**第2章　プログラムの書き方とJava変数**

この章では、プログラムの書き方と、Javaの変数、宣言について説明します。

**1.プログラムの書き方**

Javaでは、プログラムを「クラス(class)」という単位で作成します。

プログラムを書くにあたり、まずクラスを構成する要素、プログラムの書き方など基本的なルールを確認しましょう。

ここでは、第1章で作成したプログラムを例に見ていきましょう。

[MyClass.java]

1. package JSample;
2. public class MyClass {
3. public static void main(String[] args) {
4. System.out.println("Hello World");
5. }
6. }

**1-1.パッケージ文**

まず1行目は[パッケージ文]といいます。プログラムが含まれているパッケージの名前を書き、プログラムの先頭行にkならず書く必要があります。

**1-2.クラス宣言**

Javaでは、プログラムを「クラス」という単位で作成します。2行目はクラスが始まる部分で、「クラス宣言」といいます。

書かれている[public class]はクラス宣言のキーワードで、[MyClass]がクラスの名前です。

クラスの名前は自由に決めることができますが、先頭は必ず「英字の大文字」で書きます。

**{・・・ }** で囲まれた範囲をブロックといい、2行目最後のから6行目までの中にクラスの具体的な内容が書かれています。

**1-3.メソッド**

プログラムの中身にあたる３，４，５行の部分を[メソッド]といいます。クラスの中にはメソッドをいくつでも書くことができます。

3行目が[メソッド宣言]です。このメソッドの名前は[main]で、プログラムの開始点を表す特別なメソッドです。プログラムの開始点に[main]以外の名前を使うことはできません。

**1-4.命令文**

メソッドの中で、具体的な指示（処理を実行する）を書くのが[命令文]です。

命令文の最後には、セミコロン(**;**)を付けます。このセミコロンが命令文の終わりを示す記号となります。

ここでは4行目が命令文にあたり、「"Hello World"と出力しなさい」という指示を出しています。

このように、クラスの中にメソッドを書き、その中に命令文を書くのがJavaの標準的な書き方です。

**2.Javaの基本のデータ型と変数**

**2-1.Javaのデータ型**

Java言語ではデータの種類を厳密に区分して使用します。それはデータ処理を行うコンピュータにとって重要な概念であるためです。

Java言語では、データの種類のことを[データ型]といいます。

「1」や「10」は数値データ、「あ」や「A」は文字データ、「こんにちは」や「Hello」は文字列データというように、データ型にはいくつか種類があります。

下でご説明しますが、変数を宣言するときにはデータ型を指定する必要があります。

データ型について次のように表にまとめました。

値の範囲については覚える必要はありませんが、それぞれのデータ型でどのようなデータを扱えるのかを理解するようにしましょう。

|  |  |  |  |
| --- | --- | --- | --- |
| **種別** | **型** | **説明** | **値の範囲** |
| 整数型 | byte | 整数(1バイト) | -128～127 |
| short | 整数(2バイト) | -32768～32767 |
| int | 整数(4バイト) | -2147483648～2147483647 |
| long | 整数(8バイト) | -9223372036854775808 ～ 9223372036854775807 |
| 浮動小数点型 | float | 小数点のある数字  (4バイト) | 1.4×10の-45乗，3.4028235×10の38乗 |
| double | 小数点のある数字  (8バイト) | 4.9e-324(最小) ～ 1.7976931348623157e308(最大) |
| 論理型 | boolean | 真、偽を表す | true またはfalse |
| 文字型 | char | 文字(2バイト) | 1文字 |
| 文字列型 | String | 文字列 | 扱えるデータの長さは整数型と同じ（約21億） |

**2-2.Java変数**

プログラムでは、変数を使って処理を書きます。そのため変数の基本的な使い方を理解することが重要となります。

ここでいう[変数]とは値を格納するための箱のようなもので、数学で「x」や「y」を変数といって使ったのとほぼ同じモノです。

**2-2-1.変数の宣言**

変数を使うためには、使う変数の「型」と［名前」を宣言します。

・宣言の方法は以下のようになります。

データ型　変数名；

→末尾にはセミコロンが必須。

**例：**

String myName ;  
 この宣言は「int型」の値を格納できる「myName」という名前の箱を準備したということを表します。

・変数を宣言した後に初めて値を代入することができます。

**例：**

String myName ;  
myName = "yamada" ;

この宣言は「String型」の値を格納できる「myName」という名前の箱を準備し、その箱に「yamada」というString型の値を代入する、という指示を出したことを表します。

初期値は、以下のようにして代入しても構いません。

データ型　変数名　= 初期値；  
**例：**

String myName = "yamada";

変数は、宣言したデータ型に対応する値しか代入することはできません。

それぞれのデータ型の宣言の具体的な方法は以下の通りとなります。

* **int** - 123や-123のように、整数（整数）を小数点なしで格納します。

**例：** int myNum = 123;

* **float** - 浮動小数点数を、19.99や、-19.99などの小数で格納します。

**例：** float myFloatNum = 12.34f;

* **char** - 「a」や「A」などの単一文字を格納します。Char値は一重引用符で囲まれています。

**例：** char myLetter = 'a';

* **boolean** - **2つ**の状態を持つ値を格納します。**true**または**false**

**例：** boolean isTrue = true;

* **String** - "Hello"などのテキストを保存します。文字列値は二重引用符で囲まれています。

**例：** String myText = "Hello";

（以上は変数の宣言と初期化が同時に行われます。）

**例**

[JSample2\_1.java]

1. package JSample;
2. public class JSample2\_1 {
3. public static void main(String[] args) {
4. int a;
5. int b;
6. a = 10;
7. b = 20;
8. int result = a + b;
9. System.out.print( " 結果は、" + result );
10. }
11. }

**実行結果**

結果は、30

* 9行目のprint(~)メソッドは、**（）**内の変数を表示するために使用します。
* テキストを出力する場合は、**" "**で囲う必要があります。
* 9行目カッコ内のテキストと変数の両方を組み合わせるには、 **+** 文字を使用します。

**2-2-2.変数の有効範囲**

1-2.クラスの宣言でも取り上げましたが、Javaコードにはたびたび **{ }**が登場します。

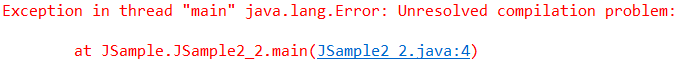
**{** ・・・**}** を「ブロック」と呼び、変数の有効範囲はブロックによって決まります。  
変数の有効範囲は、ブロックの内側となり、ブロックの内側に定義した変数は、そのブロックの外側では無効となります。

**例**

[JSample2\_2.java]

1. package JSample;
2. public class JSample2\_2 {
3. public static void main(String[] args) {
4. int one = 1;
5. int two = 2;
6. }
7. System.out.println("one:" + one);
8. System.out.println("two:" + two);
9. }

**実行結果**



例のように int型の[one],[two]という変数は、3行目から6行目までのブロックの中で宣言された変数です。

しかし、println()による出力の命令文は3-6行目のブロックより外の、7行目と8行目に書かれています。

この場合、[one],[two]という変数の有効範囲外で出力指示を行っているため、実行結果はエラーとなります。

**練習**

**問題**[JEx2\_1.java]

1-1.構文の例の変数を定義、表示してください。

* package JSample;
* public class JEx2\_1 {
* public static void main(String[] args) {
* //\*\*例：\*\* int myNum = 123;
* //System.out.println("変数str は " + str + " です");
* }
* }

**実行結果**  
変数myNumは 123 です  
変数myFloatNum は 12.34 です  
変数myLetter は a です  
変数isTrue は true です

変数myText は Hello です