

3章問題

3-2解答

解答.E

解説

Booleanはtrueもしくはfalseが返るように設定する決まりがある。(※ p71の表3-4参照)

その為、A.B.C.Dは「文字列や数値」
となっているので不正解となる。

3-4解答

- 解答.B

- 6行目の式

$\text{If}((i < j) \parallel (k-- > j)) \{ \dots \}$

①論理演算子である「 \parallel 」は最初の条件で当てはまればもう片方の式は見ないでtrueとなる

② $(i=5 < j=10)$ i が j より小さい ○ \rightarrow この時点で $(k-- > j)$ は見ない。

- 9行目の式

$\text{If}((i < j) \&\& (--k < j)) \{ \dots \}$

3-4解答

③論理演算子である「&&」は「かつ」という意味となる為、どちらの式についても見て両方を満たせばtrueとなる。

④ (i=5 < j=10) iがjより小さい ○ かつ

(--k=14 < j=10) kがjより小さい ×

両方の条件を満たしていない為、9行目に関してはfalseとなり表示されない

3-6解答

- 解答.E

- 4行目

`counts[1] = (counts[2] == 2) ? counts[3] : 99`

※count[1]に右の式の結果を代入

①`(counts[2] == 2) → (3 == 2)? 4 : 99`

count[2]の値が、2であるならcount[3]の値をcount[1]に代入
そうでないなら99を代入

②count[2]は3であるため、「99」がcount[1]に代入される

3-10解答

解答.C

- ・switch文については、p95を参照。
- ①case文ではbreakの記載がないと、当てはまる条件から下の結果を出力してしまう。
- ②よって、val=100が代入されている為、 14行目以降が全て出力対象となります。

3-12解答

- 解答.B

- 5行目

`if(str1 == str2)`

- ① `str1`と`str2`は同じ「Java」という文字だが、「==」演算子が比
べているのは、同じ参照先であるかどうかなので、`false`となる
。

3-12解答

- 6行目

`if(str1.equals(str2))`

- ①equalsは値そのものを比較する為、str1とstr2は同じ文字列(java)となるのでtrueとなる。

3-13解答

解答.C

- ・4行目でstr1をstr2に代入している

①if(str1 == str2)の条件については、4行目でstr2はstr1を参照するように設定されている為trueとなる

②if(str1.equals(str2))の条件についても、値が同じためtrueとなる

3-18解答

解答 .A

①concat()→対象の文字を追加

(追加した後に、変数に詰めてあげる必要有)

②append()→対象の文字を追加

(変数の値に直接追加になる為、appendの後は特に何もする必要はなし)

※性能面を考えるとappendを使用した方が良い。

concatを使用する際は、一度変数に詰める処理が必要な為表示する際にappendよりも処理が多くなる。

3-18解答

- concatを使用する利点はあるのか？

調べてみると、通常はあまり使用しない方が良いみたいです。
単文字だけ連結の場合に使用するみたいです。