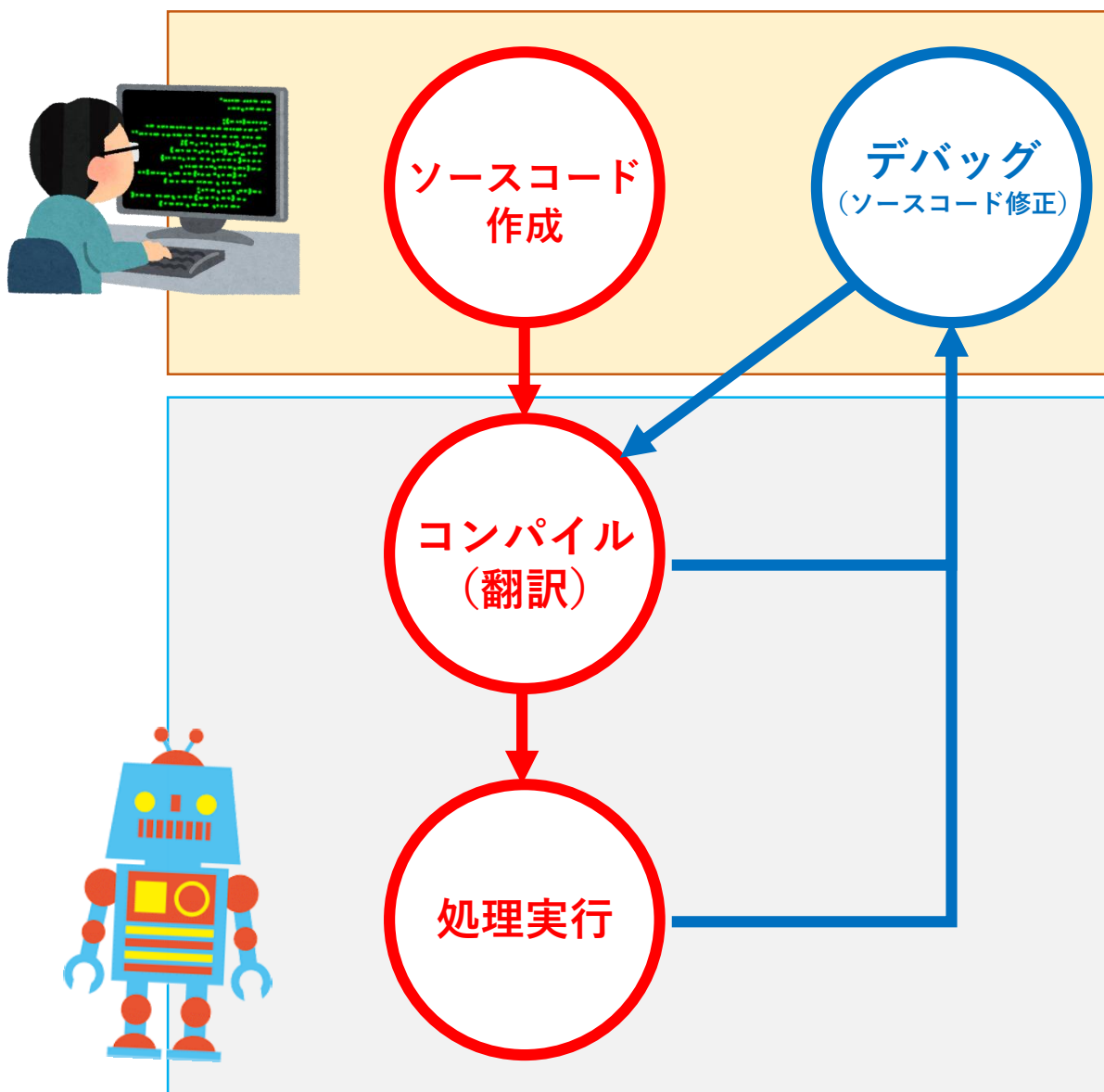




ウズウズカレツジ プログラマーコース

Javaプログラミングの手順



《プログラミングの手順》

□開発者は以下の作業でプログラミングを行います。

- ①ソースコード作成
- ②コンパイル
- ③処理実行

□ソースコードはプログラマーが書くコンピュータへの命令文のことです。

□ソースコードはプログラマーというあくまで人間が理解できる文字列で記述されたもので、そのままではコンピュータは理解することができず実行もできません。書いたソースコードはコンピュータに実行させる前に必ずコンピュータ語に翻訳してやつ必要があり、この翻訳作業のことを**コンパイル**と言います。

□コンパイルや処理実行のタイミングで発覚したソースコードのミスを修正することを**デバッグ**と言います。



《ソースコードの作成》

- ソースコードはテキストエディタや統合開発環境(後述)を用いて作成します。
- ソースファイルは任意の作業フォルダに格納します。今回の場合は「C:¥WorkSpace」です。



```
コマンド プロンプト
Microsoft Windows [Version 10.0.16299.431]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\uzuz>
```

カレントディレクトリ
(初期状態)

《ソースコードの実行》

- **コマンドプロンプト**は命令文（コマンド）を入力することでWindowsコンピューターに様々な処理を実行させることが可能な、人間とPCで対話するためのツールです。
- コマンドプロンプトの起動後、以下の3ステップでソースコードの実行が可能です。

① カレントディレクトリの変更

※以下をコマンドプロンプト上で実行

```
cd C:\Workspace
```

② コンパイル

※以下をコマンドプロンプト上で実行

```
javac Sample1_03_1.java
```

③ 実行

※以下をコマンドプロンプト上で実行

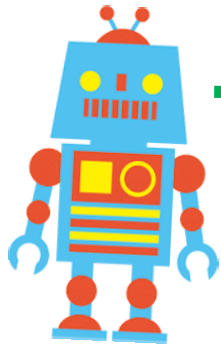
```
java Sample1_03_1
```

～ソースコード実行の3ステップ～

①カレントディレクトリの変更

カレントディレクトリ（コンピュータのアクセス先のフォルダ）をこれから実行するファイルが格納されているフォルダに切り替えます。

```
cd C:¥Workspace
```



アクセス先



Windows 8 (C:)



Workspace



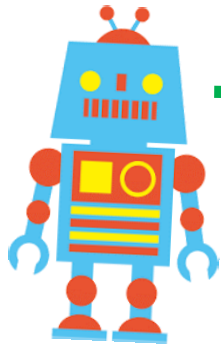
Sample1_03_1.java

～ソースコード実行の3ステップ～

①カレントディレクトリの変更

カレントディレクトリ（コンピュータのアクセス先のフォルダ）をこれから実行するファイルが格納されているフォルダに切り替えます。

```
cd C:¥Workspace
```



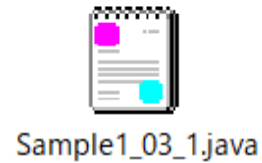
アクセス先



Windows 8 (C:)



Workspace



Sample1_03_1.java



Sample1_03_1.class

ソースコードファイル
人間が理解できる
文字列で記述

生成

バイトコードファイル
コンピュータが理解できる
文字列で記述

②コンパイル

ソースコードファイルを読み込み、コンピュータが理解できる文字列に翻訳したファイル（バイトコードファイル）を生成する。

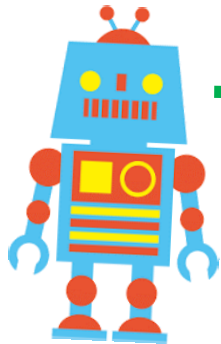
```
javac Sample1_03_1.java
```

～ソースコード実行の3ステップ～

①カレントディレクトリの変更

カレントディレクトリ（コンピュータのアクセス先のフォルダ）をこれから実行するファイルが格納されているフォルダに切り替えます。

```
cd C:¥Workspace
```



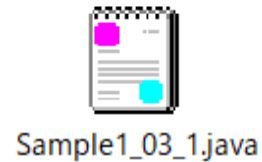
アクセス先



Windows 8 (C:)



Workspace



Sample1_03_1.java



Sample1_03_1.class

生成

ソースコードファイル
人間が理解できる
文字列で記述

バイトコードファイル
コンピュータが理解できる
文字列で記述

②コンパイル

ソースコードファイルを読み込み、コンピュータが理解できる文字列に翻訳したファイル（バイトコードファイル）を生成する。

```
javac Sample1_03_1.java
```

③実行

コンピュータにバイトコードファイルを実行させる。
実行で指定するファイル名に拡張子は不要。

```
java Sample1_03_1
```

実行

～ソースコード概要～

Sample1_03_1.java

```
1 class Sample1_03_1 {  
2     ^    public static void main (String[] args) {  
3     ^    ^    ^  
4     ^    ^    System.out.println("HelloWorld!!");  
5     ^    ^    ^  
6     ^    }  
7 }
```


～ソースコード概要～

お決まりの部分

ファイル名と合わせることに注意

Sample1_03_1.java

```
1 class Sample1_03_1 {  
2     ^   public static void main (String[] args) {  
3     ^   ^   ^  
4     ^   ^   System.out.println("HelloWorld!!");  
5     ^   ^   ^  
6     ^   }  
7 }
```

～ソースコード概要～

コンピュータに
命令する内容

Sample1_

0

2

3

4

1 class Samp 1_03_1 {←

2 ^ public static void main (String[] args) {←

3 ^ ^ ←

4 ^ ^ System.out.println("HelloWorld!!");←

5 ^ ^ ←

6 ^ }←

7 }←

1. Javaプログラミング概要

《全角スペースを含むソースコード（文法エラー）》

```
Sample1_03_2.java
1 class Sample1_03_2 {
2   ^ public static void main (String[] args) {
3   ^   ^
4   ^   System.out.println("Hello World!! Error is cleared!!");
5   ^   ^
6   ^   }
7 }
```

コンパイルエラー

```
C:\Workspace>javac Sample1_03_2.java
Sample1_03_2.java:4: エラー: '¥u3000' は不正な文字です
    System.out.println(¥u3000"Hello World!! Error is cleared!!");
                        ^
Sample1_03_2.java:4: エラー: ';' がありません
    System.out.println(¥u3000"Hello World!! Error is cleared!!");
                        ^
Sample1_03_2.java:4: エラー: 式の開始が不正です
    System.out.println(¥u3000"Hello World!! Error is cleared!!");
                        ^
エラー3個
```

《デバッグ》

- ソースコードは使用するプログラミング言語の文法に従って記述しなければなりません。
全角スペースなどの間違い文法を含んだソースコードをコンパイルすると、コンパイルに失敗してバイトコードファイルも生成されません。
- エラーの原因を究明・除去することで正常に稼働するように対処することを**デバッグ**と言います。
- コンパイルに失敗すると通常はエラーの内容を記述したメッセージ（エラーメッセージ）が表示されます。
デバッグを行う際はこのメッセージを参考にしましょう。
- デバッグの秘訣は以下の2点です。
 - ・ エラーメッセージを参考にエラー箇所と原因を推察する
 - ・ 正常なソースコードと見比べる

<演習：Ex1_03_1>

「Ex1_03_1.java」を改修して、自分の名前をコマンドプロンプト上に表示させてみましょう！

UZUZでコーポレートドッグを
担当しているモコといいます！
よろしくね！

```
C:¥WorkSpace>javac Ex1_03_1. java  
C:¥WorkSpace>java Ex1_03_1  
就活犬 モコ
```



<演習：Ex1_03_2>

「Ex1_03_2.java」のエラーが潜んでいる箇所を探して
デバッグしてみましょう！！

```
C:¥Workspace>javac Ex1_03_2.java
```

```
C:¥Workspace>java Ex1_03_2  
Hello World!! Error is cleared!!
```



うふふ・・・
エラーは1つとは限らないよ・・・