

課題 5-1 のプログラム例

```
//車を表すクラス
class Car2 {
    private int num;
    private double gas;
    private Date buydate;
    //有効期限を表すフィールド
    private Date expiredate;
    //コンストラクタ
    public Car2() {
        num = 0; gas = 0.0;
        buydate = new Date();
        //Data インスタンスを作成する
        expiredate = new Date();
    }
    public Car2(int n, double g) {
        num = n; gas = g;
        buydate = new Date();
        //Data インスタンスを作成する
        expiredate = new Date();
    }
    //expiredate にインスタンスを参照させる
    public void setExpireDate(Date d) {
        expiredate = d;
    }
    //expiredate のインスタンスを取得する
    public Date getExpireDate() {
        return new Date(expiredate);
    }
    public Date getExpireDate2() {
        return expiredate;
    }
    public void setBuyDate(Date d) {
        buydate = d;
    }
    public Date getBuyDate() {
        return new Date(buydate);
    }
    public Date getBuyDate2() {
        return buydate;
    }
    //インスタンスの情報を出力する
    public void show() {
        System.out.println("(num)" + num
            + " (gas)" + gas);
        System.out.print("(Buy Date)");
        buydate.show();
        System.out.print("(Expire Date)");
```

```
        expiredate.show();
    }
    public void setCar(int n, double g) {
        num = n; gas = g;
    }
}

//expiredate に対する動作確認をするクラス
class Pd5car2 {
    public static void main(String[] args) {
        Date d1, d2; Car2 c1;
        d1 = new Date(2015, 5, 15);
        d2 = new Date(2017, 5, 15);

        c1 = new Car2(1234, 50); c1.show();
        c1.setBuyDate(d1);
        c1.setExpireDate(d2); c1.show();

        d1 = c1.getExpireDate2();
        d1.setYMD(2020, 5, 15); c1.show();
        d1 = c1.getExpireDate();
        d1.setYMD(2015, 4, 24); c1.show();
    }
}
```

課題 5-2～5-4 のプログラム例

```
//Data と has-A 関係をもつクラス
class Add {
    //Data のクラス型の変数を宣言する
    private Data d1;
    private Data d2;
    //d1, d2 にインスタンスを参照させる
    public void setD1(Data d) {
        d1 = d;
    }
    public void setD2(Data d) {
        d2 = d;
    }
    //d1, d2 の情報を出力する
    public void show() {
        System.out.print("(d1) ");
        d1.show();
        System.out.print("(d2) ");
        d2.show();
    }
    //d1 と d2 の参照を交換する
    public void swap() {
        Data d3;
```

```

        d3 = d1;
        d1 = d2;
        d2 = d3;
    }
    //d1 へ d2 を結合する
    public Data addD1D2() {
        Data d3;
        d3 = new Data(d1);
        d3.add(d2);
        return d3;
    }
}

//整数と文字列の値を持つクラス
class Data {
    private int num;
    private String str;
    //引数付きコンストラクタ
    public Data(Data data) {
        this(data.num, data.str);
    }
    public Data(int n, String s) {
        num = n;
        str = s;
    }
    //フィールドを出力する
    public void show() {
        System.out.printf("num: %d, str: %s\n",
                           num, str);
    }
    //フィールドを結合する
    public void add(Data d) {
        num = num + d.num;
        str = str + d.str;
    }
}

//クラス Add の動作を確認するクラス
class Pd5data1 {
    public static void main(String[] args) {
        Data ins1, ins2;
        Add add1;
        //インスタンスを生成する
        ins1 = new Data(4, "foo");
        ins2 = new Data(10, "bar");
        add1 = new Add();
        //インスタンスの参照を引数で渡す
        add1.setD1(ins1);
        add1.setD2(ins2);
        add1.show();
    }
}

//メソッド swap の動作を確認するクラス
class Pd5data2 {
    public static void main(String[] args) {
        Data ins1, ins2;
        Add add1;
        //インスタンスを生成する
        ins1 = new Data(4, "foo");
        ins2 = new Data(10, "bar");
        add1 = new Add();
        add1.setD1(ins1);
        add1.setD2(ins2);
        add1.show();
        //add1 の d1 と d2 の参照入れ替える
        add1.swap();
        add1.show();
    }
}

//メソッド addD1D2 の動作を確認するクラス
class Pd5data3 {
    public static void main(String[] args) {
        Data ins1, ins2;
        Add add1;
        //インスタンスを生成する
        ins1 = new Data(4, "foo");
        ins2 = new Data(10, "bar");
        add1 = new Add();
        add1.setD1(ins1);
        add1.setD2(ins2);
        add1.show();
        //addD1D2 でインスタンスを結合する
        Data ins3 = add1.addD1D2();
        ins3.show();
        add1.show();
    }
}

```