

ソースコードの基本ルール

読みにくいソースコードは
品質が低いと評価される！

全角文字はNG！
(コメント内とデータは除く)

コメント（注釈）をつけて
誰が見ても読みやすい
ソースコードを！

《ソースコードの基本ルール①》

- ソースコードは**基本的に半角英数文字**を用いて記述します。
全角文字はデータとしてやコメント（後述）内では扱うことが可能ですが、命令文の中では使用できません。
- ソースコードの品質を表す重要な要素の1つとして「**可読性**」があります。
改行やインデント（後述）を用いて、誰が見ても分かりやすいソースコードを心掛けましょう。
- ソースコードを読む人間に向けた注釈のことを**コメント**と言います。コメントをつけることで可読性を飛躍的に上げることができます。
コメントはコンパイルや処理実行の際に無視されるため、全角文字を含めて何を記載しても問題ありません。

① 行コメント「//」

「//」より右側をコメントとみなします。

② 範囲コメント「/* */」

「/*」から始まり「*/」で終わる領域をコメントとみなします。

```
Sample1_04_1.java
1  /*-< Sample1_04_1 >-----*/
2  定価1000円の商品の税込価格を計算して表示するプログラムです。
3  消費税は8%で計算しています。
4  -----*/
5  class Sample1_04_1 {
6      ^   public static void main (String[] args) {
7      ^   ^
8      ^   ^   int    teika    = 1000 ;           //定価の金額
9      ^   ^   ^
10     ^   ^   double zeiritu = 1.08 ;           //消費税
11     ^   ^   ^
12     ^   ^   double siharai = teika * zeiritu ; //支払い額を計算
13     ^   ^   ^
14     ^   ^   System.out.println(siharai) ;    //支払い額を画面に表示
15     ^   ^   ^
16     ^   ^   }
17 }
```

Sample1_04_1.java

```
1  /*-< Sample1_04_1 >-----<
2  定価1000円の商品の税込価格を計算して表示するプログラムです。<
3  消費税は8%で計算しています。<
4  -----*/<
5  class Sample1_04_1 {<
6  ^ public static void main (String[] args) {<
7  ^     double teika<
8  ^     ^<
9  ^     ^<
10 ^     double zeiritu = 1.08 ;<
11 ^     ^<
12 ^     double siharai = teika * zeiritu ;<
13 ^     ^<
14 ^     System.out.println(siharai) ;<
15 ^     ^<
16 ^ }<
17 }
```

範囲コメント

行コメント

//定価の金額<

//消費税<

//支払い額を計算<

//支払い額を画面に表示<

ソースコードは
ブロックによって構造が
定義されている！

インデントを活用することで
ブロックのネスト構造が
分かりやすくなる！

《ソースコードの基本ルール②》

- 中カッコ { } で囲われた領域を**ブロック**と言います。
{ } の中に { } を記述することも可能で、このことを**ネスト**と言います。
- 全てのソースコードはお決まりとして「class」という語句から始まるブロックに囲われています。
このブロックのことを**クラスブロック**と言います。
また、クラスブロック直下でネストされているブロックのことを**メソッドブロック**と言います。
基本的にコンピュータに実行させたい**命令文はメソッドブロック内に記述**します。
- ネストの構造が複雑になってくると、どこからどこまでが { } の範囲か分かりづらくなってきます。この対策として { } で囲まれているブロック内の各行を字下げすることが一般的で、この字下げのことを**インデント**と言います。
インデントは**タブ1つ**か**半角スペース4つ**を用いることが一般的です。
なお、タブや半角スペースはコンパイルや処理実行において無視されます。

```
Sample1_04_1.java
1  /*--< Sample1_04_1 >-----*/
2  定価1000円の商品の税込価格を計算して表示するプログラムです。
3  消費税は8%で計算しています。
4  -----*/
5  class Sample1_04_1 {
6      ^   public static void main (String[] args) {
7      ^   ^   <
8      ^   ^   int    teika    = 1000 ;           //定価の金額
9      ^   ^   <
10     ^   ^   double zeiritu = 1.08 ;           //消費税
11     ^   ^   <
12     ^   ^   double siharai = teika * zeiritu ; //支払い額を計算
13     ^   ^   <
14     ^   ^   System.out.println(siharai) ;     //支払い額を画面に表示
15     ^   ^   <
16     ^   }
17 }
```

クラスブロック

メソッドブロック

命令群

```
1 Sample1_04_1 >
2 画面1000円の商品の税金を計算して表示するプログラムです。
3 消費税は8%で計算しています。
4 -----*/
5 class Sample1_04_1 {
6 ^ public static void main (String[] args) {
7 ^   ^
8 ^   ^ int    teika    = 1000 ;           //定価の金額
9 ^   ^   ^
10 ^  ^ double zeiritu = 1.08 ;           //消費税
11 ^  ^   ^
12 ^  ^ double siharai = teika * zeiritu ; //支払い額を計算
13 ^  ^   ^
14 ^  ^ System.out.println(siharai) ;    //支払い額を画面に表示
15 ^  ^   ^
16 ^  ^ }
17 }
```

インデント

```
1 >-----<
2 定価1000円の商品の税込価格を計算して表示するプログラムです。<
3 消費税は8%で計算しています。<
4 -----*/<
5 class Sample1_04_1 {<
6     public static void main (String[] args) {<
7         <
8         int    teika    = 1000 ;           //定価の金額<
9         <
10        double zeiritu = 1.08 ;           //消費税<
11        <
12        double siharai = teika * zeiritu ; //支払い額を計算<
13        <
14        System.out.println(siharai) ;     //支払い額を画面に表示<
15        <
16    }<
17 }
```


メソッドブロック内に
命令文を記述する！

適切なポイントで
改行をつける！

命令文の終わりには必ず
『;』をつける！

命令群は上から順番に
処理されていく！

《ソースコードの基本ルール③》

- 前述のとおりJavaで実行させたい命令文は基本的にメソッドブロック内に記述します。

命令文の最後尾には必ず『;（セミicolon）』を記述する必要があります。（日本語の「。」、英語の「.」と同じ）構造を表す{ } など、命令文でない部分で『;』は不要であることにご注意ください。

- 読みやすいソースコードを書くために、適切に**改行**を使用するように心がけましょう。
基本的には **1 命令文につき 1 回改行**するようにしましょう。
改行自体に意味はなく、タブや半角スペースと同じくコンパイルや処理実行において無視されます。

<ポイント>

『{』 『}』 『;』 が出てきたら改行しましょう！

- メソッドブロック内に定義された命令群は上から順に処理されていきます。

```
Sample1_04_1.java
1  /*-< Sample1_04_1 >-----*/
2  定価1000円の商品の税込価格を計算して表示するプログラムです。
3  消費税は8%で計算しています。
4  -----*/
5  class Sample1_04_1 {
6      ^   public static void main (String[] args) {
7      ^   ^   ^
8      ^   ^   int    teika    = 1000 ;           //定価の金額
9      ^   ^   ^
10     ^   ^   double zeiritu = 1.08 ;           //消費税
11     ^   ^   ^
12     ^   ^   double siharai = teika * zeiritu ; //支払い額を計算
13     ^   ^   ^
14     ^   ^   System.out.println(siharai) ;     //支払い額を画面に表示
15     ^   ^   ^
16     ^   ^   }
17 }
```

Sample1_04_1.java

上から順に
処理される

命令文の区切りとして
『;』をつける
(同時に改行もつける)

```
1 1 >-----<
2 定価の税込価格を計算して表示するプログラムです。<
3 消費税を計算しています。<
4 -----<
5 class Sample1_04_1 {<
6 ^ public static void main (String[] args) {<
7 ^ ^<
8 ^ ^ int teika = 1000;<
9 ^ ^<
10 ^ ^ double zeiritu = 1.08;<
11 ^ ^<
12 ^ ^ double siharai = teika * zeiritu;<
13 ^ ^<
14 ^ ^ System.out.println(siharai);<
15 ^ ^<
16 ^ }<
17 }
```

//定価の金額<

//消費税<

//支払い額を計算<

//支払い額を画面に表示<

< 演習：Ex1_04_1 >

- (1) 「Ex1_04_1.java」 に適切な改行を入れていきましょう
- (2) 「Ex1_04_1.java」 に適切なインデントを入れていきましょう
- (3) 「Ex1_04_1.java」 の冒頭に以下のコメントを入れましょう

九九を表示するプログラムです。
数値を1つ入力して実行することで入力した数の段のみを表示することができます。
数値を入力せずに実行した場合は全ての段が表示されます。

内容はまだ
わからなくて
いいよ！

《ポイント》

- ・ 『{』 『}』 『;』 で改行を入れましょう！
ただし、以下の『;』 では改行を入れないでください。
『for(int j = 1 ; j <= 9 ; j++)』 『for(int j = 1 ; j <= 9 ; j++)』
- ・ インデントは**タブ**でつけましょう！（半角スペースを使用するとよくズレます）

★サクラエディタをインストールしましょう！

追加タスクの選択
実行する追加タスクを選択してください。

サクラエディタ インストール時に実行する追加タスクを選択して、「次へ」をクリックしてください。

- ☒ スタートメニューを作成(S)
- ☒ Quick Launchにアイコン作成(Q)
- ☒ プログラム一覧に追加(P)
- ☒ デスクトップにアイコン作成(D)
- ☒ 「SAKURAで開く」メニューの追加(E)
- ☐ 起動時に常駐(B)
- ☒ 送るに追加(T)

< 戻る(B) 次へ(N) > キャンセル

チェックして右クリックメニューに
「SAKURAで開く」を追加しましょう！