

第一週

Javaの概要、オーバーロード

Javaについて

- Javaは1995年にSunMicrosystems社で開発されたプログラミング言語。
- オブジェクト指向言語。
- 仮想マシン上で実行される。
- 中間コードをJVMがネイティブコードに変換するのでプラットフォームに依存しない。
- CやC++と似たような構文を引き継いでいる。
- 高いセキュリティやネットワーク機能。

Javaについて

- プログラミング言語のシェア数のランキング

TIOBE index

http://www.tiobe.com/tiobe_index

- Javaはウェブページで検索してヒットするページ数が1位!

いつ使うのか？

- 情報工学PBL:システム開発演習(2年後期)
- コンピュータ知能学実験(3年前期)
- オブジェクト指向言語(3年前期)
- オブジェクト指向言語応用演習(3年前期)
- 情報システム学実験(3年後期)
- 視覚情報処理応用演習(3年後期)
- その他:研究室など

開発環境

- JavaのIDE(統合開発環境)
 - Eclipse...図書館やC棟のパソコンに入ってる。
 - NetBeans...R棟の演習室のパソコンに入っている。
- 今回はEclipseを使います(演習室のバージョンは4.4)。



Sample1.java

```
public class Sample1 {  
    public static void main(String[] args) {  
        System.out.println("Hello Java World !");  
    }  
}
```

Sample1.java

```
public class Sample1 {  
    public static void main(String[] args) {  
        System.out.println("Hello Java World!");  
    }  
}
```

publicなSample1というクラスの始まり
publicとクラスについては今度やります。

Sample1.java

```
public class Sample1 {  
    public static void main(String[] args) {  
        System.out.println("Hello MPC !");  
    }  
}
```

C言語でいうところのmain関数。

Javaでは関数ではなくてメソッドと呼ぶ。

つまりこれはSample1クラスのもっているmainメソッド。

Sample1.java

```
public class Sample1 {  
    public static void main(String[] args) {  
        System.out.println("Hello MPC !");  
    }  
}
```

C言語でいうところのprintf()

System.out.println()は必ず改行するので改行したくない場合は

System.out.print()を使う

詳しくは後程

ファイルとクラスに関して注意点

- ファイル名とクラス名は一緒にする。
- 一つのjavaファイルには最低一つのクラスを実装する。

変数

Javaの変数には次のようなものがあります。

型	サイズ
boolean	Trueまたはfalseの二値
char	2バイト文字(\u0000～\uffff)
byte	1バイト整数(-128～127)
short	2バイト整数(-32768～32767)
int	4バイト整数(-2147483648～2147483647)
long	8バイト整数(-9223372036854775808～9223372036854775807)
float	4バイト単精度浮動小数点数
double	8バイト倍精度浮動小数点数

変数

Javaの変数には次のようなものがあります。

型	サイズ
→ boolean	Trueまたはfalseの二値
char	2バイト文字(\u0000～\uffff)
→ byte	1バイト整数(-128～127)
short	2バイト整数(-32768～32767)
int	4バイト整数(-2147483648～2147483647)
long	8バイト整数(-9223372036854775808～9223372036854775807)
float	4バイト単精度浮動小数点数
double	8バイト倍精度浮動小数点数

boolean型

- trueとfalseの真理値を持つ。
- C言語でゲームを作るときにflag変数をint型で持っていたがJavaにはboolean型があるのでこちらを使う。

long型について注意

C言語ではlong型の変数に値を代入するときにはint型と同じように代入できたがJavaでは

long var = 1000L;

という風に代入する値の後ろにLをつけなければなりません。

System.out.println()

- Javaの標準出力
- %dや%fなどの変換指定文字は使わない。

```
System.out.println("文字列"+変数名...);
```

このように文字列と変数名を'+'で連結して出力する。変数の後にさらに文字列や変数を持ってきたい場合も'+'で連結すればいい。

四則演算、制御文

- Javaにも+, -, *, /, %などの演算子がある
- 制御文はif, for, while, switch-case, do-whileなどC言語とさほど変わらない

Sample2.java

```
public class Sample2 {  
  
    public static void main(String[] args) {  
        int a = 1+1;  
        int b = 10-8;  
        int c = 20*8;  
        int d = 100/20;  
        int e = 3%2;  
        System.out.println("1+1="+a);  
        System.out.println("10-8="+b);  
        System.out.println("20 × 8="+c);  
        System.out.println("100/20="+d);  
        System.out.println("3/2の余り="+e);  
    }  
}
```

C言語との違い(制御文)

for文の中で変数を宣言できる。

```
for(int i=0;i<10;++i){
```

while文の無限ループはwhile(1){ではなくwhile(true){

```
while(true){
```

Sample3.java

```
public class Sample3 {  
    public static void main(String[] args) {  
        int a=0;  
        for(int i=0;i<10;++i){  
            System.out.println("i="+i);  
        }  
        while(true){  
  
            switch(a){  
                case 0:
```

for文の中で変数が宣言できるが...

JavaはC言語と違い変数をいつでもどこでも宣言することができます。

for文やwhile文の中でもその限りではありません。ただし、注意すべき点がfor文の中で宣言した変数はそのfor文の中だけでしか使えません。while文も 마찬가지です。

キャスト

- キャストとは現在の型から違う型への変換操作のこと。
- 型変換には暗黙の型変換と明示的な型変換がある。

キャスト

Javaにおける型変換は以下のように行います。

(変換したい型名)変数名

この型変換は明示的な型変換です。

明示的な型変換と暗黙の型変換

int型からlong型
float型からdouble型

この二つの型変換は変換後の型のサイズが変換前より大きいのでそのまま代入することができます。これを暗黙の型変換といいます。

明示的な型変換と暗黙の型変換

long型からint型
double型からfloat型

これらの型変換は先ほどの型変換と違い明示的な型変換が必要となります。

変換後の型が変換前の型のサイズより小さいので明示的な型変換が必要になります

実数型から整数型

小数点を含んだ実数から整数へのキャストは明示的な型変換が必要になります。

45.2  45

明示的な型変換を行うと四捨五入されます。

配列

Javaの配列は以下のように宣言します。

```
int[] a = new int[5];
```

変に思うところが数か所あると思いますが今回はこういうものだと思って使ってください。

String型

JavaにはString型と呼ばれる型があります。これは文字列を扱う型です。C言語では文字列を扱う際にはchar型の配列を使っていました。しかし、String型を使うと以下のように書くことができます。

```
String str = “こんにちは”;  
System.out.println(str);
```

C言語では2バイト文字に対応していませんでしたがJavaでは2バイト文字も扱えます。

オーバーロード

Cでは同じ名前の関数は実装できませんでした。

- `void function(double a); //function関数の宣言`
- `int function(int a); //エラー:既に同名の関数が存在!`

オーバーロード

Javaでは条件を満たせば同じ名前の関数(メソッド)を複数実装することが出来ます。これをオーバーロード(多重定義)といいます。

- `int function(int a)` //function関数の宣言
- `void function(double a)` //同名の関数を宣言可能

オーバーロード出来る条件

- 引数の個数が異なる
- 引数の型が異なる
- 引数の順番が異なる

以上のいずれかを満たせばOK！

オーバーロード

```
public class Sample4{  
    void method(int a){}  
    /*char method(int a){}*///返り値の型だけが違うのはNG  
    char method(float a){}///引数が違うとき返り値が違ってても  
OK  
    void method(double a){}///引数の型が違うのでOK  
    /*void method(int b){}*///仮引数名だけが違のはNG  
    void method(int a, int b){}///引数の数が違うのでOK  
    //次の2つは引数の順番が違うのでOK  
    void method(String s, char c){}  
    void method(char c, String s){}  
}
```

演習問題

- 九九表を表示するプログラムを作成してください。
- 足し算した結果を返す sum 関数をそれぞれのデータ型で実装してください。
- バブルソートをする関数を実装して昇順で表示してください。