

Java Bronze

第3章 条件文（分岐文）

if文・if-else文 / switch文

- 制御文とは
- if文、if-else文
 1. if文
 2. if-else文
 3. else if文
 4. 条件演算子
 5. if文のネスト
- switch文

- 制御文…プログラムの流れをコントロールする文

Java言語には条件文(分岐文)、繰り返し文、繰り返し制御文の3種類の制御文がある

ある条件ごとに処理する内容を分けるための文

①if文・if-else文

②switch文

①if文 ・ if-else文

1. if文

5/16

if文…条件式をもち、その条件を評価した結果に応じた処理を行う

構文

```
if (条件式) {  
    処理文 ;    // ifブロック : 条件式の結果がtrueであれば実行  
}
```

1. if文

6/16

“boolean値”

① 条件の判定
(trueまたはfalse)

② ①の結果が
trueのとき

③ ①の結果が
falseのとき

ex)

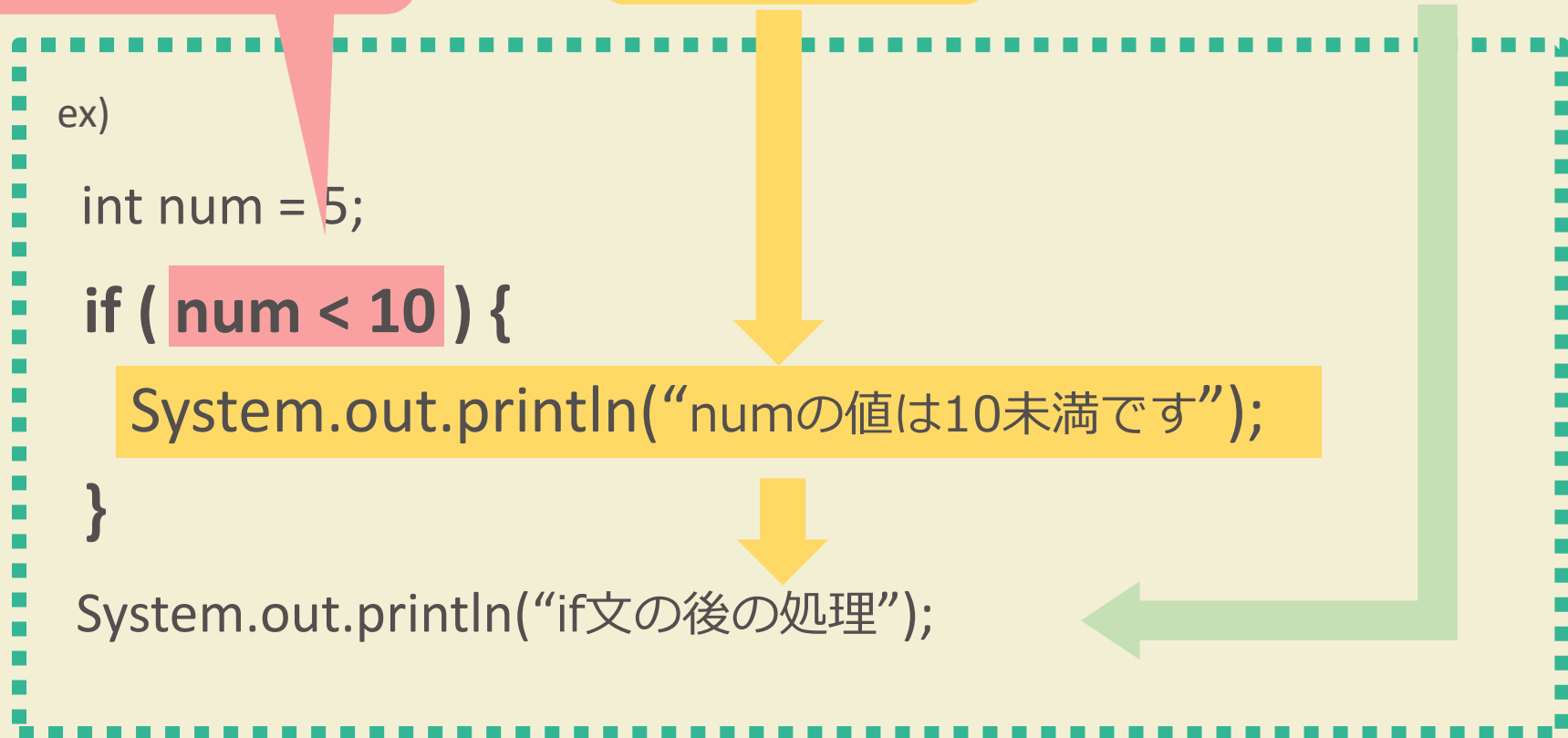
```
int num = 5;
```

```
if ( num < 10 ) {
```

```
    System.out.println(“numの値は10未満です”);
```

```
}
```

```
System.out.println(“if文の後の処理”);
```



2. if-else文

7/16

if-else文…条件式の結果がtrueだったときだけでなく、falseだったときの処理も記述

構文

```
if (条件式) {  
    処理文1;    // 条件式の結果がtrueのときに実行 (ifブロック)  
} else {  
    処理文2;    // 条件式の結果がfalseのときに実行 (elseブロック)  
}
```

2. if-else文

8/16

“boolean値”

① 条件の判定
(trueまたはfalse)

② ①の結果が
trueのとき

③ ①の結果が
falseのとき

ex)

```
int num = 20;
```

```
if ( num < 10 ) {
```

```
    System.out.println(“numの値は10未満です”);
```

```
} else {
```

```
    System.out.println(“numの値は10以上です”);
```

```
}
```


3. else if文

9/16

else if文… 3 分岐以上の多分岐処理を行う

構文

```
if (条件式1) {  
    処理文1;    // 条件式1の結果がtrueのときに実行 (ifブロック)  
} else if (条件式2) {  
    処理文2;    // 条件式2の結果がtrueのときに実行 (else ifブロック)  
} else {  
    処理文3;    // 条件式1および2の結果がfalseのときに実行 (else ifブロック)  
}
```

3. else if文

10/16

① 1つ目の条件判定
(trueまたはfalse)

② ①の結果が
trueのとき

③ ①の結果が
falseのとき
2つ目の条件判定
(trueまたはfalse)

④ ③の結果が
trueのとき

⑤ ③の結果も
falseのとき

ex)

```
char c = 'c';
```

```
if ( c == 'a' ) {
```

```
    System.out.println("cの値はaです");
```

```
} else if ( c == 'b' ) {
```

```
    System.out.println("cの値はbです");
```

```
} else {
```

```
    System.out.println("cの値はaでもbでもありません");
```

```
}
```

4. 条件演算子

11/16

条件演算子…条件式の結果がtrueなら式1を返し、
falseなら式2を返す演算子

構文

条件式 ? 式1 : 式2

ex)

```
String str = "numの値は";
```

```
str += num < 10 ? "10未満" : "10以上";
```

条件式

条件式の結果がfalseなら
str += "10以上";

条件式の結果がtrueなら
str += "10未満";

4. 条件演算子

12/16

- 条件演算子はif文より短くて簡単に書ける
- 条件演算子は、条件によって代入する値を変えるという処理でよく使う
- 条件演算子は値を返すためのもの
条件によって行う処理を分けたいときは、普通にif文を使う

5. ifのネスト

13/16

- if文は、ブロック内にさらに別のif文を記述することができる

これをネスト（入れ子）という

ネストを使用すると条件を重ねることができる

②switch文



1. switch文

15/16

switch文…多分岐処理

式の評価結果値のデータ型は
byte, char, short, int, enum, String

構文

```
switch (式) { // switchブロック
    case 定数1 : // 式の結果が定数1と一致したとき、以下の処理文を実行
        処理文1;
    case 定数2 : // 式の結果が定数2と一致したとき、以下の処理文を実行
        処理文2;
    .....
    default : // どのcaseにも一致しなかったとき、以下の処理文を実行
        処理文n;
}
```

1. switch文

16/16

ex)

```
switch ( i ) {
```

```
  case 1 :
```

① iが1のとき

```
    System.out.println("1");
```

```
    break;
```

break文があるので
switchから抜ける

```
  case 2 :
```

② iが2のとき

```
    System.out.println("2");
```

```
  default :
```

```
    System.out.println("default");
```

```
}
```

break文がないので
続けてdefaultの処理
文を実行