**3.4 参考：コメント(comment)とJAVADOC**

コメント(comment)はコードの説明です

### １）1行コメント

|  |  |
| --- | --- |
| 1  2  3  4 | **public** **static** **void** main(String[] args) {  // 1行コメント  String a, b;  } |

### 2)複数行コメント

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **public** **static** **void** main(String[] args) {  String a, b;  /\*  a = "coding";  b = "everybody";  System.out.println(a+b);  \*/  } |

3）　JavaDocコメント

JavaDocは、Javaプログラミング言語で使用されるドキュメンテーションツールの一種です。JavaDocは、Javaのクラス、インタフェース、メソッド、フィールドなどの要素の説明や使い方を記述するためのコメント形式を提供します。

JavaDocコメントは、クラスやメソッドの先頭に「/\*\*」から始まり、「\*/」で終わる形式で書かれます。この形式でコメントを記述すると、JavaDocツールがそれを解析して、HTML形式のドキュメンテーションページを生成することができます。

JavaDocを使用することで、Javaプログラムのメンテナンスや再利用性を高めることができます。また、JavaDocによって生成されたドキュメンテーションページを参照することで、Javaの要素の使い方や機能について理解することができます。

以下は、Javaドキュメンテーションコメントの例です。これは、クラスのJavaDocコメントであり、クラスの概要を説明し、使用方法を示す例です。

/\*\*

\* このクラスは、ユーザーのプロフィールを管理するためのものです。

\* プロフィール情報には、名前、年齢、住所などが含まれます。

\*

\* このクラスのインスタンスを作成すると、新しいプロフィールが作成されます。

\* プロフィールの情報を取得するためには、getterメソッドを使用します。

\* プロフィールの情報を設定するためには、setterメソッドを使用します。

\*/

public class UserProfile {

private String name;

private int age;

private String address;

/\*\*

\* 新しいプロフィールを作成します。

\*

\* @param name プロフィールに設定する名前

\* @param age プロフィールに設定する年齢

\* @param address プロフィールに設定する住所

\*/

public UserProfile(String name, int age, String address) {

this.name = name;

this.age = age;

this.address = address;

}

/\*\*

\* プロフィールの名前を取得します。

\*

\* @return プロフィールの名前

\*/

public String getName() {

return name;

}

/\*\*

\* プロフィールの年齢を取得します。

\*

\* @return プロフィールの年齢

\*/

public int getAge() {

return age;

}

/\*\*

\* プロフィールの住所を取得します。

\*

\* @return プロフィールの住所

\*/

public String getAddress() {

return address;

}

/\*\*

\* プロフィールの名前を設定します。

\*

\* @param name 設定する名前

\*/

public void setName(String name) {

this.name = name;

}

/\*\*

\* プロフィールの年齢を設定します。

\*

\* @param age 設定する年齢

\*/

public void setAge(int age) {

this.age = age;

}

/\*\*

\* プロフィールの住所を設定します。

\*

\* @param address 設定する住所

\*/

public void setAddress(String address) {

this.address = address;

}

}

4)JAVAの文末にはセミコロン（;）

JAVAの文末にはセミコロン（;）をつけます。これは、文（statement）の終わりを示すための記号で、JAVAのコンパイラが正しい文の区切りを認識できるようにするために必要です。例えば、以下のようなJAVAの文では、各行の末尾にセミコロンが必要です。