

確認○×問題

次の各文は正しいか誤っているか答えなさい。

- (1) 新しいクラスを宣言するとき既存のクラスを利用することはできない
 - × ※既存のクラスを再利用して効率的に新しいクラスを構成することができる
- (2) 新しいクラスが既存のクラスのメンバーを受け継ぐことを継承という
 - ※新しいクラスは継承されたメンバーと追加分のメンバーにより宣言される
- (3) クラスの拡張における既存のクラスをサブクラスという
 - × ※既存のクラスをスーパークラスといい、新しいクラスをサブクラスという
- (4) サブクラスからスーパークラスの `private` メンバーにアクセスすることはできない
 - ※`public` や `protected` メンバーであればアクセスできる
- (5) クラスの拡張はキーワード `extends` により指定する
 - ※『`class` 新しいクラス `extends` 既存のクラス』の順で宣言する
- (6) `super();` はコンストラクタのどこに書いても良い
 - × ※`super();` や `this();` はコンストラクタの先頭行に書く必要がある
- (7) 引数なしの `super();` によりスーパークラスの引数なしのコンストラクタが実行される
 - ※`super();` の引数によりオーバーロードされたコンストラクタを指定できる
- (8) スーパークラスの `protected` メンバーはサブクラスからアクセスが可能である
 - ※スーパークラスの `private` メンバーにはサブクラスからでさえアクセスができない

```
1 //////////////////////////////////////////////////
2 // 課題1 次は文房具全般を表す文房具クラスです。この文房具クラス...
3 //////////////////////////////////////////////////
4
5 // 文房具クラス (スーパークラス)
6 class Stationery
7 {
8     public String whattodo; // 機能
9
10    public void showStationery(){
11        System.out.println("機能:"+whattodo);
12    }
13 }
14
15 // ボールペンクラス (サブクラス)
16 class BallpointPen extends Stationery
17 {
18     public String color; // 色
19     public double size; // 太さ(mm)
20
21     public void showBallpointPen(){
22         System.out.println("【ボールペン】");
23         showStationery();
24         System.out.println("色:"+color);
25         System.out.println("太さ(mm):"+size);
26     }
27 }
28
29 // ノートクラス (サブクラス)
30 class Notebook extends Stationery
31 {
32     public char type; // A, B
33     public int size; // 1, 2, 3, 4, 5
34
35     public void showNotebook(){
36         System.out.println("【ノート】");
37         showStationery();
38         System.out.println("サイズ:"+type+size);
39     }
40 }
41
42 // 物差しクラス (サブクラス)
43 class Ruler extends Stationery
44 {
45     public int size; // 長さ(cm)
46
47     public void showRuler(){
48         System.out.println("【物差し】");
49         showStationery();
50         System.out.println("長さ(cm):"+size);
51     }
52 }
53
54 // 修正液クラス (サブクラス)
55 class Whiteout extends Stationery
56 {
57     public int amount; // 容量(ml)
58
59     public void showWhiteout(){
60         System.out.println("【修正液】");
61         showStationery();
62         System.out.println("容量(ml):"+amount);
63     }
64 }
65
66 class Assignment6_1
67 {
68     public static void main(String[] args){
69         // 作成したサブクラスの動作を確認 (ボールペンクラスの場合)
70         BallpointPen mypen=new BallpointPen();
71         mypen.whattodo="字を書く";
72         mypen.color="青";
73         mypen.size=0.7;
74         mypen.showBallpointPen();
75     }
76 }
77
```

```
1 //////////////////////////////////////////////////
2 // 課題2 次は建物全般を表す建物クラスです。この建物クラス...
3 //////////////////////////////////////////////////
4
5 // 建物クラス (スーパークラス)
6 class Building
7 {
8     public int floors; // 何階建
9     public int height; // 高さ
10
11     public void showBuilding() {
12         System.out.println(floors+"階建");
13         System.out.println("高さ："+height+"m");
14     }
15 }
16
17 // マンションクラス (サブクラス)
18 class Mansion extends Building
19 {
20     public int families; // 世帯数
21
22     public void showMansion() {
23         showBuilding();
24         System.out.println("世帯数："+families);
25     }
26 }
27
28 // デパートクラス (サブクラス)
29 class Department extends Building
30 {
31     public int shops; // 店舗数
32     public int restaurants; // 食堂数
33
34     public void showDepartment() {
35         showBuilding();
36         System.out.println("店舗数："+shops);
37         System.out.println("食堂数："+restaurants);
38     }
39 }
40
41 // 大学クラス (サブクラス)
42 class University extends Building
43 {
44     public int laboratories; // 研究室数
45     public int lecture_rooms; // 講義室数
46
47     public void showUniversity() {
48         showBuilding();
49         System.out.println("研究室数："+laboratories);
50         System.out.println("講義室数："+lecture_rooms);
51     }
52 }
53
54 class Assignment6_2
55 {
56     public static void main(String[] args) {
57         // 作成したサブクラスの動作を確認 (マンションクラスの場合)
58         Mansion mymansion=new Mansion();
59         mymansion.floors=10;
60         mymansion.height=20;
61         mymansion.families=50;
62         mymansion.showMansion();
63     }
64 }
65
```

```

1 //////////////////////////////////////////////////
2 // 課題3 次は物体全般を表す物体クラスです。この物体クラス...
3 //////////////////////////////////////////////////
4
5 // 物体クラス (スーパークラス)
6 class Objects
7 {
8     public String name; // オブジェクトの名前
9     public double[] size; // サイズ縦、横、奥行 (mm)
10    public double weight; // 重さ (g)
11
12    public Objects()
13    {
14        // フィールドの初期設定
15        name="No name";
16        size=new double[3];
17        for(int i=0;i<size.length;i++)
18            size[i]=0.0;
19        weight=0.0;
20    }
21    public Objects(String n, double[] s, double w)
22    {
23        // 与えられたデータをフィールドに設定
24        name=n;
25        size=s;
26        weight=w;
27    }
28
29    public void show()
30    {
31        // 各フィールドを表示
32        System.out.println("Name:"+name);
33        System.out.println("size:H"+size[0]+", W"+size[1]+", D"+size[2]+"(mm)");
34        System.out.println("Weight:"+weight+"(g)");
35    }
36 }
37
38 // 携帯クラス (サブクラス)
39 class Mobile extends Objects
40 {
41     public int per_minute_charge; // 通話料金 (円/分)
42
43     public Mobile()
44     {
45         per_minute_charge=0;
46     }
47     public Mobile(String n, double[] s, double w, int p)
48     {
49         super(n,s,w);
50         per_minute_charge=p;
51     }
52     public void showall()
53     {
54         show();
55         System.out.println("Per-minute charge:"+per_minute_charge+"(yen)");
56     }
57 }
58
59 class Assignment6_3
60 {
61     public static void main(String[] args)
62     {
63         double[] size={100,50,12};
64         Mobile mymobile=new Mobile("J102",size,50,40);
65
66         mymobile.showall();
67     }
68 }
69

```

```

1 //////////////////////////////////////////////////
2 // 課題4 MP3プレーヤや携帯など電子ガジェットを管理する...
3 //////////////////////////////////////////////////
4
5 // スーパークラスGadget
6 class Gadget
7 {
8     private String name; // 製品名
9     private int price;    // 価格
10
11     public Gadget(String n, int p)
12     {
13         name=n;
14         price=p;
15     }
16     public void show()
17     {
18         System.out.println("製品名："+name+"、価格："+price+"円");
19     }
20 }
21
22 // サブクラスMp3Player
23 class Mp3Player extends Gadget
24 {
25     private int memory; // 容量MB
26
27     public Mp3Player(String n, int p, int m)
28     {
29         super(n, p); // スーパークラスのコンストラクタを呼ぶ
30         memory = m;
31     }
32     public void print()
33     {
34         System.out.println("[MP3Player]");
35         show(); // 継承されたメソッドを使用
36         System.out.println("メモリ："+memory+"MB");
37     }
38 }
39
40 // サブクラスCellPhone
41 class CellPhone extends Gadget
42 {
43     private String number; // 番号
44
45     public CellPhone(String n, int p, String num)
46     {
47         super(n, p); // スーパークラスのコンストラクタを呼ぶ
48         number = num;
49     }
50     public void print()
51     {
52         System.out.println("[CellPhone]");
53         show(); // 継承されたメソッドを使用
54         System.out.println("番号："+number);
55     }
56 }
57
58 class Assignment6_4
59 {
60     public static void main(String[] args)
61     {
62         Mp3Player myplayer = new Mp3Player("Creative Zen nano",9980,1024);
63         CellPhone myphone = new CellPhone("Nokia 6280",37000,"09012345678");
64
65         myplayer.print();
66         myphone.print();
67     }
68 }
69
70

```

```

1 ///////////////////////////////////////////////////
2 // 課題5 次は仕事するヒトを表わすクラスWorkerである。...
3 ///////////////////////////////////////////////////
4
5 // Workerクラス (スーパークラス)
6 class Worker
7 {
8     public String name; // 名前
9     public int age;      // 年齢
10
11     public void showWorker(){
12         System.out.println("名前:"+name);
13         System.out.println("年齢:"+age);
14     }
15 }
16
17 // 医者クラス (サブクラス)
18 class Doctor extends Worker
19 {
20     public int patients; // 患者数
21
22     public void showDoctor(){
23         System.out.println("■医者■");
24         showWorker();
25         System.out.println("患者数:"+patients);
26     }
27 }
28
29 // 運転手クラス (サブクラス)
30 class Driver extends Worker
31 {
32     public int career; // 運転歴
33
34     public void showDriver(){
35         System.out.println("■運転手■");
36         showWorker();
37         System.out.println("運転歴:"+career);
38     }
39 }
40
41 // 先生クラス (サブクラス)
42 class Professor extends Worker
43 {
44     public String major; // 専門
45
46     public void showProfessor(){
47         System.out.println("■先生■");
48         showWorker();
49         System.out.println("専門:"+major);
50     }
51 }
52
53 // 医者の場合のコード例
54 class Assignment6_5
55 {
56     public static void main(String[] args){
57         Doctor doc=new Doctor();
58         doc.name="福工大 太郎";
59         doc.age=25;
60         doc.patients=55;
61         doc.showDoctor();
62     }
63 }
64

```

```
1 ////////////////////////////////////////////
2 // 課題6 動物園や植物園など公園を管理するクラスを作り...
3 ////////////////////////////////////////////
4
5 // 公園クラス (スーパークラス)
6 class Park
7 {
8     public String name;    // 名称
9     public int price;      // 入場料
10
11     public Park(String n, int p){
12         name = n;
13         price = p;
14     }
15     public void show(){
16         System.out.println(name+" (入場料"+price+"円) ");
17     }
18 }
19
20 // 動物園クラス (サブクラス)
21 class Zoo extends Park
22 {
23     public int animals;    // 動物数
24
25     public Zoo(String n, int p, int a){
26         super(n, p);
27         animals = a;
28     }
29     public void print(){
30         show();
31         System.out.println(animals+"種類の動物が待っています。");
32     }
33 }
34
35 // 植物園クラス (サブクラス)
36 class Garden extends Park
37 {
38     public int plants;    // 植物数
39
40     public Garden(String n, int p, int pt){
41         super(n, p);
42         plants = pt;
43     }
44     public void print(){
45         show();
46         System.out.println(plants+"種の植物を観察できます!");
47     }
48 }
49
50 class Assignment6_6
51 {
52     public static void main(String[] args){
53         Zoo myzoo=new Zoo("福岡動物園", 400, 130);
54         Garden mygarden=new Garden("鳥取花回廊", 1000, 400);
55         myzoo.print();
56         mygarden.print();
57     }
58 }
59
```