

1. NetBeansの導入と簡単なjavaプログラム

2016/10/3

1 プログラミング言語の選択

1.1 使用する言語の要件

- 対象をできるだけ素直に表現できること
- アルゴリズムをできるだけ素直に表現できること
- コードを再利用することができること
- 適切に文法のチェックが働き、誤り箇所がわかりやすいこと
- 実行時のエラー箇所がわかりやすいこと
- モジュール化が容易であること
- できればGUIを容易に構築できること

1.2 Java

- Object 指向言語
 - 対象とその操作または運動という概念でコードを設計
 - クラスの継承：抽象的モデルから具体的モデルへ展開可能
 - 情報の秘匿：内部変数を隠蔽
- C/C++と似た言語の構造：基本文法、制御構造はほぼ同じ
- OS非依存
- GUIを標準で持つ
- 無償で利用できる

- ポインタが無い

課題1 以下の手順で、自らのPCについて、javaの環境の有無、及びそのバージョンを確認しなさい。

1. コントロールパネルを開く
2. 「プログラム」「プログラムと機能」と開き、インストールされているアプリケーション一覧を開く
3. ”Java SE Development Kit”のバージョンを確認する
4. 現在(2016/8/20)の最新版は、”Java SE Development Kit 8 update 102”

課題2 以下の手順でJavaをインストールまたは、最新版にしなさい

1. 自分が使っているOSが32ビットなのか、64ビットなのかを確認する 「コントロールパネル」「システムセキュリティ」「システム」
2. 32bitの場合に使用するJDK <http://aoba.cc.saga-u.ac.jp/local/jdk-8u102-windows-i586.exe>
3. 64bitの場合に使用するJDK <http://aoba.cc.saga-u.ac.jp/local/jdk-8u102-windows-x64.exe>
4. Windows 以外の場合
 - (a) <http://www.oracle.com/technetwork/java/javase/downloads/>へアクセス
 - (b) Java Platform (JDK) 8u102 を選択
 - (c) Accept License Agreement をチェック
 - (d) OSに対応したものをダウンロード
5. ダウンロードしたファイルを選択してインストール
6. インストール後、バージョンを確認しなさい

2 統合開発環境のインストール

2.1 統合開発環境

- 言語の構造に対応したエディタ
- ヒントの表示 (メソッド候補、ドキュメントなど)
- エラー表示とソースコードとの対応付け
- 利用できる部品の例示

課題 3

1. NetBeans がインストールされているか、そのバージョンがいくつかを調べる。最新は 8.1
2. Windows の場合、<http://aoba.cc.saga-u.ac.jp/local/netbeans-8.1-javase-windows.exe>
3. Windows 以外の場合
 - (a) <https://netbeans.org/downloads/?pagelang=ja> へアクセス
 - (b) 最小セット (左端) を選択して、ダウンロードとインストール

3 Hello World

簡単なプログラムを作成します。NetBeans を起動して、以下の手順で準備します。

- 画面左に「プロジェクト」ウィンドウが出ていることを確認する。出ていない場合には、「ウィンドウ」から「プロジェクト」を選択して、表示する。
- 「ファイル」メニューから「新規プロジェクト」を指定する。
- 「カテゴリ」では「Java」を、「プロジェクト」では「Java アプリケーション」を選択する。
- プロジェクト名をつける。今回は Introduction とする。プロジェクトの場所を選ぶ。デフォルトのままでもよい。「メイン・クラスの作成」のチェックを外す。

課題4 java では、プログラムを package というグループで管理します。今日のサンプルプログラムも一つの package として管理します。

- NetBeans のプロジェクトウィンドウ中で、Introduction プロジェクトのソースファイルフォルダのアイコンでマウス右ボタン 「新規」 「Java パッケージ」で新しいパッケージを作成する。名前は helloWorld とする。最初が小文字であることに注意。
- NetBeans のプロジェクトウィンドウ中で、Introduction プロジェクトのソースファイルフォルダ中のパッケージ helloWorld のアイコンでマウス右ボタン 「Java クラス」を選択し、Program HelloWorld.java を作成する。
- 同様に、Program Main.java を作成する。
- NetBeans のプロジェクトウィンドウ中で、Introduction プロジェクトのソースファイルフォルダのアイコンでマウス右ボタン 「消去してビルド」によって、コンパイル。
- コンパイルできたら、NetBeans のプロジェクトウィンドウ中で、Introduction プロジェクトのソースファイルフォルダ中の Main.java のアイコンで、マウス右ボタン 「実行」で実行する。

HelloWorld.java

```
package helloWorld;
```

```
/**  
 * First java example  
 *  
 * @author tadaiki  
 */  
public class HelloWorld {  
  
    //クラス内のフィールド (変数)  
    private final String message;  
  
    /**  
     * コンストラクタ : クラスのインスタンスを生成する  
     *  
     * @param message 表示するメッセージ  
     */  
    public HelloWorld(String message) {  
        //引数のmessageをインスタンスのmessageに代入  
        //thisは、このコンストラクタで生成されるインスタンス  
        this.message = message;  
    }  
  
    /**  
     * メッセージの印刷  
     */  
    public void printMessage() {  
        System.out.println(message);  
    }  
}
```

Main.java

```
package helloWorld;
```

```
/**
```

```
 * First java example
```

```
 *
```

```
 * @author tadaki
```

```
 */
```

```
public class Main {
```

```
    /**
```

```
     * 実行はここから
```

```
     *
```

```
     * mainは、クラスのインスタンスを操作する場所。
```

```
     *
```

```
     * mainに操作の本体などを書かないこと。
```

```
     *
```

```
     * @param args the command line arguments
```

```
     */
```

```
public static void main(String[] args) {
```

```
    String message = "こんにちは";
```

```
    //HelloWorldクラスのインスタンスの生成
```

```
    HelloWorld hello = new HelloWorld(message);
```

```
    //メッセージ印刷メソッド呼び出し
```

```
    hello.printMessage();
```

```
}
```

```
}
```