Come to HUFS Meet the World!

객체지향프로그래밍

24.11.06 실습

- 담당 교수 : 전 병 환 교수님
- 조교 : 윤 종 엄
- 조교 메일 : juyoon@hufs.ac.kr

10월 30일 실습1 정답 코드

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

```java
class CreditCardPayment {
    public void pay(double amount) {
        System.out.println("Paid " + amount + " using Credit Card.");
    }
}
class PayPalPayment {
    public void pay(double amount) {
        System.out.println("Paid " + amount + " using PayPal.");
    }
}


class BankTransferPayment {
    public void pay(double amount) {
        System.out.println("Paid " + amount + " using Bank Transfer.");
    }
}

class PaymentProcessor {
    private CreditCardPayment creditCardPayment;
    private PayPalPayment payPalPayment;
    private BankTransferPayment bankTransferPayment;

    public PaymentProcessor() {
        this.creditCardPayment = new CreditCardPayment();
        this.payPalPayment = new PayPalPayment();
        this.bankTransferPayment = new BankTransferPayment();

    }
    public void processPayment(String method, double amount) {

        switch (method) {
            case "CreditCard":
                creditCardPayment.pay(amount);
                break;
            case "PayPal":
                payPalPayment.pay(amount);
                break;
            case "BankTransfer":
                bankTransferPayment.pay(amount);
                break;
            default:
                System.out.println("Unsupported payment method.");
                break;
        }
    }
}

public class Payment {
    public static void main(String[] args) {
        PaymentProcessor paymentProcessor = new PaymentProcessor();
        paymentProcessor.processPayment("CreditCard", 100.0);
        paymentProcessor.processPayment("PayPal", 150.0);
        paymentProcessor.processPayment("BankTransfer", 200.0);
    }
}

```

10월 30일 실습2 정답 코드

한국외국어대학교
HANKUK UNIVERSITY OF FOREIGN STUDIES

```java
class PaymentMethod {
    public void pay(double amount) {
        System.out.println("Default payment of " + amount);
    }
}

class CreditCardPayment extends PaymentMethod {
    @Override
    public void pay(double amount) {
        System.out.println("Paid " + amount + " using Credit Card.");
    }
}

class PayPalPayment extends PaymentMethod {
    @Override
    public void pay(double amount) {
        System.out.println("Paid " + amount + " using PayPal.");
    }
}

class BankTransferPayment extends PaymentMethod {
    @Override
    public void pay(double amount) {
        System.out.println("Paid " + amount + " using Bank Transfer~.");
    }
}

class PaymentProcessor {
    private PaymentMethod paymentMethod;

    public PaymentProcessor(PaymentMethod paymentMethod) {
        this.paymentMethod = paymentMethod;
    }

    public void processPayment(double amount) {
        paymentMethod.pay(amount);
    }
}

public class PaymentDIP {
    public static void main(String[] args) {
        PaymentMethod creditCardPayment = new CreditCardPayment();
        PaymentMethod payPalPayment = new PayPalPayment();
        PaymentMethod bankTransferPayment = new BankTransferPayment();

        PaymentProcessor paymentProcessor1 = new PaymentProcessor(creditCardPayment);
        paymentProcessor1.processPayment(100.0);

        PaymentProcessor paymentProcessor2 = new PaymentProcessor(payPalPayment);
        paymentProcessor2.processPayment(150.0);

        PaymentProcessor paymentProcessor3 = new PaymentProcessor(bankTransferPayment);
        paymentProcessor3.processPayment(200.0);
    }
}
```

자바 인터페이스의 전체적인 특징

- 인터페이스의 객체 생성 불가

오류

```
new PhoneInterface(); // 오류. 인터페이스 PhoneInterface 객체 생성 불가
```

- 인터페이스 타입의 레퍼런스 변수 선언 가능

```
PhoneInterface galaxy; // galaxy는 인터페이스에 대한 레퍼런스 변수
```

- 인터페이스 구현
  - 인터페이스를 상속받는 클래스는 인터페이스의 모든 추상 메소드 반드시 구현
- 다른 인터페이스 상속 가능
- 인터페이스의 다중 상속 가능

예제 5-8 인터페이스 구현

57

PhoneInterface 인터페이스를 구현하고 flash() 메소드를 추가한 SamsungPhone 클래스를 작성하라.

```
** Phone **
띠리리리링
전화가 왔습니다.
전화기에 불이 켜졌습니다.
```

```
interface PhoneInterface { // 인터페이스 선언
   final int TIMEOUT = 10000; // 상수 필드 선언
   void sendCall(); // 추상 메소드
   void receiveCall(); // 추상 메소드
   default void printLogo() { // default 메소드
      System.out.println("** Phone **");
   }
}
class SamsungPhone implements PhoneInterface { // 인터페이스 구현
   // PhoneInterface의 모든 추상 메소드 구현
   @Override
   public void sendCall() {
      System.out.println("띠리리리링");
   }
   @Override
   public void receiveCall() {
      System.out.println("전화가 왔습니다.");
   }

   // 메소드 추가 작성
   public void flash() { System.out.println("전화기에 불이 켜졌습니다."); }
}
public class InterfaceEx {
   public static void main(String[] args) {
      SamsungPhone phone = new SamsungPhone();
      phone.printLogo();
      phone.sendCall();
      phone.receiveCall();
      phone.flash();
   }
}
```

다중 인터페이스 구현

클래스는 하나 이상의 인터페이스를 구현할 수 있음

```
interface AIInterface {
   void recognizeSpeech(); // 음성 인식
   void synthesizeSpeech(); // 음성 합성
}

class AIPhone implements MobilePhoneInterface, AIInterface { // 인터페이스 구현
   // MobilePhoneInterface의 모든 메소드를 구현한다.
   public void sendCall() { ... }
   public void receiveCall() { ... }
   public void sendSMS() { ... }
   public void receiveSMS() { ... }

   // AIInterface의 모든 메소드를 구현한다.
   public void recognizeSpeech() { ... } // 음성 인식
   public void synthesizeSpeech() { ... } // 음성 합성

   // 추가적으로 다른 메소드를 작성할 수 있다.
   public int touch() { ... }
}
```

클래스에서 인터페이스의 메소드를 구현할 때 public을 생략하면 오류 발생

실습 1 : Drawing 프로그램 작성

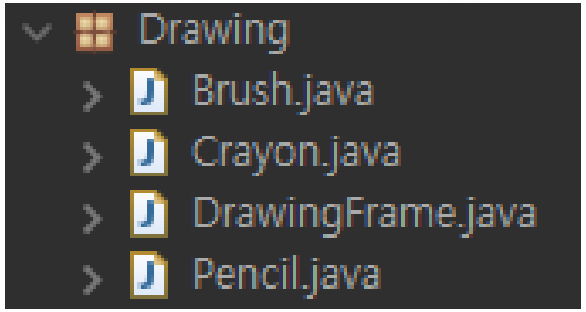
한국외국어대학교 HANKUK UNIVERSITY OF FOREIGN STUDIES

문제 설명:

- 아래의 코드는 현재 **Brush** 기능과 **Pencil** 기능만 처리하고 있다.

1) 크레파스 (**Crayon**) 클래스를 추가하라

2) **DrawingFrame**에서 **Brush** 와 **Pencil** 과 같은 원리로 동작하도록 코드를 추가하라



**** class 분리하여 작성**

아래 코드를 기반으로 문제 해결

한국외국어대학교 HANKUK UNIVERSITY OF FOREIGN STUDIES

```java
import java.util.ArrayList;


public class DrawingFrame {


    public void drawingBrush(Brush b) {

        b.draw();
    }

    public void drawingPencil(Pencil p) {

        p.draw();
    }

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        DrawingFrame f = new DrawingFrame();

        Brush b1 = new Brush();
        Brush b2 = new Brush(3, "red");

        Pencil p1 = new Pencil();
        Pencil p2 = new Pencil(5, "black");

        ArrayList<Brush> brushList = new ArrayList<>();
        brushList.add(b1);
        brushList.add(b2);

        ArrayList<Pencil> pencilList = new ArrayList<>();
        pencilList.add(p1);
        pencilList.add(p2);


        for(int i=0; i<brushList.size(); i++) {
            f.drawingBrush(brushList.get(i));
        }

        for(int i=0; i<pencilList.size(); i++) {
            f.drawingPencil(pencilList.get(i));
        }


    }

}
```

```java
public class Brush {

    private String brushColor;
    private int brushSize;

    public Brush(){
        setColor("white");
        setSize(1);
    }

    public Brush(int size, String color){
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        brushColor = color;
    }

    public void setSize(int thickness) {
        brushSize = thickness;
    }

    public void draw() {
        System.out.println("brush drawing with a " + brushColor+" color and "+brushSize+" size.");
    }
}
```

```java
public class Pencil {

    private String pencilColor;
    private int pencilSize;

    public Pencil(){
        setColor("white");
        setSize(1);
    }

    public Pencil(int size, String color){
        setColor(color);
        setSize(size);
    }

    public void setColor(String color) {
        pencilColor = color;
    }

    public void setSize(int thickness) {
        pencilSize = thickness;
    }

    public void draw() {
        System.out.println("pencil drawing with a " + pencilColor+" color and "+pencilSize+" size.");
    }
}
```

실습 2 : Drawing 프로그램 작성(다형성 적용)

한국외국어대학교 HANKUK UNIVERSITY OF FOREIGN STUDIES

문제 설명:

- 실습 1번에서 완성한 코드의 문제점을 해결해보자

1) 실습 1번 코드의 문제점은 무엇인가?

2) Tool 추상 클래스를 작성 후 **Brush, Pencil, Crayon**에 상속하라

3) 다형성을 사용하는 효율적인 **DrawingFrame(클라이언트 코드)**로 수정하라

아래 코드를 기반으로 동작하도록 작성

한국외국어대학교 HANKUK UNIVERSITY OF FOREIGN STUDIES

```java
import java.util.ArrayList;

public class DrawingFrame {

    //다형성에 해당하는 함수 작성


    public static void main(String[] args) {
        // TODO Auto-generated method stub

        DrawingFrame f = new DrawingFrame();

        Brush b1 = new Brush();
        Brush b2 = new Brush(3, "red");

        Pencil p1 = new Pencil();
        Pencil p2 = new Pencil(5, "black");

        ArrayList<Tool> toolList = new ArrayList<>();
        toolList.add(b1);
        toolList.add(b2);
        toolList.add(p1);
        toolList.add(p2);

        //for문 출력



    }

}
```

DrawingFrame(클라이언트 코드)

```
brush drawing with a white color and 1 size.
brush drawing with a red color and 3 size.
pencil drawing with a white color and 1 size.
pencil drawing with a black color and 5 size.
```

출력 예시

실습 3 : Drawing 프로그램 작성 (인터페이스 활용)

한국외국어대학교 HANKUK UNIVERSITY OF FOREIGN STUDIES

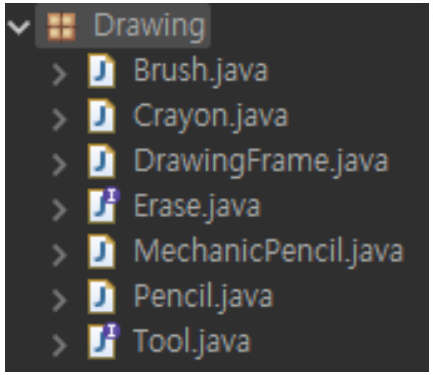
문제 설명:

- 연필과 지우개 기능이 동시에 있는 샤프를 추가하고자 한다.

1) 실습 2번에서 만든 **Tool 추상클래스**를 **인터페이스**로 변경하고**, Erase 인터페이스**를 새롭게 생성하라

2) 연필과 지우개 기능이 동시에 있는 **샤프(MechanicalPencil) 클래스**를 만들어 **Tool**과 **Erase**를 동시에 상속받는 코드를 작성하라

3) DrawingFrame(클라이언트 코드)에서 샤프 기능을 추가하라



완성된 소스 구성