

復習 1 次のクラス Person にコンストラクタを宣言してください。

コンストラクタ

```
// 人クラスの宣言
class Person
{
    // 人情報
    private String name; // 名前
    private int age;     // 年齢

    // コンストラクタ
    …ここに宣言しましょう…

    // 人情報の表示
    public void showPerson(){
        System.out.println(name+"さん (" +age+"歳) ");
    }
}

class Review1
{
    public static void main(String[] args){
        // コンストラクタにより人オブジェクトを生成
        Person you = new Person("太郎",21);

        // 人情報の表示
        you.showPerson();
    }
}
```

〔実行結果〕

太郎さん (21 歳)

復習 2 次のクラス Pilot にパイロット情報を設定するメソッドを宣言してください。
メソッドのオーバーロード

```
// パイロットクラスの宣言
class Pilot
{
    // パイロット情報
    private String name; // 氏名
    private int aircraft; // 操縦航空機番号
    private double ftime; // 総飛行時間

    // パイロット情報の設定
    ...ここに宣言しましょう...

    // パイロット情報の表示
    public void showPilot(){
        System.out.println(name+" (ボーイング"+aircraft+", "+ftime+"時間) ");
    }
}

class Review2
{
    public static void main(String[] args){
        // パイロットオブジェクトの生成
        Pilot you = new Pilot();

        // パイロット情報の設定
        you.setPilot("次郎");
        you.setPilot(787);
        you.setPilot(530.5);

        // パイロット情報の表示
        you.showPilot();
    }
}
```

〔実行結果〕

次郎 (ボーイング 787, 530.5 時間)

復習 3 次のクラス Book を拡張して、クラス Quiz を宣言してください。

クラスの拡張、メンバーの継承

```
// 本クラスの宣言
class Book
{
    // 本情報
    private String title; // タイトル
    private int page;     // ページ数

    // タイトルの設定
    public void setTitle(String t){
        title = t;
    }
    // ページ数の設定
    public void setPage(int p){
        page = p;
    }
    // 本情報の表示
    public void showBook(){
        System.out.println(title+", "+page+"ページ");
    }
}

// クイズクラスの宣言
class Quiz extends Book
{
    ...ここに宣言しましょう...
}

class Review3
{
    public static void main(String[] args){
        // クイズ本の生成
        Quiz mybook = new Quiz();

        // クイズ本の情報設定
        mybook.setTitle("数独 2014");
        mybook.setPage(104);
        mybook.setQuestions(100);

        // クイズ本の情報表示
        mybook.showQuiz();
    }
}
```

〔実行結果〕

数独 2014, 104 ページ
全 100 問収録!!

復習 4 インタフェース `Building` を実装してクラス `Tower` を宣言してください。

インタフェース、メソッドのオーバーライド

```
// 建物インタフェースの宣言
interface Building
{
    // 建物名の設定
    void setName(String s);
    // 建物情報の表示
    void showBuilding();
}

// タワークラスの宣言
class Tower implements Building
{
    ...ここに宣言しましょう...
}

class Review4
{
    public static void main(String[] args){
        // タワーの生成
        Tower mytower = new Tower();

        // タワー情報の設定
        mytower.setName("東京スカイツリー");
        mytower.setHeight(634);

        // タワー情報の表示
        mytower.showBuilding();
    }
}
```

〔実行結果〕

東京スカイツリー(634m)