

※概要欄から提供プログラムをダウンロード！ ↓

▶ ポイント

1. 継承、スーパークラス、サブクラス
2. メソッドのオーバーライド
3. this キーワード、super キーワード



▶ 問題

1

タクシーを管理する設計図クラスとして Taxi1 クラスを定義します。Taxi1 クラスは自動車を管理する Car1 クラスを継承するものとします。Car1.java は提供プログラムをそのまま使います。

Car1 クラスと Taxi1 クラスがもつメンバ変数とメソッドは以下の表のとおりです。

Car1 クラス	
メンバ変数	
private int no	自動車のナンバー
メソッド	
public void setNo(int n)	変数 no に引数 n を設定する
public void display()	変数 no の値を表示する

Taxi1 クラス	
メンバ変数	
private int price	タクシーの料金
メソッド	
public void start()	変数 price に 420 を設定する
public void run()	変数 price に 80 を加算する
public void display()	変数 price の値を表示する

Taxi1.java を作成したら、実行用クラス DriveTaxi1 を実行します。DriveTaxi1.java は提供プログラムをそのまま使います。

実行例

```
>java DriveTaxi1  
料金は 500 円です
```

Car1.java

```
1 | public class Car1{  
2 |     private int no;  
3 |  
4 |     public void setNo(int n){  
5 |         no = n;  
6 |     }  
7 |     public void display(){  
8 |         System.out.println("ナンバーは" + no + "です");  
9 |     }  
10| }
```

DriveTaxi1.java

```
1 | public class DriveTaxi1{  
2 |     public static void main(String[] args){  
3 |         Taxi1 t1 = new Taxi1();  
4 |         t1.setNo(2525);  
5 |         t1.start();  
6 |         t1.run();  
7 |         t1.display();  
8 |     }  
9 | }
```

2

問題1で作成したTaxi1クラスにコンストラクタを追加してTaxi2クラスを作成します。Taxi1.javaをコピーしてTaxi2.javaを作成し、クラス名をTaxi2に変更してください。また、Taxi2クラスではdisplayメソッド内で自動車のナンバーも表示するように変更します。

Taxi2クラスはCar2クラスを継承するものとします。Car2.javaは提供プログラムをそのまま使います。Car2クラスとTaxi2クラスがもつメンバ変数とメソッドは以下の表のとおりです。

Car2 クラス	
メンバ変数	
private int no	自動車のナンバー
メソッド	
public Car2(int no)	変数 no に引数を設定する
public void display()	変数 no の値を表示する

Taxi2 クラス	
メンバ変数	
private int price	タクシーの料金
メソッド	
public Taxi2(int no)	変数 no に引数を設定する 変数 price に 0 を設定する
public Taxi2(int no, int price)	変数 no と変数 price に引数を設定する
public void start()	変数 price に 420 を設定する
public void run()	変数 price に 80 を加算する
public void display()	変数 no と変数 price の値を表示する

Taxi2.java を作成したら、実行用クラス DriveTaxi2 を実行します。DriveTaxi2.java は提供プログラムをそのまま使います。

実行例

```
>java DriveTaxi2
ナンバーは 2525 です
料金は 500 円です
```

Car2.java

```
1 | public class Car2{
2 |     private int no;
3 |
4 |     public Car2(int no){
5 |         this.no = no;
6 |     }
7 |     public void display(){
8 |         System.out.println("ナンバーは" + no + "です");
9 |     }
10| }
```

DriveTaxi2.java

```
1 | public class DriveTaxi2{
2 |     public static void main(String[] args){
3 |         Taxi2 t2 = new Taxi2(2525);
4 |         t2.start();
5 |         t2.run();
6 |         t2.display();
7 |     }
8 | }
```

3

次のプログラムをコンパイルして実行したとき、どのように表示されるでしょうか。

Car.java

```
1 | class Car{
2 |     Car(){
3 |         System.out.println("Car()");
4 |     }
5 |     Car(int no){
6 |         System.out.println("Car(int no)");
7 |     }
8 | }
```

Taxi.java

```
1 | class Taxi extends Car{
2 |     Taxi(){
3 |         this(9999);
4 |         System.out.println("Taxi()");
5 |     }
6 |     Taxi(int no){
7 |         super(no);
8 |         System.out.println("Taxi(int no)");
9 |     }
10 |    Taxi(String no){
11 |        System.out.println("Taxi(String no)");
12 |    }
13 | }
```

DriveTaxi.java

```
1 | class DriveTaxi{
2 |     public static void main(String[] args){
3 |         Car c = new Car();
4 |         System.out.println("-----");
5 |         Taxi t1 = new Taxi(2525);
6 |         System.out.println("-----");
7 |         Taxi t0 = new Taxi();
8 |         System.out.println("-----");
9 |         Taxi t2 = new Taxi("2525");
10 |     }
11 | }
```

実行例

>*java DriveTaxi*

?



＼フリーラーニング（無料で学べる場）をもっと広げたい！／
チャンネル登録や拡散よろしくお願いします！



NEXT DOOR

