**Swing スレッド①**

*をかすプログラム①（シングルスレッド）*

スレッドをうと、くプログラムをくことができます。

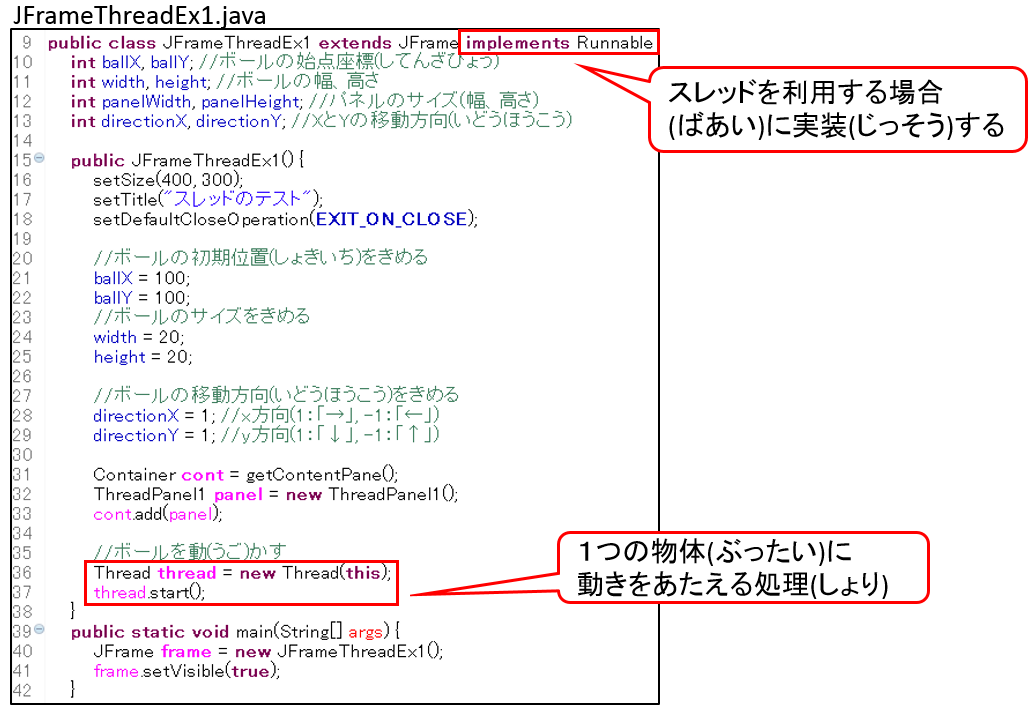
まずはいがいていく、アニメーションをってみましょう。



まずはのプログラムをしてしてみましょう。

フレームをします。

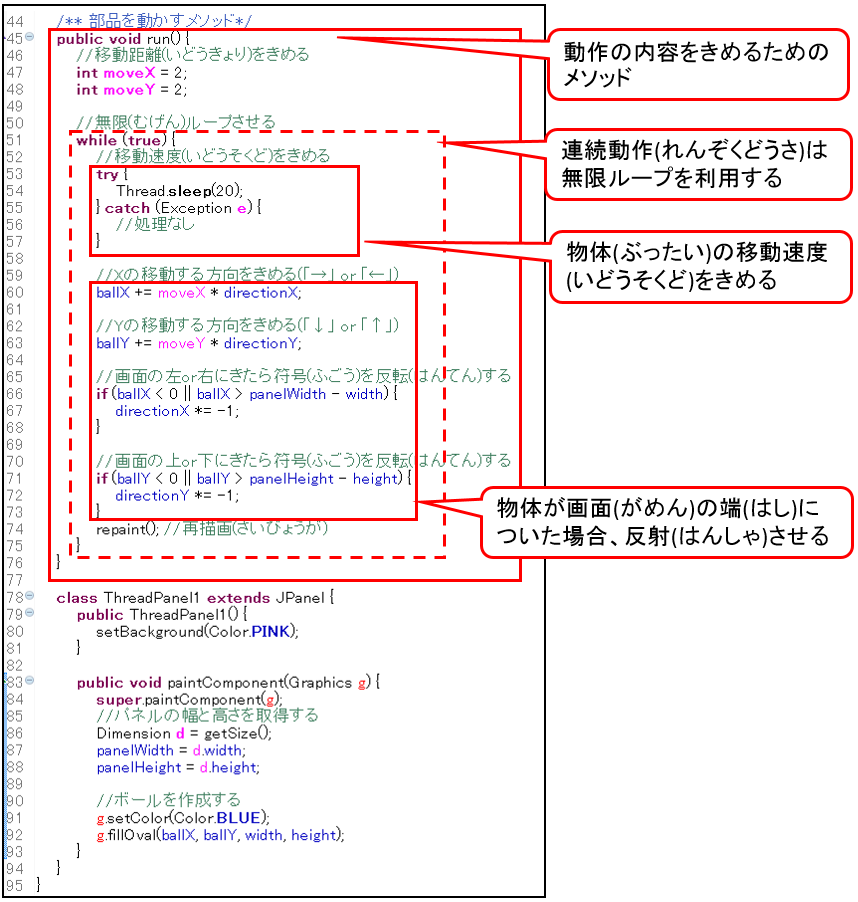
2



5

1

フレームのrunメソッドとパネルクラスをします。



5

6

2

4

3

2

プログラムイメージはにしています。

***Point 1*** *をかすためのスレッドクラス*

Thread スレッドの変数 = new Thread(this);

をかすにはスレッドクラスをします。

スレッドの変数.start();

startメソッドによってをします。



！

***Point 2*** *のをきめるrunメソッド*

public void run() {

//動作を記載する

}

runメソッドはPoint1でをかしたのをするメソッドです。

・・・ implements Runnable {

またrunメソッドをするはあわせて｢Runnable｣インターフェースをします。

***Point 3****アニメーション()のループ*

アニメーションとはのをですることをしています。

while (true) {

//動作用プログラム

}

runメソッドのにループするプログラムをし、ブロックにのをする

プログラムをします。

***Point 4*** *の(をする)をきめる*

try {

Thread.sleep(速度[msec]));

} catch (Exception e) {

//処理なし

}

のメソッドをすることでのをすることができます。

はmsec (1000msec = 1sec)

このメソッドをするは(try～catch)もするがあります。

***Point 5*** *をかすのプログラムのれ*

スレッドをする、のれ（①～④）でプログラムがします。

このれをしながらプログラムをしていきましょう。

Thread thread = new Thread(this);

thread.start();

=================================================

public void run() {

while (true) {

try {

Thread.sleep(20);

} catch (Exception e) {

//処理なし

}

//動作プログラム

repaint();

}

}

1. Threadのstartメソッドの
2. ループごとに

③ ④をループ

1. runメソッドのプログラムをみむ
2. ループでの

***Point 6*** *のにしたにさせる*

のにがした、にがくようにしています。

の

の

の

の

if (Y座標 < 0 || Y座標 > パネルの - の) {

Y \*= -1;

}

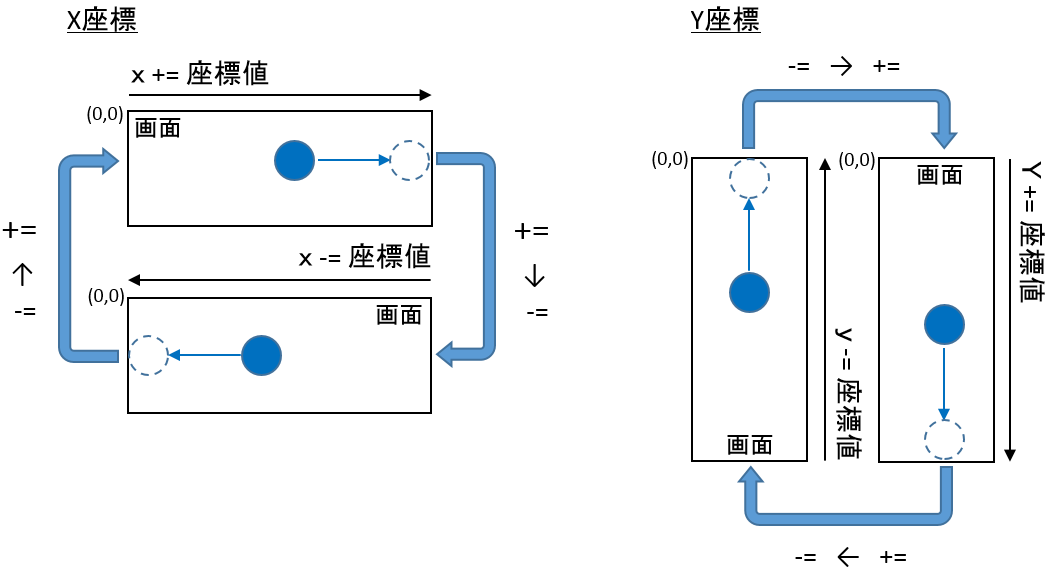
if (X < 0 || X > パネルの - の) {

X \*= -1;

}

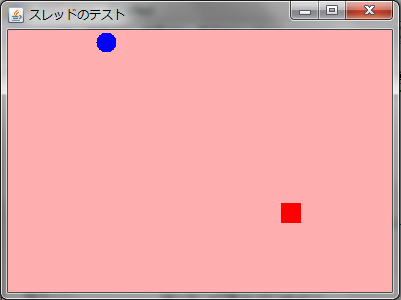
のにした、-1をけすることで｢+ ⇔ -｣をりえます。

このようにすることでのでがしているようながれます。



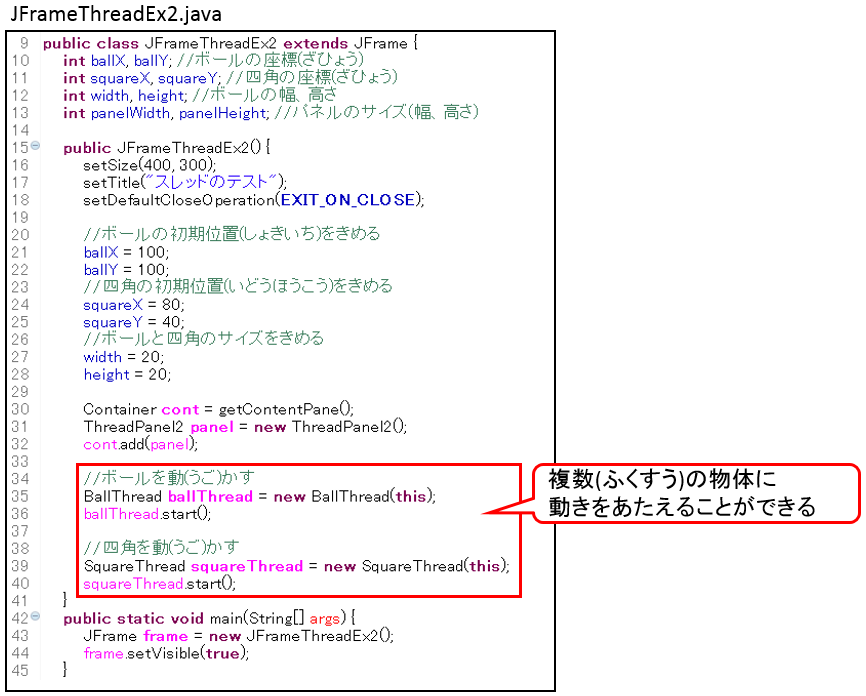
*をかすプログラム②（マルチスレッド）*

スレッドクラスをしたクラスでのをする、マルチスレッドにしてみましょう。



しくのプログラムをしてしてみましょう。

フレームをします

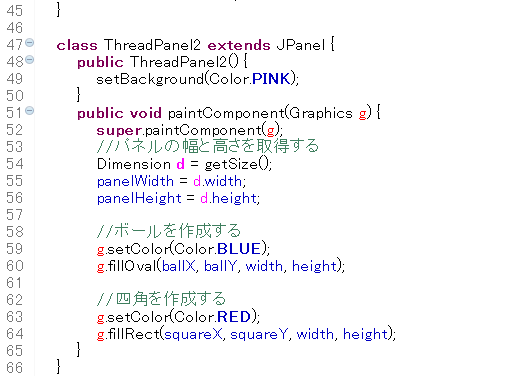


7

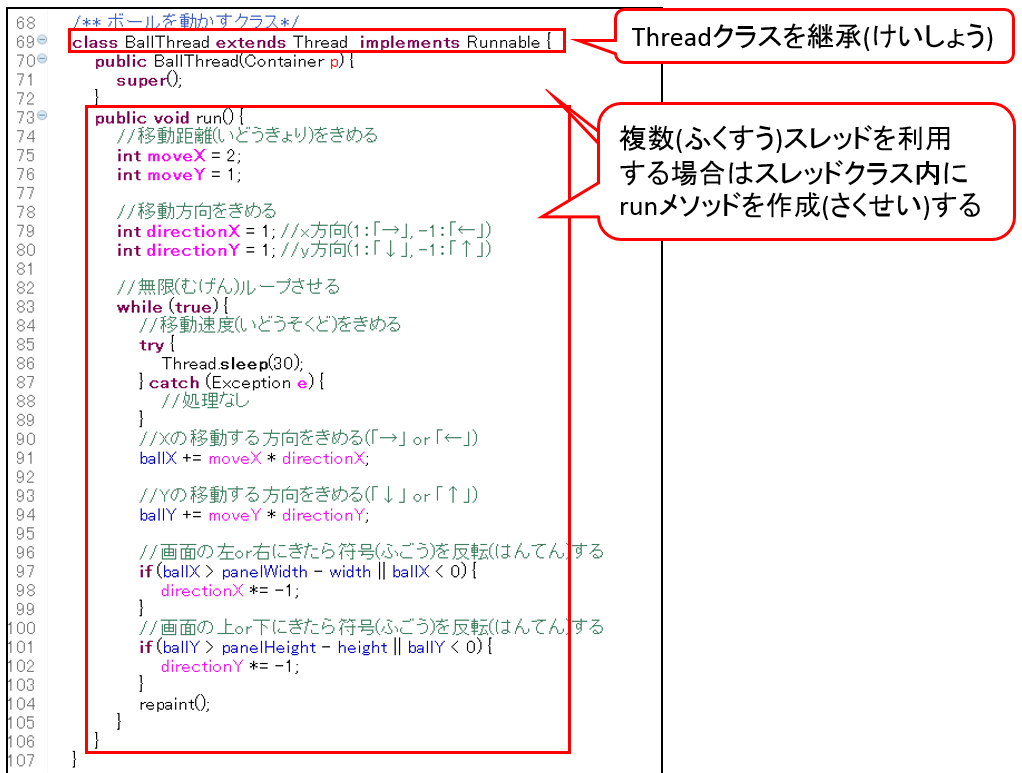
プログラムの



パネルクラスをします。

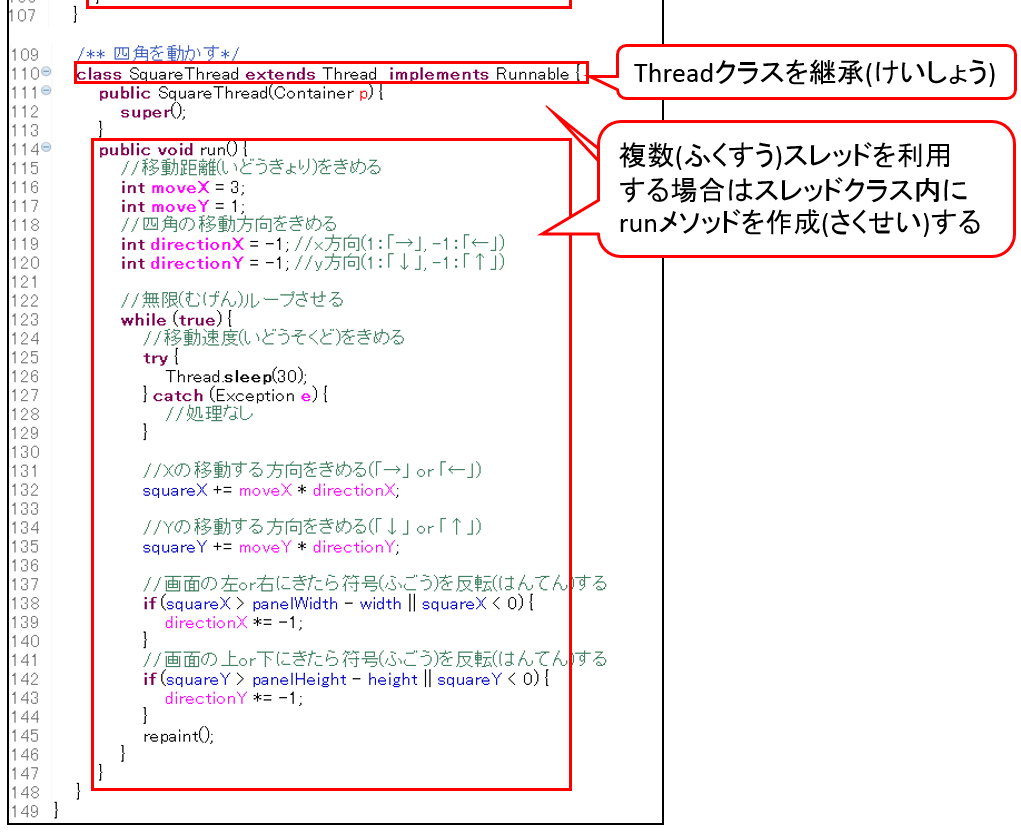


スレッドクラスをします



8

8



8

8

***Point 7*** *のをにかす*

スレッドクラスをすることでのを1つのでにかすことができます。

Thread スレッド１の変数 = new Thread(this);

スレッド１の変数.start();

Thread スレッド２の変数 = new Thread(this);

スレッド２の変数.start();

・

・

・

Threadクラスのをすることで、させるのを２つにやせます

***Point 8*** *のスレッドにをする*

のスレッドクラスをする、をするrunメソッドはのThreadクラスにします。

class スレッドクラス名 extends Thread implements Runnable {

//省略(しょうりゃく)

public void run() {

//動作用プログラム

}

}

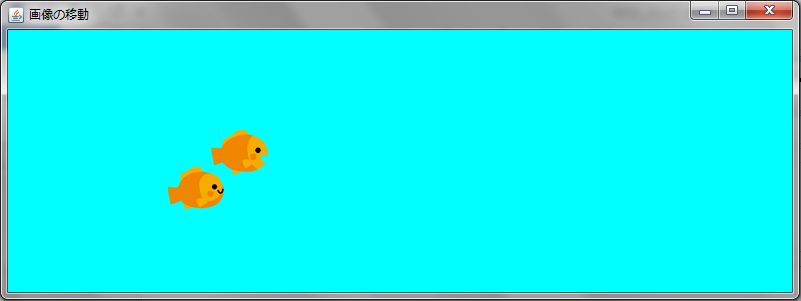
このはスレッドクラスにして｢Runnable｣インターフェースをするがあります。

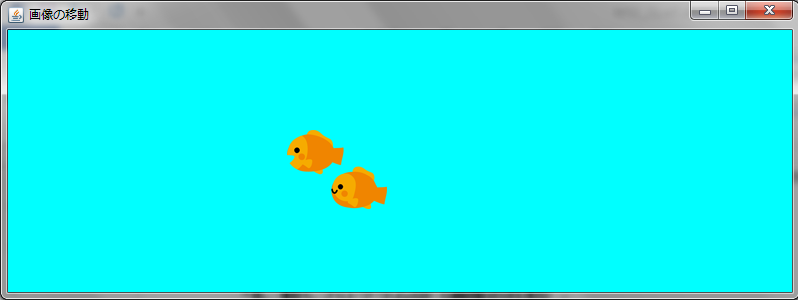
*をかすプログラム③（の）*

をし、スレッドをってかしてみましょう。

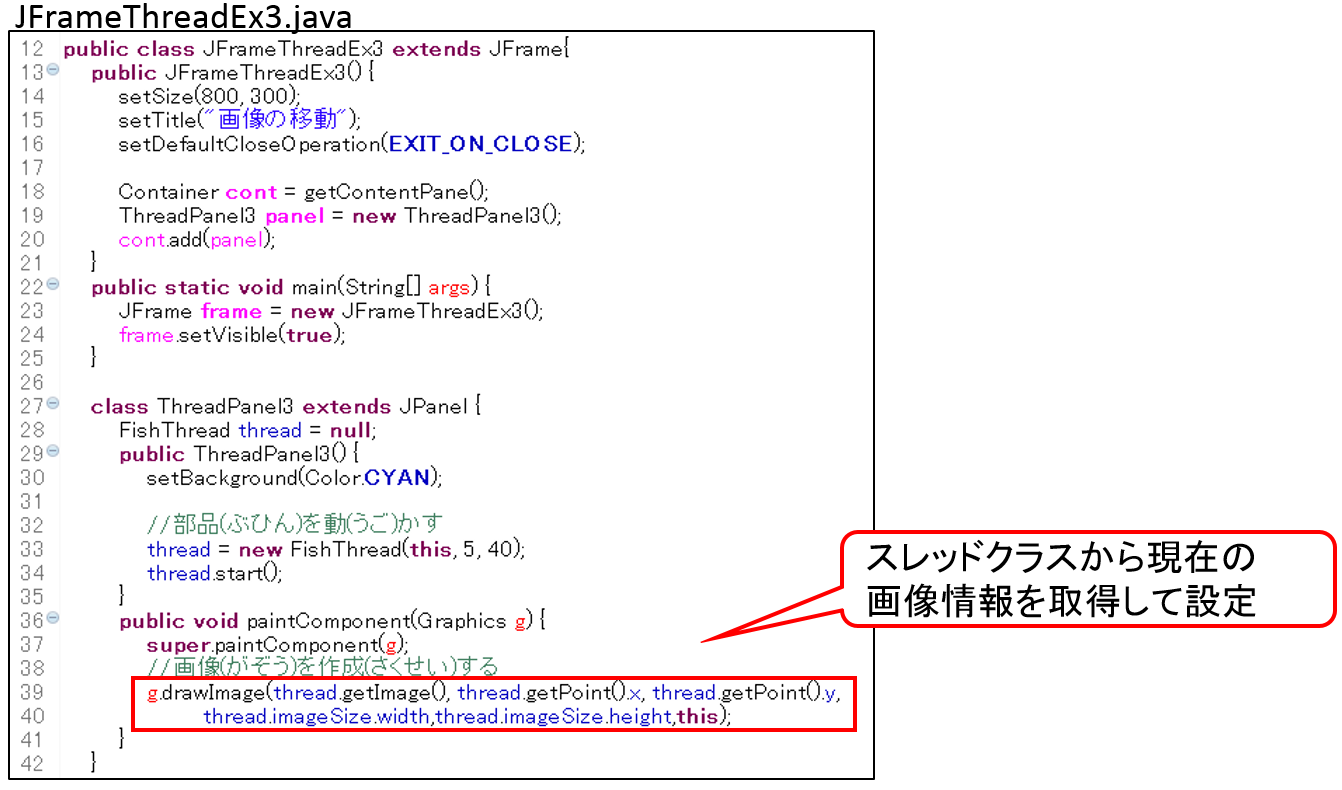
がのをいでいるようなプログラムをります。

ののがるようにもしましょう。

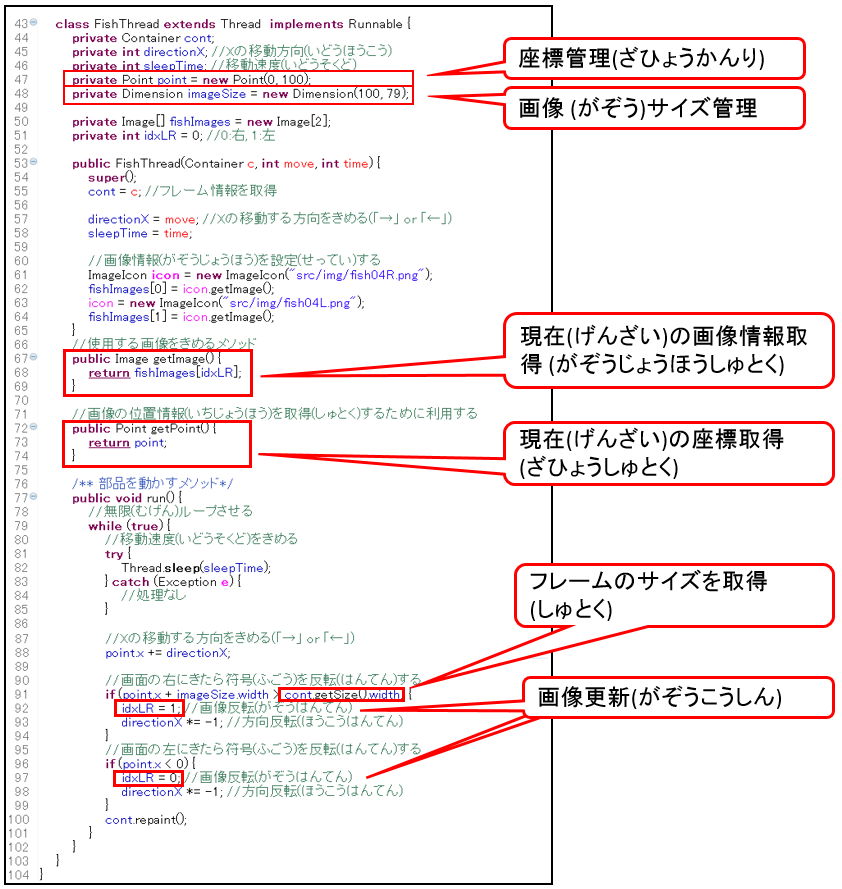




しくのプログラムをしてしてみましょう。



スレッドクラスをします



12

11

11

11

9

10

9

プログラムイメージはにしています。

5

6

5

***Point 9*** *のけし*

Point ポイントクラスの変数 = new Point(X軸の値, Y軸の値);

PointクラスをしてX,Yのを、することができます。

public Point getPoint() {

return ポイントクラスの変数;

}

ゲッターメソッド(アクセッサ)をすることでのクラスでもできるようにします。

***Point 10*** *のけし*

Dimension 画像サイズの変数 = new Dimension(100, 79)

Dimensionクラスをしての、さのを、することができます。

のとさはのでができます。

・int の　＝ サイズの.width

・int のさ　＝ サイズの.height

***Point 11*** *のりえ*

public Image getImage() {

return イメージクラスの配列の変数[index];

}

ゲッターメソッド(アクセッサ)をすることでどのをするかをし、

のクラスでもできるようにします。

このプログラムではindexのをrunメソッドでりえています。

***Point 12*** フレーム（コンテナ）の

のクラスのコンストラクタにとして｢Container｣のをすことで、フレームの

クラスでもフレーム(コンテナ)のをすることができます。

｢Container｣のをし、フレームのとさをのでできます。

・int フレームの　＝ コンテナの.getSize().width

・int フレームのさ　＝ コンテナの.getSize().height

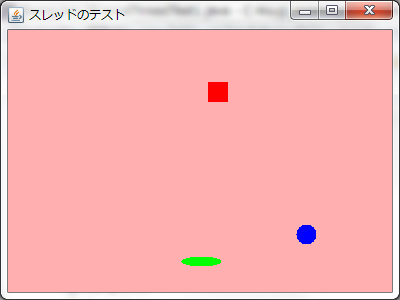
**Q1**

「をかすプログラム②(JFrameThreadEx2)」をにしてのようにくプログラムをしてください。

Javaクラスは｢JFrameThreadTest1｣としてプログラムをしてください。

スレッドをひとつし、３つめのをかしてみましょう。

のイメージではをしていますが、、、のはとします。



**Q2**

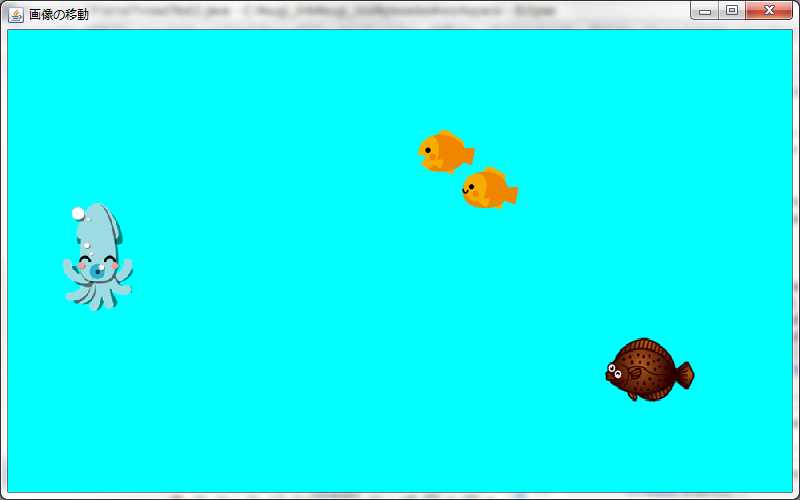
「をかすプログラム③(JFrameThreadEx3)」をにしてのようにくプログラムをしてください。

Javaクラスは｢JFrameThreadTest2｣としてプログラムをしてください。

さをえてにく（①）とにく（②）をしましょう。

またがわるたびにがりわるようにしましょう。

するの、の、サイズはとします。

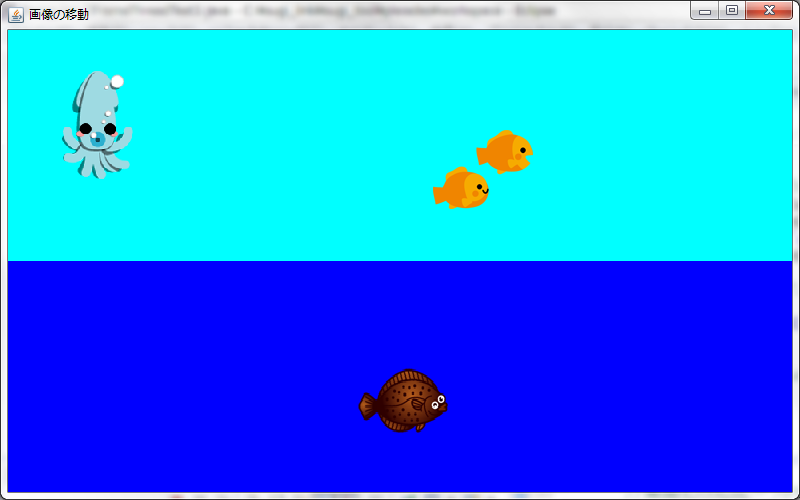
　　 

**Q3**

「Q2」をにしてのようにくプログラムをしてください。

Javaクラスは｢JFrameThreadTest3｣としてプログラムをしてください。

にはQ2とじですが、のパネルにしてをかしてみましょう。



パネル②

パネル①