
18 章

1.

問1 A C D

スーパークラスにサブクラスのオブジェクトを代入する操作だけが正しい。

問2 D E

instanceof でチェックするクラス型が、ObjectOutputStream か、そのスーパークラスであるときのみに true になる。

2. B. 20

継承関係にあるクラス同士で、同じ名前のフィールド変数を定義することは問題ない。その場合、常に、自クラスで定義した変数が優先される。

Exec クラスでは、Rect 型オブジェクトをそのスーパークラスである Shape 型の変数 shape に代入している。disp メソッドは Rect クラスでオーバーライドされているので、shape.disp() で実行されるのは Rect クラスの disp メソッドである。disp メソッドで出力する val は、自クラス、つまり、Rect クラスの val なので、20 と表示される。

さらに詳しい解説は、20 章の「フィールドの隠ぺい」(P.491) を参照すること。

3.

```
package pass18_03_1;
import lib.Input;
```

```
public class ArithmeticMean extends Mean{
    private double sum;      // 合計値
    private int times;      // データ数

    // 入力したデータの平均値を計算して返す
    @Override
    public double process(){
        sum = 0;
        double dt;
        while((dt=Input.getDouble())!=0){ // 値を入力する
            sum += dt;                    // sum に類型する
            times++;                      // データ数をカウントする
        }
        return sum/times;                // 平均を計算して返す
    }

    // 平均値を受け取って表示する
    @Override
    public void display(double mean){
        System.out.println("平均="+mean);
    }
}
```

以下は、平均値を計算して表示する汎用クラス Compute と、実際に、Compute クラスのオブジェクトを作って、run メソッドを実行する Exec クラス。問題文にあるクラスと同じものを参考までに表示しておく。Compute クラスの run メソッドの引数は、Mean 型であるが、Exec クラスでは、サブクラスの ArithmeticMean 型のオブジェクトを指定している。

```
package pass18_03_1;
public class Compute {
    public void run(Mean mean){
        double answer = mean.process();    // 平均を計算して答えを返す
        mean.display(answer);              // 答えを表示する
    }
}

public class Exec {
    public static void main(String[] args){
        Compute com = new Compute();
        com.run( new ArithmeticMean() );
    }
}
```