

# プログラミング入門1

第6回

Switch文

プロジェクトの持ち運び

# 授業開始前に

ログオン後、

不要なファイルを削除し  
て待機してください

# 前回のテーマ

- while 文を用いた繰り返し実行
  - for文との使い分け
- 複雑な条件判定
  - &&「かつ」 ||「または」 を使って

復習： while文はfor文から「初期化式」を外に出し、「ステップを進める式」を 繰り返し実行される文に移したものの

```
for (int i = 1; i < 10; i++) {  
    System.out.println("i = " + i);  
}
```

```
int i = 1;  
while ( i < 10 ) {  
    System.out.println("i = " + i);  
    i++;  
}
```

# 復習： for文、while文それぞれの使いどころ

ループに入る時点で繰り返しの回数が決まっているときは

```
...  
for (int i = 0; i < n; i++){  
    System.out.println("Hello");  
}
```

繰り返しの回数が指定できないときは番兵 (sentinel) をおく

```
...  
int input;  
input = Integer.parseInt(reader.readLine());  
while (input != 0) {  
    System.out.println(input);  
    input = Integer.parseInt(reader.readLine());  
}
```

ステップを進める式 (ループ制御変数の値を更新する式) はいろいろな姿をとる

# 復習： for文に相応しい問題

10000円を利率5%の複利で預金した際の、10年後の預金額を計算する

```
// 元本
double amount = 10000;
// 10回繰り返す (0年後から始めて; 10年後まで; 1年ずつ考える)
for (int year = 0; year < 10; year++) {
    amount *= 1.05;
}
System.out.println("10年後は" + amount + "円");
```

## 実行結果

10年後は16288.946267774418円

# 復習: while文に相応しい問題

10000円を利率5%の複利で預金した際に、預金額が20000円を超えるまでの年数を計算する

```
double amount = 10000; // 元本
int year = 0; // 開始した時点では0年後

// 20000円を超えるまで繰り返す (= 20000円以下のうちは繰り返す)
// → 20000円以下であることを監視する番兵を置く

while (amount <= 20000) {
    // 1年後に利子をつける
    year++;
    amount *= 1.05;
}

System.out.println(year + "年後に20000円を超える");
```

実行結果

15年後に20000円を超える

# 今回のテーマ

- switch 文
  - 主にbreak文とともに
  - 条件分岐のもうひとつのやり方
- for文, while文におけるbreak 文
  - ループ抜け出しの非常手段
- プロジェクトの持ち運び
  - Eclipseの機能から
  - export
  - import

# 多くの選択肢からひとつを選んで実行する

## こんな形でよく使う

switch文を使うと



```
i f (n==2)
  B1
el se i f (n==3)
  B2
el se i f (n==5)
  B3
el se i f (n==7)
  B4
el se
  B5
```

```
swi tch(n){
  case 2:
    B1
    break;
  case 3:
    B2
    break;
  case 5:
    B3
    break;
  case 7:
    B4
    break;
  defaul t:
    B5
}
```

# 多くの選択肢からひとつを選んで実行する

ときどき使う形

break文がなければ、条件に合致するcaseに飛んで以後のすべての文が実行される

```
if (n==2){
    B1
    B2
    B3
    B4
    B5
}
else if (n==3){
    B2
    B3
    B4
    B5
}
else if (n==5){
    B3
    B4
    B5
}
else if (n==7){
    B4
    B5
}
else
    B5
```

```
switch(n){
    case 2:
        B1
    case 3:
        B2
    case 5:
        B3
    case 7:
        B4
    default:
        B5
}
```

if-else文  
を使って同じ  
ことができる。

# switch 文の構造と意味

```
switch (整数型の式) {  
  case 整数型の定数:  
    文の列  
  case 整数型の定数:  
    文の列  
  ...  
  default:  
    文の列  
}
```

caseラベル  
と呼ばれる

- まず「整数型の式」の値が計算される。
- その値と等しい「整数型の定数」の書かれているcaseが選ばれて、そのcaseの文の列が実行される。このようなcaseがなければ、defaultの文の列が実行される。
- break文があるとswitch文をぬける
- もしbreak文がなければ次のcaseの文の列が実行される。

コロン(セミコロンではない)

# switchの例

```
public class Modular3 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力してください:");
        int input = Integer.parseInt(reader.readLine());

        switch (input) {
            case 0:
                System.out.println("0が入力されました");
                break;
            case 1:
                System.out.println("入力された値は1");
                break;
            case 2:
                System.out.println("2です");
                break;
            default:
                System.out.println("0でも1でも2でもありません");
        }
    }
}
```

# input == 0 の場合は

```
public class Modular3 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力してください:");
        int input = Integer.parseInt(reader.readLine());

        switch (input) {
            case 0:
                System.out.println("0が入力されました");
                break;
            case 1:
                System.out.println("入力された値は1");
                break;
            case 2:
                System.out.println("2です");
                break;
            default:
                System.out.println("0でも1でも2でもありません");
        }
    }
}
```

# input == 1 の場合は

```
public class Modular3 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力してください:");
        int input = Integer.parseInt(reader.readLine());

        switch (input) {
            case 0:
                System.out.println("0が入力されました");
                break;
            case 1:
                System.out.println("入力された値は1");
                break;
            case 2:
                System.out.println("2です");
                break;
            default:
                System.out.println("0でも1でも2でもありません");
        }
    }
}
```

# input == 2 の場合は

```
public class Modular3 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力してください:");
        int input = Integer.parseInt(reader.readLine());

        switch (input) {
            case 0:
                System.out.println("0が入力されました");
                break;
            case 1:
                System.out.println("入力された値は1");
                break;
            case 2:
                System.out.println("2です");
                break;
            default:
                System.out.println("0でも1でも2でもありません");
        }
    }
}
```

# input の値が0,1,2以外の場合

```
public class Modular3 {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力してください:");
        int input = Integer.parseInt(reader.readLine());

        switch (input) {
            case 0:
                System.out.println("0が入力されました");
                break;
            case 1:
                System.out.println("入力された値は1");
                break;
            case 2:
                System.out.println("2です");
                break;
            default:
                System.out.println("0でも1でも2でもありません");
        }
    }
}
```

# caseを重ねることもある

```
public class MultiCase {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("整数を入力してください:");
        int input = Integer.parseInt(reader.readLine());
        switch (input % 10) {
            case 0:
                System.out.println("10の倍数です");
                break;
            // 10 で割った余りが 1 または 9
            case 1:
            case 9:
                System.out.println("もう少しで10の倍数でした");
                break;
            case 5:
                System.out.println("10の倍数ではありませんが5の倍数です");
                break;
            default:
                System.out.println("10の倍数ではありません");
        }
    }
}
```

# ループ抜け出しのbreak文

## valueが素数か判定する

```
int divisor = 0;
for (int i = 2; i < value; i++) {
    if (value % i == 0) {
        divisor = i;
        break;
    }
}

if (divisor==0)
    System.out.println(value + " is prime");
else
    System.out.println(value + " is divisible by " + divisor);
```

素朴な素数判定法である。

break文は途中でのループ抜け出しに使える。

## 2からvalue-1までの数で順に割ってみればよいが

```
int divisor = 0;
for (int i = 2; i < value; i++) {
    if (value % i == 0) {
        divisor = i;
        break;
    }
}

if (divisor == 0)
    System.out.println(value + " is prime");
else
    System.out.println(value + " is divisible by " + divisor);
```

## ある i で割り切れてしまったら

```
int divisor = 0;
for (int i = 2; i < value; i++) {
    if (value % i == 0) {
        divisor = i;
        break;
    }
}

if (divisor==0)
    System.out.println(value + " is prime");
else
    System.out.println(value + " is divisible by " + divisor);
```

divisorに割り切った数iを代入してループを打ち切る

# 判定

```
int divisor = 0;
for (int i = 2; i < value; i++) {
    if (value % i == 0) {
        divisor = i;
        break;
    }
}

if (divisor==0)
    System.out.println(value + " is prime");
else
    System.out.println(value + " is divisible by " + divisor);
```

divisor==0なら割り切るiが無かったということ。

divisor!=0なら割り切るiがあったということ。

# 一緒にやってみよう

- 今回の演習で使うテストドライバをいつものようにインストールする
  - テストドライバの導入に成功すると
    - プロジェクト「java20XX」の中の「test」というフォルダに「j1.lesson06.xml」という名前のファイルが作成される。
    - このファイルには今週使用するテスト一式が記述されている。
- j1.lesson06 というパッケージを作成する
- 講義資料にあるId2Classというプログラムを、このパッケージに作成する
  - 講義資料にある手順でテスト、実行までやること
- 講義資料にあるプロジェクトの持ち運びの解説をみてzipファイルを作成せよ。

# Id2Classの解説

```
public class Id2Class {
    public static void main(String[] args) throws IOException {
        BufferedReader reader =
            new BufferedReader(new InputStreamReader(System.in));
        System.out.println("学籍番号の下4桁を入力:");
        int id = Integer.parseInt(reader.readLine());
        // id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号
        switch (id / 100) {
            case 0:
                System.out.println("Aクラス");
                break;
            case 1:
                System.out.println("Bクラス");
                break;
            case 10:
                System.out.println("Cクラス");
                break;
            case 11:
                System.out.println("Dクラス");
                break;
            default:
                System.out.println("不明なクラス");
        }
    }
}
```

# 入力の準備から学生番号(4桁)の入力まで

```
public class Id2Class {  
    public static void main(String[] args) throws IOException {  
        BufferedReader reader =  
            new BufferedReader(new InputStreamReader(System.in));  
        System.out.println("学籍番号の下4桁を入力:");  
        int id = Integer.parseInt(reader.readLine());  
        // id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
        switch (id / 100) {  
            case 0:  
                System.out.println("Aクラス");  
                break;  
            case 1:  
                System.out.println("Bクラス");  
                break;  
            case 10:  
                System.out.println("Cクラス");  
                break;  
            case 11:  
                System.out.println("Dクラス");  
                break;  
            default:  
                System.out.println("不明なクラス");  
        }  
    }  
}
```

# 学生番号(4桁)から上位2桁を取り出す

$$\textcircled{n_3n_2n_1n_0} / 100 \longrightarrow n_3n_2$$

```
.....  
// id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
switch (id / 100) {  
    case 0:  
        System.out.println("Aクラス");  
        break;  
    case 1:  
        System.out.println("Bクラス");  
        break;  
    case 10:  
        System.out.println("Cクラス");  
        break;  
    case 11:  
        System.out.println("Dクラス");  
        break;  
    default:  
        System.out.println("不明なクラス");  
}
```

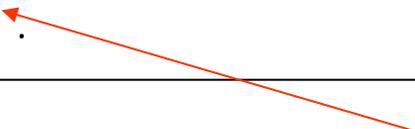
# 上位2桁が 00の場合

```
.....  
// id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
switch (id / 100) {  
    case 0:  
        System.out.println("Aクラス");  
        break;  
    case 1:  
        System.out.println("Bクラス");  
        break;  
    case 10:  
        System.out.println("Cクラス");  
        break;  
    case 11:  
        System.out.println("Dクラス");  
        break;  
    default:  
        System.out.println("不明なクラス");  
}  
.....
```

Java 1 第6回  
break文が実行されるとここへ

# 上位2桁が 01 の場合

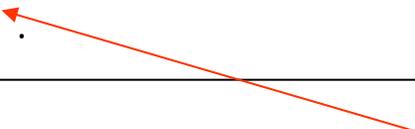
```
.....  
// id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
switch (id / 100) {  
    case 0:  
        System.out.println("Aクラス");  
        break;  
    case 1:  
        System.out.println("Bクラス");  
        break;  
    case 10:  
        System.out.println("Cクラス");  
        break;  
    case 11:  
        System.out.println("Dクラス");  
        break;  
    default:  
        System.out.println("不明なクラス");  
}  
.....
```



Java 1 第6回  
break文が実行されるとここへ

# 上位2桁が 10 の場合

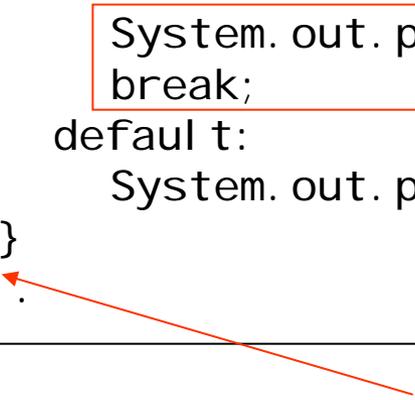
```
.....  
// id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
switch (id / 100) {  
    case 0:  
        System.out.println("Aクラス");  
        break;  
    case 1:  
        System.out.println("Bクラス");  
        break;  
    case 10:  
        System.out.println("Cクラス");  
        break;  
    case 11:  
        System.out.println("Dクラス");  
        break;  
    default:  
        System.out.println("不明なクラス");  
}  
.....
```



Java 1 第6回  
break文が実行されるとここへ

# 上位2桁が 11 の場合

```
.....  
// id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
switch (id / 100) {  
    case 0:  
        System.out.println("Aクラス");  
        break;  
    case 1:  
        System.out.println("Bクラス");  
        break;  
    case 10:  
        System.out.println("Cクラス");  
        break;  
    case 11:  
        System.out.println("Dクラス");  
        break;  
    default:  
        System.out.println("不明なクラス");  
}  
.....
```



Java 1 第6回  
break文が実行されるとここへ

# その他の場合

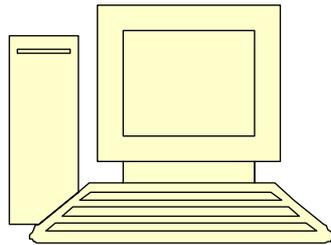
```
.....  
// id = 学科番号 * 1000 + クラス番号 * 100 + 2桁の出席番号  
switch (id / 100) {  
    case 0:  
        System.out.println("Aクラス");  
        break;  
    case 1:  
        System.out.println("Bクラス");  
        break;  
    case 10:  
        System.out.println("Cクラス");  
        break;  
    case 11:  
        System.out.println("Dクラス");  
        break;  
    default:  
        System.out.println("不明なクラス");  
}  
.....
```

Java 1 第6回  
switchが終了してここへ

# プロジェクトの持ち運び 大学から自宅へ

export/importの詳細については第1回講義資料を参照

ラボ教室の作業環境



Eclipse

↓ プロジェクトをexport

java2005.zip

zipファイル

個人の作業環境(自宅など)



Eclipse

↑ プロジェクトをimport

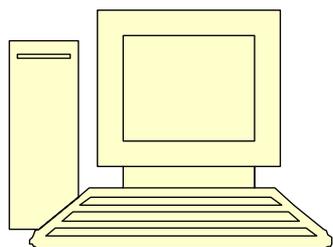
java2005.zip

zipファイル

← Eメール添付で →

# 自習した成果を大学の環境に持ち込む

ラボ教室の作業環境



Eclipse

プロジェクトをimport

java2005.zip

zipファイル

個人の作業環境(自宅など)



Eclipse

プロジェクトをexport

java2005.zip

zipファイル

Eメール添付で

# 課題

各自のペースで  
「第06週目の課題」をやってみよう

# 課題0601のヒント

入力する

```
if (入力された数が0以上){  
    switch( 入力された数の1の位){  
        ... 場合わけする....  
    }  
else  
    ....
```

注 1の位は10で割った余り

# 課題0602のヒント

1000円札、5000円札、10000円札の枚数を数える変数を準備

入力する

```
while(入力された数が0でない){
    switch(入力された数){
        case 1000:
            .....
            break;
        .....
        default:
            "お札ではありません" と表示
    }
    入力する
}
```

枚数、合計金額の出力など

# 課題0603のヒント

max, min を保持する変数、入力のための変数などを用意する  
初期値を慎重に吟味する必要がある。

入力する

```
while (入力が0以上100以下){  
    入力とmaxの比較をして必要ならmaxを更新  
    入力とminの比較をして必要ならminを更新  
    入力する  
}
```

結果を表示する

(maxの初期値を-1にしておく  
入力が無かった場合の対応も可能)