

Java講座

第3回－後半

情報科学部コンピュータ科学科
2年 根岸 拓郎

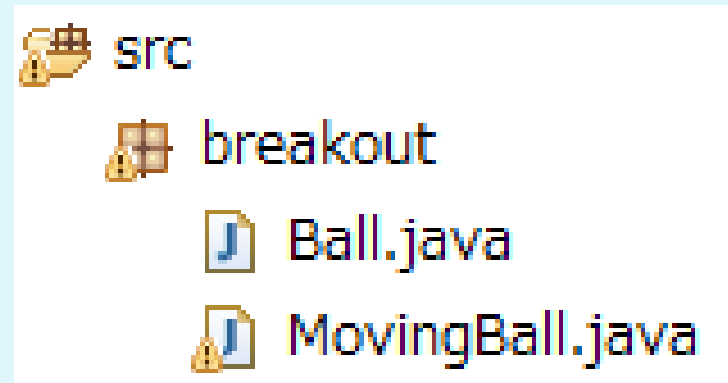
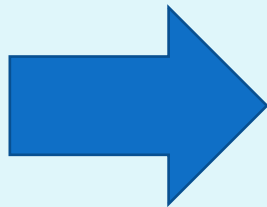
前回のブロック崩しの続き

- ◆ 前回のプログラムから徐々にブロック崩しに近づけていきましょう！

まず始める前に

- ◆ まずブロック崩し用のパッケージ「breakout」パッケージを作りましょう。
- ◆ そして「breakout」パッケージ内に前回作成した「Ball」クラスと「MovingBall」クラスを移しましょう。

◆ こんな感じで



スレッドについて

スレッドについて

- ◆ 前回とりあえずスレッドを作成しました。

```
thread = new Thread(this); // スレッドを作成
thread.start();           // スレッドをスタート⇒これでrun()メソッドが開始
```

```
public void run() {
    while(true){ // 無限ループ
        try{
            Thread.sleep(10); // 0.01秒停止
            repaint(); // 再びpaint()メソッドを起動します
        }catch (InterruptedException e) {
            // Thread.sleep()を使う際に、このtry-catchが必要
        }
    }
}
```

- ◆ 「implements Runnable」をすることで、MovingBallクラスはrun()メソッドをオーバーライドしなければいけません。
- ◆ このrun()メソッドとThreadを使って並列処理を行い、連続で描画を行います。

スレッドの意味

- ◆ 通常プログラムは1つの流れで動きます。

```
1 ↓ void start(){  
    int a = 10;  
    show();  
    a += 10;  
    JOptionPane.showMessageDialog(null, “加算後は” + a + “です。”);  
4 ↓ }  
2 ↓ void show(){  
    JOptionPane.showMessageDialog(null, “aは” + a + “です。”);  
}
```

- ◆ スレッドはこのプログラムの流れを複数増やせます。

①

```
thread = new Thread(this); // スレッドを作成  
thread.start();           // スレッドをスタート⇒これでrun()メソッドが開始
```

①の流れの中で、`thread.start()`をすると...

②の流れも出来て`run()`メソッドが実行される

②

```
public void run() {  
    while(true){ // 無限ループ  
        try{  
            Thread.sleep(10); // 0.01秒停止  
            repaint(); // 再びpaint()メソッドを起動します  
        }catch (InterruptedException e) {  
            // Thread.sleep()を使う際に、このtry-catchが必要  
        }  
    }  
}
```


スレッドの停止

- ◆ `Thread.sleep(<時間>);` で、<時間>ミリ秒停止できます。

※1000ミリ秒=1秒なので、`Thread.sleep(10);`だと0.01秒停止

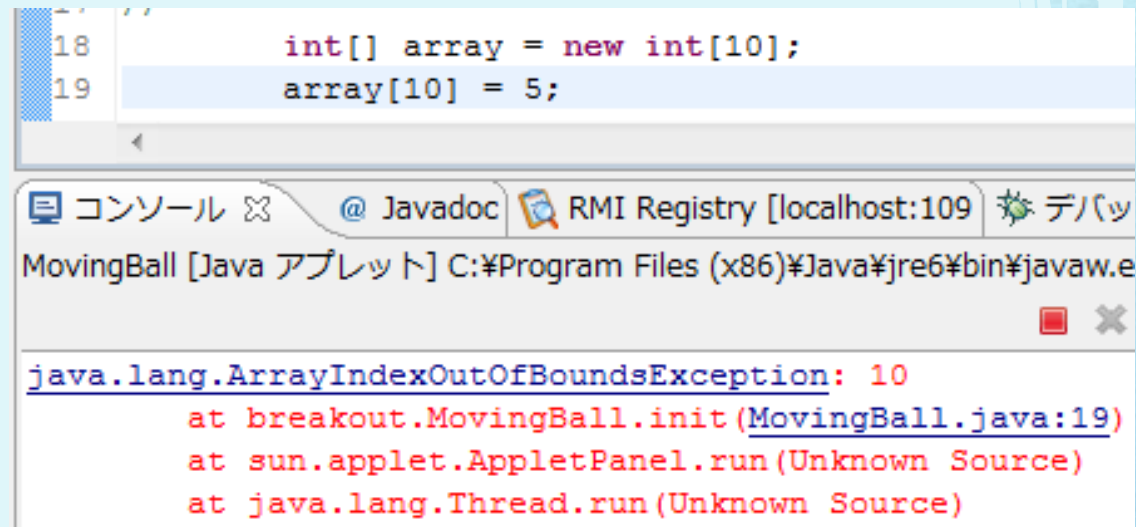
- ◆ ボールがカクカク動くなら、この<時間>の値を変えてみよう

try-catch

ちょっとむずかしいので「こんなのあるんだ」程度で

◆ try-catchは「例外処理」と言って、予測されるプログラム中のエラーを捕まえて、処理を施せます

◆ 左のプログラムはエラーが出ています
※10個分の配列を作ったらarrayは[0]~[9]しかないよね。



```
18     int[] array = new int[10];
19     array[10] = 5;
```

コンソール @ Javadoc RMI Registry [localhost:109] デバッ

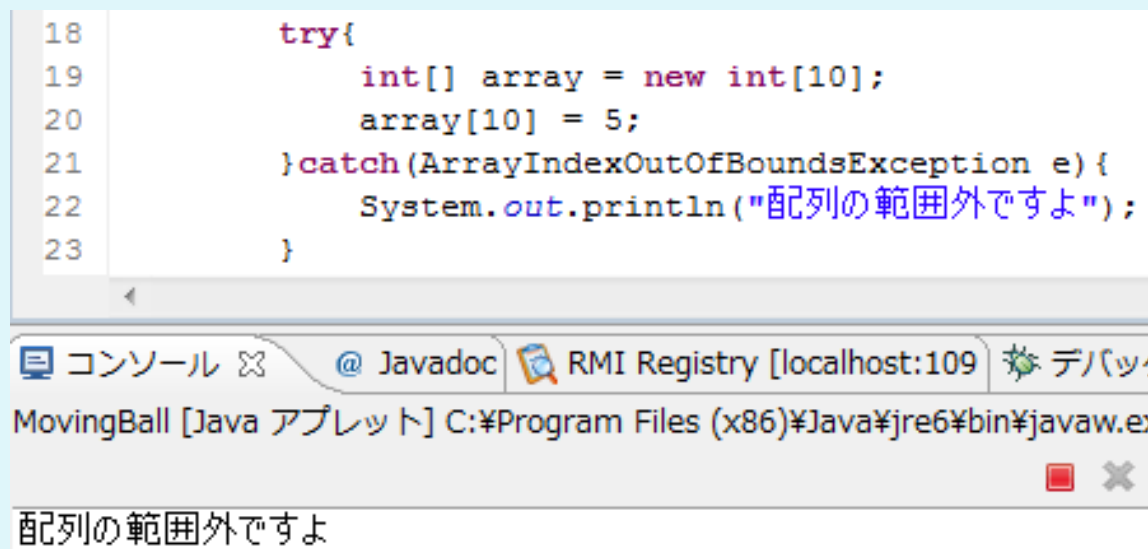
MovingBall [Java アプレット] C:\Program Files (x86)\Java\jre6\bin\javaw.e

```
java.lang.ArrayIndexOutOfBoundsException: 10
    at breakout.MovingBall.init(MovingBall.java:19)
    at sun.applet.AppletPanel.run(Unknown Source)
    at java.lang.Thread.run(Unknown Source)
```

- ◆ 先ほどのプログラムをtry-catchで囲ってみます
- ◆ 予測されるのは「ArrayIndexOutOfBoundsException」
- ◆ 配列の範囲外へのアクセスです

- ◆ これでエラーが
でなくなりました

```
18     try{
19         int[] array = new int[10];
20         array[10] = 5;
21     }catch(ArrayIndexOutOfBoundsException e){
22         System.out.println("配列の範囲外ですよ");
23     }
```



The screenshot shows a Java IDE window with a code editor and a console. The code editor contains a try-catch block that attempts to access an array element at index 10, which is out of bounds. The catch block prints the message "配列の範囲外ですよ" (Out of array bounds). The console window below shows the output of the program, which is "配列の範囲外ですよ".

- ◆ `run()`メソッド内の`try-catch`は
`Thread.sleep()`に対する例外処理で、
「`InterruptedException`」は「割り込み」の例外
です

```
while(true){ // 無限ループ
    try{
        Thread.sleep(10); // 0.01秒停止
        repaint(); // 再びpaint()メソッドを起動します
    }catch (InterruptedException e) {
        // Thread.sleep()を使う際に、このtry-catchが必要
    }
}
```

キー入力

- ◆ インタフェースを使って
キー入力をさせてみよう！
- ◆ さらに左右の矢印キーで
動くバーを作ってみよう！

キー入力

- ◆ MovingBallクラスの「implements Runnable」の部分を「implements Runnable, KeyListener」としてみよう
- ◆ するとMovingBallクラスに赤線が出てエラーが…
- ◆ エラー箇所にもうすを合わせて「実装されていないメソッドを追加」をクリック！

↓ ここクリックでも出ます

```
7 public class MovingBall exten
```

keyPressedメソッド

- ◆ 左下のメソッドが追加されるはずですが。
- ◆ それぞれの機能については前半のスライド参照

- ◆ keyPressedメソッドは
Enter、スペース、矢印キー
が押された時に起動します

```
@Override
public void keyPressed(KeyEvent e) {
    // TODO 自動生成されたメソッド・スタブ
}

@Override
public void keyReleased(KeyEvent e) {
    // TODO 自動生成されたメソッド・スタブ
}

@Override
public void keyTyped(KeyEvent e) {
    // TODO 自動生成されたメソッド・スタブ
}
```

- ◆ keyPressedメソッドの中身を次の通りに書き換えてみよう
- ◆ これで左右のキーが押された時の処理をそれぞれに書いてあげればいい

```
@Override
public void keyPressed(KeyEvent e) {
    switch (e.getKeyCode()) {
        case KeyEvent.VK_LEFT:
            // ココに左キーが押されたときの処理
            break;
        case KeyEvent.VK_RIGHT:
            // ココに右キーが押されたときの処理
            break;
    }
}
```


Barクラスを作ろう

- ◆ Ballクラスと同じように、左右に動くバーを表すBarクラスを作ってみよう
- ◆ ほとんどBallクラスの応用で作れるはず・・・

問題

- ◆ 先ほどのkeyPressedメソッド、Barクラスなどの中身を書き換えて、バーを動かしてみよう
- ◆ もちろん新しくメソッドを作ったり変数を作ったりしてOKです
- ◆ 出来た人はボールがバーにぶつかったらボールが跳ね返るようなプログラムに挑戦してみよう

終わり