オブジェクト指向プログラミング　第 2 回

条件分岐，配列の基礎
フレームの表示

担当：高橋，佐藤聖也

1　条件分岐

1.1　if 文

次のプログラム **IfSample.java** は if 文を用いた条件分岐の例です．条件文中の関係演算子 `A==B` (A と B が等しい)，`A>B` ($A > B$)，`A<B` ($A < B$)，`A>=B` ($A \geq B$)，`A<=B` ($A \leq B$)，`A!=B` ($A \neq B$)，および論理演算子 `P&&Q` (P かつ Q)，`P||Q` (P または Q) の使い方を覚えておきましょう．

IfSample.java

```
class IfSample{
    public static void main(String [] args){
        int i = 1, j = 2; //j=3; としたときどうなるか？
        System.out.println("i = "+i+", j = "+j+"\n");

        System.out.println("i>2||j==2 の真偽値："+(i>2||j==2));
        if(i>2||j==2){
            System.out.println("if 文の中身が実行されます．");
        }else{
            System.out.println("else 文の中身が実行されます．");
        }
        System.out.println();
        System.out.println("i> 2&&j==2 の真偽値： "+(i>2&&j==2));
        System.out.println("i==2&&j==2 の真偽値： "+(i==2&&j==2));
        System.out.println("i< 2&&j==2 の真偽値： "+(i<2&&j==2));
        System.out.println("j!=2 の真偽値： "+(j!=2));
        if(i>2&&j==2){
            System.out.print("if(i>2&&j==2){...}");
        }else if(i==2&&j==2){
            System.out.print("else if(i==2&&j==2){...}");
        }else if(i<2&&j==2){
            System.out.print("else if(i<2&&j==2){...}");
        }else{
            System.out.print("else{...}");
        }
        System.out.println("の中身が実行されます．");
    }
}
```

1.2　switch 文

次のプログラム **SwitchSample.java** は switch 文を用いた条件分岐の例です．`switch(条件文){...}` に関して，条件文の値が `case` の直後に書かれた値に一致するとき，`case 値:` と break; の間に書かれた文が実行されます．`case 値:` から break; までは複数行にわたっていても構いませんが，break; が書かれていない場合は次の `case 値:` の範囲も実行されてしまうため，break; を書き忘れないよう注意する必要があります．

SwitchSample.java

```
class SwitchSample{
    public static void main(String [] args){
        int i = -23;
        char c; int d;
        switch(i%5){
            case 0 :
                c = ’あ’;
                d = 0;
                break;
            case 1 :
                c = ’い’;
                d = 1;
                break;
            case 2 :
                c = ’う’;
                d = 2;
                break;
            case 3 :
                c = ’え’;
                d = 3;
                break;
            case 4 :
                c = ’お’;
                d = 4;
                break;
            default:
                c = ’ん’;
                d = 100;
                break;
        }
        System.out.println("i%5 = "+i%5+" , c = "+c+" , d = "+ d);
    }
}
```

2　配列

次のプログラム **ArraySample.java** は 1 元配列と 2 元配列の使用例です．数学のベクトルや行列と同様，配列とはインデックスされた変数の組です．**ArraySample.java** では，ベクトル $\mathbf{x}$ の成分 x_i を配列 `x[i]` に対応させ，ベクトル $\mathbf{x}$ の長さ $|\mathbf{x}| := \sqrt{x_0^2 + x_1^2 + x_2^2}$ を `Lx` として求めています．

配列の定義は

```
double [ ] x = new double[配列サイズ];
```

のように書き，C 言語とは少し違っている点に注意してください．また，配列の定義と初期化を分ける場合と，配列の定義と初期化を同時に行う場合の，両方の書き方を覚えておきましょう．

ArraySample.java

```
class ArraySample{
    public static void main(String [] args){
        //配列の定義と初期化（その 1）
        double [] x = new double[3];
        x[0] = 1; x[1] = 1; x[2] = 1;
        //配列の定義と初期化（その 2）
        double [] y = {2,2,2};
        //ベクトル x の長さ：Lx = |x|
        double Lx = 0;
        for(int i=0; i<3; i++){
            System.out.print("x("+i+") = "+x[i]+",\t");
            System.out.println("y("+i+") = "+y[i]);
            Lx = Lx + x[i]*x[i];
        }
        Lx = Math.sqrt(Lx);
        System.out.println("\n"+"|x| = " + Lx + "\n");

        //2 元配列の定義と初期化（その 1）
        double [][] A = new double[2][2];
        A[0][0] = 5; A[0][1] = -3;
        A[1][0] = 2; A[1][1] = 1;
        //2 元配列の定義と初期化（その 2）
        double [][] B = {{1,2},{3,4}};
        System.out.println("行列 A");
        for(int i=0; i<2; i++){
            for(int j=0; j<2; j++){
                System.out.print(A[i][j]+"\t");
            }
            System.out.println();
        }
        System.out.println("\n"+"行列 B");
        for(int i=0; i<2; i++){
            for(int j=0; j<2; j++){
                System.out.print(B[i][j]+"\t");
            }
        System.out.println();
        }
    }
}
```

3　フレームの表示

さて，ここまでは，C 言語によるプログラミングとほとんど変わらない内容でした．ここからは，少しずつオブジェクト指向プログラミングの世界に入っていきます．Java やその他のオブジェクト指向プログラミングの教科書を開くと，次々にカタカナの術語が出てきて，ひとつの文章を理解するのも，最初は容易でないと思います．実際にプログラムを書き，イメージを持ちながら，ひとつひとつの術語の意味を理解していくことが必要になります．今回の講義ではまず,「クラス」「インスタンス」「メソッド（インスタンスメソッド)」という言葉を覚えてください．

「クラス」と「インスタンス」の意味をつかむためには，フレーム生成がよい例題を提供してくれます．

次の **FrameSample01.java** は，GUI(Graphical User Interface) の基本となる，フレームを表示させるプログラムです．ここには，ディスプレイ上に特定の大きさのフレームを表示させるためのコードしか書かれていないため，終了ボタン（×ボタン）は機能しません（最大化・最小化ボタンは機能します）し，メニューボタンなども存在しません．表示されたフレームを終了する際は，Eclipse のコンソール右上に現れる赤い四角をクリックして（コマンドプロンプトで実行中の場合は Ctrl+C を押して），**FrameSample01.java** を強制終了してください．

FrameSample01.java

```
import java.awt.Frame;
class FrameSample01{
    public static void main(String [] args){

        //フレームクラスのインスタンスを生成
        Frame fr = new Frame();

        //フレームサイズの指定（横，縦）[ピクセル]
        fr.setSize(500, 400);
        //フレームタイトルの表示
        fr.setTitle("フレームのタイトルが書けます！");
        //フレームをディスプレイ画面に表示する
        fr.setVisible(true);

        //このフレームの×ボタンは無効なので
        //Ctrl+C を押してコマンドプロンプトから強制終了します.
    }
}
```

ここでは，Frame クラスのインスタンス生成

```
Frame fr = new Frame();
```

および

```
fr.setSize(500,400);
```

などの，メソッド（インスタンスメソッド）の呼び出し方を覚えましょう．
※「クラス」「インスタンス」「メソッド（インスタンスメソッド)」については，補足資料を用いて説明します．

練習問題

フレームタイトルを「フレーム A」「フレーム B」とした，大きさの異なる 2 つのフレームを生成するプログラム **FrameSample02.java** を作成し実行せよ．

課題

1. 行列 A, B に対して，和 $A+B$, 積 AB および BA を計算するプログラム Lec02Kadai_**00rd000.java** を作成し提出しなさい．ただし，クラス名／ファイル名の **00rd000** は各自の学籍番号とする．大文字・小文字を間違えないよう注意すること．また，

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, \qquad B = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 0 & 2 & 3 \end{pmatrix}$$

とする．

提出期限：5/25の1限目開始時間まで

提出場所：https://tdu.app.box.com/f/a501ea6e8a8949a0830292736cf75bf4