

プログラミング応用演習 b 10月8日 演習課題

プログラム仕様書作成課題

課題クラスを読み、次に示すクラスの仕様書を完成させよ。なお、仕様書は、クラス1つにつき1つ作成する。加えて、図1のようなクラス継承の模式図を作成せよ。

<【クラス名】 のプログラム仕様書>

作成者：【学籍番号】【名前】

(1) クラス

クラス名：【クラス名】

説明：【クラスが何を表現しているか、どのように利用されるか】

スーパークラス：【継承したクラス名、スーパークラスがない場合は「なし」】

(2) フィールド

変数名：【変数名】

- ・ 属性：【アクセスレベル】、【変数の種類（クラス変数、インスタンス変数、ローカル変数）】、【型】
- ・ フィールドの目的：【何の為に用意されたフィールドなのか、どのように利用されているか】

(3) コンストラクタ

コンストラクタ名：【コンストラクタ名】

- ・ 引数：【引数名】、【型】、【変数の目的】
- ・ 処理内容：【処理の内容を記述、処理が複数の手順に分けられる場合、処理する順に箇条書きで列挙する】

(4) メソッド

メソッド名：【メソッド名】

- ・ 属性：【アクセスレベル】、【変数の種類（クラス変数、インスタンス変数、ローカル変数）】、【型】
- ・ 戻り値の型：【型】
- ・ 戻り値の内容：【処理した結果、何を return すればよいか】
- ・ 引数：【引数名】、【型】、【変数の目的】
- ・ 処理内容：【処理の内容を記述、処理が複数の手順に分けられる場合、処理する順に箇条書きで列挙する】
- ・ オーバーライドの有無：【有 or 無、スーパークラスのメソッドをオーバーライドしたか】

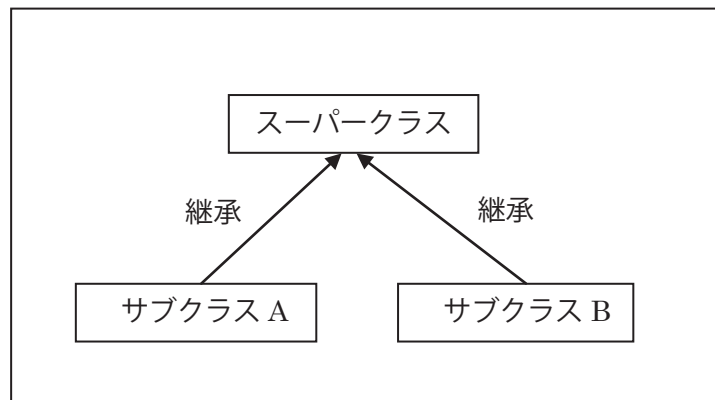


図 1. クラス継承の模式図

課題 1. 10/1 の課題 5 のプログラム仕様書を作成せよ。

```
class MyFamily {
    private String name;
    private int age;

    //コンストラクタの宣言
    MyFamily (String name, int age) {
        [ ? ]
        [ ? ]
    }

    //家族の情報を表示するメソッド
    void display() {
        System.out.println( [ ? ] );
    }
}

public class Prog1001_5{
    public static void main (String[] args){
        //オブジェクトの生成、コンストラクタの実行
        MyFamily father = new MyFamily("hiroshi",50);
        MyFamily mother = new MyFamily("midori",45);
        MyFamily brother = new MyFamily("taro",20);
        //家族の情報の表示
        [ ? ]
        [ ? ]
        [ ? ]
    }
}

/* 実行結果
Name: hiroshi,Age: 50
Name: midori,Age: 45
Name: taro,Age: 20
*/
```

課題2. 以下は、チケットの種別と枚数を入力して購入価格を出力するプログラムである。チケットの種別は a または b で表し、それぞれの価格は 1000 円、2000 円である。なお、各チケットの枚数の制限は 10 枚である。チケットの購入価格は getTicket メソッドで計算し、枚数を入力して価格を出力する。枚数が不足している場合は-1を出力する。プログラム仕様書を作成せよ。

```
import java.io.*;
class Ticket{
    int numberOfA = 10;
    int numberOfB = 10;
    int valueOfA = 1000;
    int valueOfB = 2000;
    public int getTicket(int n){
        return 0;
    }
}
class ATicket extends Ticket    {
    public int getTibket (int n){
        if(numberOfA-n >= 0) {
            numberOfA = numberOfA-n ;
            return valueOfA*n ;
        } else {
            return -1;
        }
    }
}
class BTicket extends Ticket    {
    public int getTicket (int n){
        if(numberOfB-n >= 0) {
            numberOfB = numberOfB-n ;
            return valueOfB*n ;
        } else {
            return -1;
        }
    }
}
```

```
public class Prog1008_2{
    public static void main(String[] args) throws IOException {
        BufferedReader in = new BufferedReader(new InputStreamReader(System.in));
        System.out.print("チケット種別= ");
        char c = in.readLine().charAt(0); //チケット種別を入力
        System.out.print("チケット枚数= ");
        int n = Integer.parseInt(in.readLine()); //チケット枚数を入力
        Ticket t = new Ticket();
        switch( c ){
            case 'a':
                t = new ATicket();
                break;
            case 'b':
                t = new BTicket();
                break;
            default:
                System.out.println("チケット種別が間違っています");
                break;
        }
        int v = t.getTicket(n); //購入価格の計算
        if(v >= 0) {
            System.out.println("購入価格 = "+v);
        } else {
            System.out.println("在庫枚数が不足しています");
        }
    }
}
```

```
/* 実行結果
チケット種別= a
チケット枚数= 10

購入価格 = 10000
```

```
*/
```

課題 3. 以下は、インタフェースとして、ライオン (Lion)、コウモリ (Bat)、蛇 (Snake) を持つキメラ (Chimera) のクラスである。プログラム仕様書を作成せよ。

```
interface Lion{
    String body ="らいおん";
    void breath();
}
interface Bat{
    String wing="こうもりの翼";
    void fly();
}
interface Snake(){
    String tail();
}

class Chimera implements Lion, Bat, Snake {
    public void breath(){
        System.out.println(body + "は、炎をはいた");
    }
    public void fly(){
        System.out.println(wing + "で空をとんだ");
    }
    public void poison(){
        System.out.println(tail + "かみついた");
        System.out.println("どくにおかされた");
    }
}

public class Prog1008_3{
    public static void main(String[] args){
        Chimera chime1 = new Chimera();
        System.out.println("キメラが現れた");
        chime1.breath();
        chime1.fly();
        chime1.poison();
    }
}
```

課題 4. 課題 3 の Chimera クラスと以下のコメント文を参考に、プログラムを完成させ、仕様書を作成せよ。

```
//インタフェース Human
//変数は、body は姿, skill は技で、どちらも String 型
//「能力発揮」の抽象メソッド名 = ability : 引数無し、戻り値無し
【*****】 {
    【*****】 ="人間"; //姿の名前
    【*****】 ="瞬間移動"; //技の名前
    【*****】 ; //「能力発揮」の抽象メソッド
}
//インタフェース Dragon
//変数は、power で文字列
//「変身」の抽象メソッド名 = shapeChange : 引数無し、戻り値無し
//「攻撃」の抽象メソッド名 = attack : 引数無し、戻り値無し
【*****】 {
    【*****】 ="竜の力";
    【*****】 // 「変身」の抽象メソッド
    【*****】 // 「攻撃」の抽象メソッド
}
// Human, Dragon の実装クラス DragonMan
class DragonMan 【*****】 {
    【*****】 { //「能力発揮」の実メソッド
        System.out.println(body + "の姿で" + skill + "した");
    }
    【*****】 { //「変身」の実メソッド
        System.out.println("竜に変身した");
    }
    【*****】 { //「攻撃」の実メソッド
        System.out.println(power + "で攻撃した");
    }
}

public class Prog1008_4{
    public static void main(String[] args){
        DragonMan dm = new DragonMan();
        System.out.println("竜人が現れた");
        dm.ability();
        dm.shapeChange();
        dm.attack();
    }
}

/* 実行結果
$java Prog1008_4
竜人が現れた
人間の姿で瞬間移動した
竜に変身した
竜の力で攻撃した
*/
```

課題 5. 以下は、2014 年 10 月のカレンダーを作成するプログラムである。java.util.Calendar クラスについて調べ、コメントの(****1****)～(****5****)を埋めよ。コメントを付加した上で動作を確認し、そのファイルを提出せよ。

```
import java.lang.String;
import java.lang.System;
import java.util.Calendar;
import static java.util.Calendar.*;
public class MondayCalendar {
    public static void main(String[] A00) {
        Calendar cal = Calendar.getInstance();
        // (****1****)
        int year = cal.get(YEAR);
        // (****2****)
        int month = cal.get(MONTH);
        // (****3****)
        cal.set(year, month, 1);
        // (****4****)
        int day = cal.getActualMaximum(DAY_OF_MONTH);
        // (****5****)
        int dweek = cal.get(DAY_OF_WEEK);

        System.out.printf("%d 年%d 月のカレンダー-%n", year, month + 1);
        System.out.println("月 火 水 木 金 土 日");

        // 空白の個数を求める
        int i;
        if (dweek < MONDAY) {
            i = (dweek + 7) - MONDAY;
        } else {
            i = dweek - MONDAY;
        }

        int k = 0;
        for (int j = 0; j < i; j++) {
            System.out.print(" ");
            k++;
        }
        for (int j = 1; j <= day; j++) {
            if (i < 6) {
                System.out.printf("%2d ", j);
            } else {
                System.out.printf("%2d¥n", j);
            }
            i = ++i % 7;
        }
    }
}

/*
実行結果
C:¥JAVA>java MondayCalendar
2014 年 10 月のカレンダー
月 火 水 木 金 土 日
    1  2  3  4  5
  6  7  8  9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28 29 30 31
*/
```