍼 클래스의 public 멤버는 모든 다른 클래스에서 접근할 수 있다.
   ④ 슈퍼 클래스의 디폴트 멤버는 같은 패키지에 있는 모든 다른 클래스에서 접근 가능하다.

6. 다음 빈칸에 적절한 한 줄의 코드를 삽입하라.

```
class TV {
   private int size;
   public TV(int n) {size = n;}
}
class ColorTV _________ TV {
   private int colors;
   public ColorTV(int colors, int size) {
      _____________
      this.colors = colors;
   }
}
```

7. 상속에 있어 생성자에 대해 묻는 문제이다. 실행될 때 출력되는 내용은 무엇인가?

```
class A {
   public A() { System.out.println("A"); }
   public A(int x) { System.out.println("A:" + x); }
}
class B extends A {
   public B() { super(100); }
   public B(int x) { System.out.println("B:" + x); }
}
public class Example {
   public static void main(String[] args) {
      B b = new B(11);
   }
}
```

8. 다음 코드에서 생성자로 인한 오류를 찾아내어 이유를 설명하고 오류를 수정하라.

```
class A {
   private int a;
   protected A(int i) { a = i; }
}
class B extends A {
   private int b;
   public B() { b = 0; }
}
```

9. 다음 추상 클래스의 선언이나 사용이 잘못된 것을 있는 대로 가려내고 오류를 지적하라.

①
```
abstract class A {
    void f();
}
```

②
```
abstract class A {
  void f() { System.out.println("~"); }
}
```

③
```
abstract class B {
    abstract void f();
}
class C extends B {
}
```

④
```
abstract class B {
    abstract int f();
}
class C extends B {
   void f() { System.out.println("~"); }
}
```

10. 추상 클래스를 구현하는 문제이다. 실행 결과와 같이 출력되도록 클래스 B를 완성하라.

```
abstract class OddDetector {
   protected int n;
   public OddDetector (int n) {
      this.n = n;
   }
   public abstract boolean isOdd(); // 홀수이면 true 리턴
}
public class B extends OddDetector {
   public B(int n) {
      super(n);
   }
   public static void main(String [] args) {
      B b = new B(10);
      System.out.println(b.isOdd()); // B가 짝수이므로 false 출력
   }
}
```

```
false
```

11. 다음 상속 관계의 클래스들이 있다.

```
class A {
   int i;
}
class B extends A {
   int j;
}
class C extends B {
   int k;
}
class D extends B {
   int m;
}
```

(1) 다음 중 업캐스팅을 모두 골라라?

① A a = new A(); ② B b = new C(); ③ A a = new D(); ④ D d = new D();

(2) 다음 코드를 실행한 결과는?

```
A x = new D();
System.out.println(x instanceof B);
System.out.println(x instanceof C);
```

(3) 다음 코드를 실행한 결과는?

```
System.out.println(new D() instanceof Object);
System.out.println("Java" instanceof Object);
```

12. 동적 바인딩에 관한 문제이다. 다음 코드가 있을 때 질문에 답하라.

```
class Shape {
   public void draw() { System.out.println("Shape"); }
}
class Circle extends Shape {
   public void paint() {
      ____________________________
   }
   public void draw() { System.out.println("Circle"); }
}
```

(1) Shape s=new Circle(); s.draw()가 호출되면 출력되는 내용은?
(2) s.paint()가 호출되면 "Circle"이 출력되도록 빈 칸에 적절한 코드를 삽입하라.
(3) s.paint()가 호출되면 "Shape"이 출력되도록 빈 칸에 적절한 코드를 삽입하라.

13. 동적 바인딩에 대해 다음에 답하라.

```
abstract class Shape {
   public void paint() { draw(); }
   abstract public void draw();
}
abstract class Circle extends Shape {
   private int radius;
   public Circle(int radius) { this.radius = radius; }
   double getArea() { return 3.14*radius*radius; }
}
```

(1) 다음 중 오류가 발생하는 것을 있는 대로 골라라.

① Shape s;
② Shape s = new Shape();
③ Circle c;
④ Circle c = new Circle(10);

(2) 다음 코드의 실행 결과 "반지름=10"이 출력되도록 Circle 클래스를 수정하라.

```
Circle p = new Circle(10);
p.paint();
```

14. 다형성에 대한 설명 중 틀린 것은?

① 추상 메소드를 두는 이유는 상속 받는 클래스에서 다형성을 실현하도록 하기 위함이다.
② 인터페이스도 구현하는 클래스에서 다형성을 실현하도록 하기 위함이다.
③ 다형성은 서브클래스들이 수퍼 클래스의 동일한 메소드를 서로 다르게 오버라이딩 하여 이루어진다.
④ 자바에서 다형성은 모호한(ambiguous) 문제를 일으키므로 사용하지 않는 것이 바람직하다.

15. 다음 중 인터페이스의 특징이 아닌 것은?

① 인터페이스의 객체는 생성할 수 없다.
② 인터페이스는 클래스와 같이 멤버 변수(필드)의 선언이 가능하다.
③ 인터페이스의 추상 메소드는 자동으로 public이다.
④ 클래스에서 인터페이스를 구현할 때 implements 키워드를 이용하며, 모든 추상 메소드를 작성하여야 한다.

16. 빈칸을 적절히 채우고, 실행 예시와 같이 출력되도록 클래스 TV에 필요한 메소드를 추가 작성하라.

```
_________ Device {
   void on();
   void off();
}
public class TV _____________ Device {
   public static void main(String [] args) {
      TV myTV = new TV();
      myTV.on();
      myTV.watch();
      myTV.off();
   }
}
```

```
켜졌습니다.
방송중입니다.
종료합니다.
```

실습문제

• 홀수 문제는 정답이 공개됩니다.

[1~2] 다음 TV 클래스가 있다.

```
class TV {
   private int size;
   public TV(int size) { this.size = size; }
   protected int getSize() { return size; }
}
```

super() 사용과 서브 클래스 만들기

1. 다음 main() 메소드와 실행 결과를 참고하여 TV를 상속받은 ColorTV 클래스를 작성하라. 난이도 3

```
public static void main(String [] args) {
   ColorTV myTV = new ColorTV(32, 1024);
   myTV.printProperty();
}
```

```
32인치 1024컬러
```

2. 다음 main() 메소드와 실행 결과를 참고하여 문제 1의 ColorTV를 상속받는 IPTV 클래스를 작성하라. 난이도 3

서브 클래스 만들기

```
public static void main(String[] args) {
   IPTV iptv = new IPTV("192.1.1.2", 32, 2048); // "192.1.1.2" 주소에 32인치,
                                                   2048 컬러
   iptv.printProperty();
}
```

```
나의 IPTV는 192.1.1.2 주소의 32인치 2048컬러
```

[3~4] 다음은 단위를 변환하는 추상 클래스 Converter이다.

```
import java.util.Scanner;
abstract class Converter {
   abstract protected double convert(double src); // 추상 메소드
   abstract protected String getSrcString(); // 추상 메소드
   abstract protected String getDestString(); // 추상 메소드
   protected double ratio; // 비율

   public void run() {
      Scanner scanner = new Scanner(System.in);
      System.out.println(getSrcString() + "을 " + getDestString() + "로 바꿉니다.");
      System.out.print(getSrcString() + "을 입력하세요>> ");
      double val = scanner.nextDouble();
      double res = convert(val);
      System.out.println("변환 결과: " + res + getDestString() + "입니다");
      scanner.close();
   }
}
```

추상 클래스를 상속받아 서브 클래스 만들기

3. Converter 클래스를 상속받아 원화를 달러로 변환하는 Won2Dollar 클래스를 작성하라. main() 메소드와 실행 결과는 다음과 같다. 난이도 4

```
public static void main(String args[]) {
   Won2Dollar toDollar = new Won2Dollar(1200); // 1달러는 1200원
   toDollar.run();
}
```

```
원을 달러로 바꿉니다.
원을 입력하세요>> 24000
변환 결과: 20.0달러입니다
```

추상 클래스를 상속받아 서브 클래스 만들기

4. Converter 클래스를 상속받아 Km를 mile(마일)로 변환하는 Km2Mile 클래스를 작성하라. main() 메소드와 실행 결과는 다음과 같다. 난이도 4

```
public static void main(String args[]) {
   Km2Mile toMile = new Km2Mile(1.6); // 1마일은 1.6Km
   toMile.run();
}
```

```
Km을 mile로 바꿉니다.
Km을 입력하세요>> 30
변환 결과: 18.75mile입니다
```

[5~8] 다음은 2차원 상의 한 점을 표현하는 Point 클래스이다.

```
class Point {
   private int x, y;
   public Point(int x, int y) { this.x = x; this.y = y; }
   public int getX() { return x; }
   public int getY() { return y; }
   protected void move(int x, int y) { this.x = x; this.y = y; }
}
```

5. Point를 상속받아 색을 가진 점을 나타내는 ColorPoint 클래스를 작성하라. 다음 main() 메소드를 포함하고 실행 결과와 같이 출력되게 하라. 난이도 5

목적 서브 클래스의 생성자와 메소드 작성 연습

```
public static void main(String[] args) {
   ColorPoint cp = new ColorPoint(5, 5, "YELLOW");
   cp.setXY(10, 20);
   cp.setColor("RED");
   String str = cp.toString();
   System.out.println(str + "입니다.");
}
```

생성자, setXY(), setColor(), toString() 작성 필요

```
RED색의 (10,20)의 점입니다.
```

6. Point를 상속받아 색을 가진 점을 나타내는 ColorPoint 클래스를 작성하라. 다음 main() 메소드를 포함하고 실행 결과와 같이 출력되게 하라. 난이도 5

목적 서브 클래스의 생성자와 메소드 작성 연습

```
public static void main(String[] args) {
   ColorPoint zeroPoint = new ColorPoint(); // (0,0) 위치의 BLACK 색 점
   System.out.println(zeroPoint.toString() + "입니다.");

   ColorPoint cp = new ColorPoint(10, 10);  // (10,10) 위치의 BLACK 색 점
   cp.setXY(5, 5);
   cp.setColor("RED");
   System.out.println(cp.toString() + "입니다.");
}
```

```
BLACK색의 (0,0)의 점입니다.
RED색의 (5,5)의 점입니다.
```

7. Point를 상속받아 3차원의 점을 나타내는 Point3D 클래스를 작성하라. 다음 main() 메소드를 포함하고 실행 결과와 같이 출력되게 하라. 난이도 5

목적 서브 클래스에 필요한 생성자 및 메소드 작성 연습. super 활용

```
public static void main(String[] args) {
   Point3D p = new Point3D(1,2,3); // 1, 2, 3은 각각 x, y, z축의 값.
   System.out.println(p.toString() + "입니다.");

   p.moveUp(); // z 축으로 위쪽 이동
   System.out.println(p.toString() + "입니다.");
```

```
        p.moveDown(); // z 축으로 아래쪽 이동
        p.move(10, 10); // x, y 축으로 이동
        System.out.println(p.toString() + "입니다.");

        p.move(100, 200, 300); // x, y, z 축으로 이동
        System.out.println(p.toString() + "입니다.");
    }
```

```
(1,2,3)의 점입니다.
(1,2,4)의 점입니다.
(10,10,3)의 점입니다.
(100,200,300)의 점입니다.
```

서브 클래스 생성자 및 메소드 작성, super 활용

8. Point를 상속받아 양수의 공간에서만 점을 나타내는 PositivePoint 클래스를 작성하라. 다음 main() 메소드를 포함하고 실행 결과와 같이 출력되게 하라. 난이도 5

```
public static void main(String[] args) {
    PositivePoint p = new PositivePoint();
    p.move(10, 10);
    System.out.println(p.toString() + "입니다.");

    p.move(-5, 5); // 객체 p는 음수 공간으로 이동되지 않음
    System.out.println(p.toString() + "입니다.");

    PositivePoint p2 = new PositivePoint(-10, -10);
    System.out.println(p2.toString() + "입니다.");
}
```

```
(10,10)의 점입니다.
(10,10)의 점입니다.
(0,0)의 점입니다.
```

힌트

Point 클래스의 move()를 PositivePoint 클래스에서 오버라이딩하여 재작성하고 적절히 super.move()를 호출해야 한다. PositivePoint의 2 개의 생성자에서도 적절히 super() 생성자와 super.move()를 호출해야 한다.

9. 다음 Stack 인터페이스를 상속받아 실수를 저장하는 StringStack 클래스를 구현하라.

목적 인터페이스에 대한 이해 및 클래스 구현 활용

```
interface Stack {
   int length(); // 현재 스택에 저장된 개수 리턴
   int capacity(); // 스택의 전체 저장 가능한 개수 리턴
   String pop(); // 스택의 톱(top)에 실수 저장
   boolean push(String val); // 스택의 톱(top)에 저장된 실수 리턴
}
```

그리고 다음 실행 사례와 같이 작동하도록 StackApp 클래스에 main() 메소드를 작성하라. 난이도 6

```
총 스택 저장 공간의 크기 입력 >> 3
문자열 입력 >> hello
문자열 입력 >> sunny
문자열 입력 >> smile
문자열 입력 >> happy
스택이 꽉 차서 푸시 불가!
문자열 입력 >> 그만
스택에 저장된 모든 문자열 팝 : smile sunny hello
```

"그만"을 입력하면 프로그램 종료

10. 다음은 키와 값을 하나의 아이템으로 저장하고 검색 수정이 가능한 추상 클래스가 있다.

목적 추상 클래스의 구현과 활용 연습

```
abstract class PairMap {
   protected String keyArray []; // key 들을 저장하는 배열
   protected String valueArray []; // value 들을 저장하는 배열
   abstract String get(String key); // key 값을 가진 value 리턴. 없으면 null 리턴
   abstract void put(String key, String value); // key와 value를 쌍으로 저장. 기존에
                                                key가 있으면, 값을 value로 수정
   abstract String delete(String key); // key 값을 가진 아이템(value와 함께) 삭제.
                                       삭제된 value 값 리턴
   abstract int length(); // 현재 저장된 아이템의 개수 리턴
}
```

PairMap을 상속받는 Dictionary 클래스를 구현하고, 이를 다음과 같이 활용하는 main() 메소드를 가진 클래스 DictionaryApp도 작성하라. 난이도 7

```
public static void main(String[] args) {
   Dictionary dic = new Dictionary(10);
   dic.put("황기태", "자바");
   dic.put("이재문", "파이선");
   dic.put("이재문", "C++"); // 이재문의 값을 C++로 수정
   System.out.println("이재문의 값은 " + dic.get("이재문"));
   System.out.println("황기태의 값은 " + dic.get("황기태"));
   dic.delete("황기태"); // 황기태 아이템 삭제
   System.out.println("황기태의 값은 " + dic.get("황기태")); // 삭제된 아이템 접근
}
```

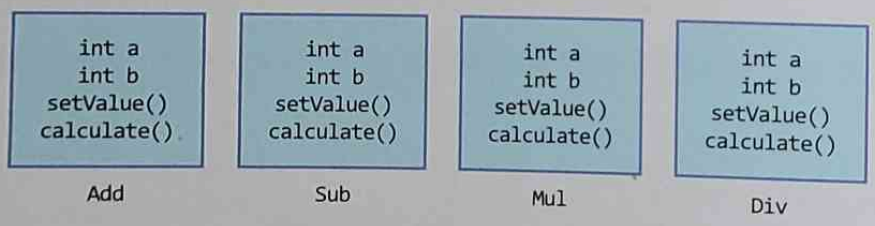
이재문 아이템 출력

황기태 아이템 출력

```
이재문의 값은 C++
황기태의 값은 자바
황기태의 값은 null
```

OO 추상 클래스, 오버라이딩, 동적바인딩

11. 철수 학생은 다음 3개의 필드와 메소드를 가진 4개의 클래스 Add, Sub, Mul, Div를 작성하려고 한다(4장 실습문제 11 참고).
   - int 타입의 a, b 필드: 2개의 피연산자
   - void setValue(int a, int b): 피연산자 값을 객체 내에 저장한다.
   - int calculate(): 클래스의 목적에 맞는 연산을 실행하고 결과를 리턴한다.



곰곰 생각해보니, Add, Sub, Mul, Div 클래스에 공통된 필드와 메소드가 존재하므로 새로운 추상 클래스 Calc를 작성하고 Calc를 상속받아 만들면 되겠다고 생각했다. 그리고 main() 메소드에서 다음 실행 사례와 같이 2개의 정수와 연산자를 입력받은 후, Add, Sub, Mul, Div 중에서 이 연산을 처리할 수 있는 객체를 생성하고 setValue()와 calculate()를 호출하여 그 결과 값을 화면에 출력하면 된다고 생각하였다. 철수처럼 프로그램을 작성하라. 난이도 5

```
두 정수와 연산자를 입력하시오>>5 7 +
12
```

12. 텍스트로 입출력하는 간단한 그래픽 편집기를 만들어보자. 본문 5.6절과 5.7절에서 사례로 든 추상 클래스 Shape과 Line, Rect, Circle 클래스 코드를 잘 완성하고 이를 활용하여 아래 시행 예시처럼 "삽입", "삭제", "모두 보기", "종료"의 4가지 그래픽 편집 기능을 가진 클래스 GraphicEditor을 작성하라. 난이도 7

상속을 이용한 응용 만들기

```
그래픽 에디터 beauty을 실행합니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>1
Line(1), Rect(2), Circle(3)>>2
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>1
Line(1), Rect(2), Circle(3)>>3
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>3
Rect
Circle
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>2
삭제할 도형의 위치>>3
삭제할 수 없습니다.
삽입(1), 삭제(2), 모두 보기(3), 종료(4)>>4
beauty을 종료합니다.
```

힌트

Shape을 추상 클래스로 작성한 사례는 다음과 같다.

```
public abstract class Shape {
   private Shape next;
   public Shape() { next = null; }
   public void setNext(Shape obj) { next = obj; } // 링크 연결
   public Shape getNext() { return next; }
   public abstract void draw(); // 추상 메소드
}
```

13. 다음은 도형의 구성을 묘사하는 인터페이스이다. 난이도 6

인터페이스에 대한 이해와 클래스 구현

```
interface Shape {
   final double PI = 3.14;          // 상수
   void draw();                     // 도형을 그리는 추상 메소드
   double getArea();                // 도형의 면적을 리턴하는 추상 메소드
   default public void redraw() {   // 디폴트 메소드
      System.out.print("--- 다시 그립니다. ");
      draw();
   }
}
```

다음 main() 메소드와 실행 결과를 참고하여, 인터페이스 Shape을 구현한 클래스 Circle를 작성하고 전체 프로그램을 완성하라.

```
public static void main(String [] args) {
  Shape donut = new Circle(10); // 반지름이 10인 원 객체
  donut.redraw();
  System.out.println("면적은 " + donut.getArea());
}
```

```
--- 다시 그립니다. 반지름이 10인 원입니다.
면적은 314.0
```

인터페이스에 대한 이해와 클래스 구현

14. 다음 main() 메소드와 실행 결과를 참고하여, 문제 13의 Shape 인터페이스를 구현한 클래스 Oval, Rect를 추가 작성하고 전체 프로그램을 완성하라. 난이도 7

```
static public void main(String [] args) {
   Shape [] list = new Shape[3];   // Shape을 상속받은 클래스 객체의 레퍼런스 배열
   list[0] = new Circle(10);       // 반지름이 10인 원 객체
   list[1] = new Oval(20, 30);     // 20x30 사각형에 내접하는 타원
   list[2] = new Rect(10, 40);     // 10x40 크기의 사각형

   for(int i=0; i<list.length; i++) list[i].redraw();
   for(int i=0; i<list.length; i++) System.out.println("면적은 " + list[i].getArea());
}
```

```
--- 다시 그립니다. 반지름이 10인 원입니다.
--- 다시 그립니다. 20x30에 내접하는 타원입니다.
--- 다시 그립니다. 10x40크기의 사각형 입니다.
면적은 314.0
면적은 1884.0000000000002
면적은 400.0
```

요약

SUMMARY

- 자바에서 상속은 부모 클래스의 필드와 메소드를 자식 클래스에게 물려주는 것이다. 부모 클래스를 슈퍼 클래스, 자식 클래스를 서브 클래스라고 한다.
- 자바는 클래스의 다중 상속을 지원하지 않는다.
- 자바에서 상속의 선언은 extends 키워드를 사용한다.
- 서브 클래스의 객체에는 슈퍼 클래스의 필드와 메소드가 포함되어 있으나 슈퍼 클래스의 private 멤버는 서브 클래스에서 접근할 수 없다. 그리고 슈퍼 클래스의 protected 멤버는 패키지 소속과 상관없이 서브 클래스에서 접근이 가능하며 동일한 패키지 내의 클래스에서도 접근이 가능하다.
- 서브 클래스의 인스턴스가 생성되면 항상 서브 클래스의 생성자 한 개와 슈퍼 클래스의 생성자 한 개가 실행된다.
- 서브 클래스 객체는 슈퍼 클래스 타입으로 자동 타입 변환이 가능하며 이를 업캐스팅(upcasting)이라고 하며, 다시 원래의 타입으로 강제 타입 변환하는 것을 다운캐스팅(downcasting)이라고 한다.
- instanceof 연산자는 결과 값이 boolean 타입이며 객체가 어떤 클래스 타입인지 판별할 수 있다.
- 슈퍼 클래스에 정의된 메소드를 서브 클래스에서 재정의하는 것을 메소드 오버라이딩(overriding)이라고 한다.
- 서브 클래스에서 슈퍼 클래스의 메소드를 오버라이딩하게 되면 서브 클래스의 인스턴스는 동일한 이름의 메소드를 두 개 가지게 된다. 이때 오버라이딩된 서브 클래스의 메소드가 항상 실행된다.
- 호출된 메소드를 실행 시간에 찾아서 실행하는 것을 동적 바인딩이라고 부르며 오버라이딩된 메소드는 동적 바인딩 방식으로 호출되고 실행된다.
- 추상 메소드(abstract method)는 메소드의 프로토타입만 있고 실행 코드를 작성하지 않은 미완성의 메소드이다. 추상 메소드를 정의하려면 메소드 이름 앞에 abstract라고 선언하여야 한다.
- 추상 클래스(abstract class)는 abstract 키워드로 선언된 클래스이며 한 개 이상의 추상 메소드(abstract)를 포함하는 경우 반드시 추상 클래스로 선언하여야 한다. 그러나 추상 메소드를 하나도 가지고 있지 않은 경우라도 추상 클래스로 선언하는 것이 가능하다.
- 추상 클래스의 객체 혹은 인스턴스는 생성될 수 없다.
- 인터페이스(interface)는 일종의 추상 클래스로서 변수 멤버를 가지지 못한다.
- 인터페이스를 정의하기 위해 interface라는 키워드를 사용한다.
- 클래스가 인터페이스를 구현할 때 implements 키워드를 사용한다. 그리고 인터페이스에 정의된 모든 메소드를 구현하여야 한다.

Bear의 Fish 먹기 게임 만들기

Open Challenge

목적
추상 클래스 이해 및 상속 구현

이 게임에는 Bear와 Fish 객체가 등장하며, 이들은 10행 20열의 격자판에서 각각 정해진 규칙에 의해 움직인다. Bear는 사용자의 키에 의해 왼쪽(a 키), 아래(s 키), 위(d 키), 오른쪽(f 키)으로 한 칸씩 움직이고, Fish는 다섯 번 중 세 번은 제자리에 있고, 나머지 두 번은 4가지 방향 중 랜덤하게 한 칸씩 움직인다. 게임은 Bear가 Fish를 먹으면(Fish의 위치로 이동) 성공으로 끝난다. 다음은 각 객체의 이동을 정의하는 move()와 각 객체의 모양을 정의하는 getShape()을 추상 메소드로 가진 추상 클래스 GameObject이다. GameObject를 상속받아 Bear와 Fish 클래스를 작성하라. 그리고 전체적인 게임을 진행하는 Game 클래스와 main() 함수를 작성하고 프로그램을 완성하라. 난이도 7

```
public abstract class GameObject { // 추상 클래스
   protected int distance; // 한 번 이동 거리
   protected int x, y; // 현재 위치(화면 맵 상의 위치)
   public GameObject(int startX, int startY, int distance) { // 초기 위치와 이동 거리 설정
      this.x = startX;
      this.y = startY;
      this.distance = distance;
   }
   public int getX() { return x; }
   public int getY() { return y; }
   public boolean collide(GameObject p) { // 이 객체가 객체 p와 충돌했으면 true 리턴
      if(this.x == p.getX() && this.y == p.getY())
         return true;
      else
         return false;
   }
   protected abstract void move(); // 이동한 후의 새로운 위치로 x, y 변경
   protected abstract char getShape(); // 객체의 모양을 나타내는 문자 리턴
}
```

키가 입력될 때마다 Bear와 Fish 객체의 move()가 순서대로 호출된다. 게임이 진행되는 과정은 다음 그림과 같으며, 게임의 종료 조건에 일치하면 게임을 종료한다.

```
** Bear의 Fish 먹기 게임을 시작합니다.**
B-------------------
--------------------
--------------------
--------------------
--------------------
-----@--------------
--------------------
--------------------
--------------------
--------------------
왼쪽(a), 아래(s), 위(d), 오른쪽(f) >> s
--------------------
B-------------------
--------------------
--------------------
--------------------
-----@--------------
--------------------
--------------------
--------------------
--------------------
왼쪽(a), 아래(s), 위(d), 오른쪽(f) >> s
--------------------
--------------------
B-------------------
--------------------
--------------------
-----@--------------
--------------------
--------------------
--------------------
--------------------
왼쪽(a), 아래(s), 위(d), 오른쪽(f) >> s
<------------------------------- 중간 과정 생략 ------------------------------->
--------------------
--------------------
--------------------
--------------------
--------------------
----B@--------------
--------------------
--------------------
--------------------
--------------------
왼쪽(a), 아래(s), 위(d), 오른쪽(f) >> f
--------------------
--------------------
--------------------
--------------------
--------------------
-----B--------------
--------------------
--------------------
--------------------
--------------------
Bear Wins!!
```

B : Bear
@ : Fish

<Enter> 키

<Enter> 키

<Enter> 키

<Enter> 키

Bear가 Fish 먹었음