
19 章

◆ 19-1

1.

```
package ex19_01_1;

public abstract class Hand {
    private int[] number;
    public abstract void play();
    public int[] getNumber(){
        return number;
    }
    public Hand(int[] number){
        this.number = number;
    }
}
```

2. D. 抽象クラス型の変数にオブジェクトを代入してもよい

A.について、必ずしもすべての抽象メソッドをオーバーライドする必要はない。自身を抽象クラスにすれば、抽象メソッドのまま残しておいてもよい。B.について、コンストラクタには `abstract` を付けることはできない。C.について、抽象クラスはオブジェクトを作れない。E.について、フィールド変数には `abstract` を付けることはできない。F.について、`abstract` は戻り値型よりも前に書かねばならない。

◆ 19-2

1.

- ① `implements`
- ② **実装する**
- ③ `interface`
- ④ ⑤ `public abstract` （順序は逆でもよい）
- ⑥ `abstract`
- ⑦ ⑧ `private protected` （順序は逆でもよい）

2. 2行目 5行目

インタフェースメソッドは `public` かつ `abstract` である。`protected` や `final` は指定できない。

◆ 19-3

1-1.

```
package ex19_03_1;
public class City implements Computable{
    private String name;
    private double population;
    public City(String name, double population) {
        this.name = name;
        this.population = population;
    }
    public String getName() {
        return name;
    }
    public double getPopulation() {
        return population;
    }
    @Override
    public String itemName() {
        return "人口";
    }
    @Override
    public double value() {
        return population;
    }
}
```

1-2.

```
package ex19_03_1;
public class Stat {
    public static String itemName(Computable com){
        return com.itemName();
    }
    public static double sum(Computable[] com){
        double sum=0;
        for(Computable c : com){
            sum += c.value();
        }
        return sum;
    }
    public static double average(Computable[] com){
        return sum(com)/com.length;
    }
}
```

1-3.

```
package ex19_03_1;
public class Exec {
    public static void main(String[] args) {
        City[] cities = {    new City("A", 100),
                             new City("B", 120),
                             new City("C", 300),
                             new City("D", 250),
                             new City("E", 85)
                             };

        double sum = Stat.sum(cities);
        double ave = Stat.average(cities);
        System.out.println(Stat.itemName(cities[0]));
        System.out.println("合計 = " + sum);
        System.out.println("平均 = " + ave);
    }
}
```