

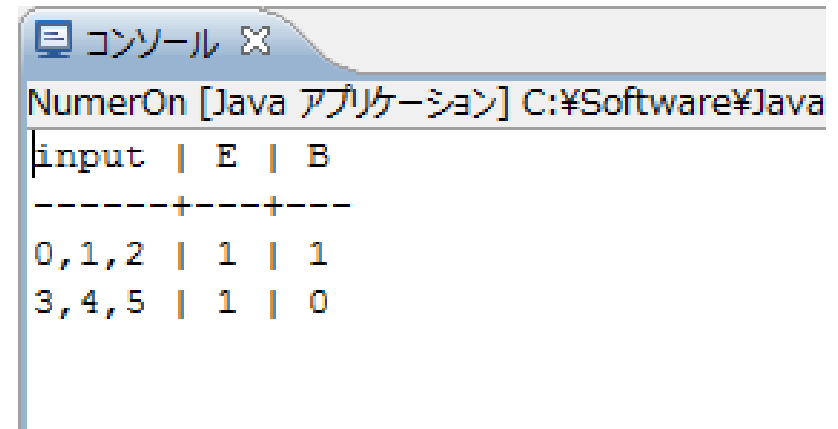
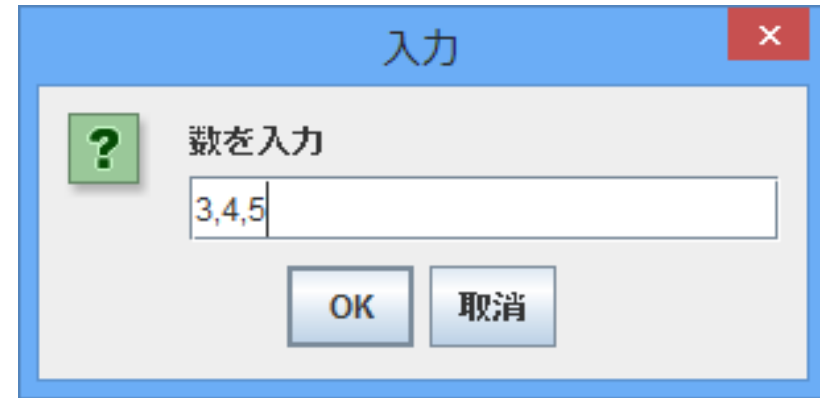
java講座第4回

担当 中山

演習1 NumerOn

NumerOnを作成せよ

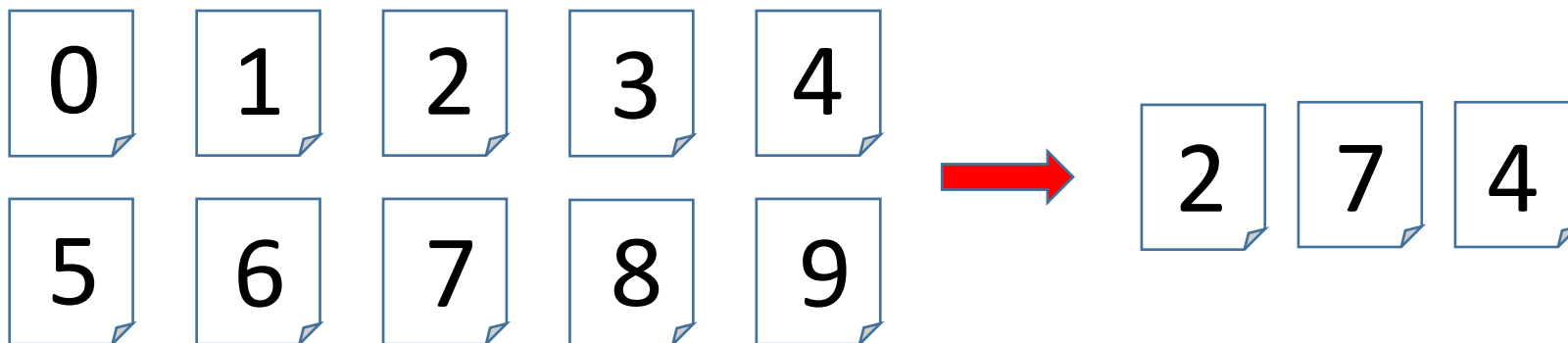
入力ダイアログからカンマ区切りで
数字を3つ入力し、それに対する結
果をコンソールに表示せよ



NumerOnとは？

数当てゲームの一種

0~9の10個の数字を重複しないようにして、3桁の数をつくる
その数を当てるゲームのこと

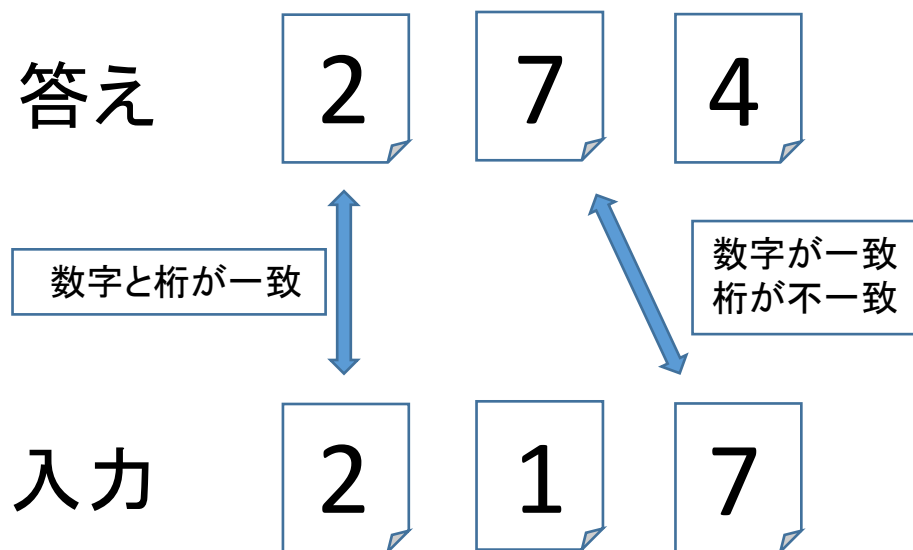


EATとBITE

EAT 数字とその桁が一致している

BITE 3桁の数にその数字が含まれているが、桁が不一致

例)

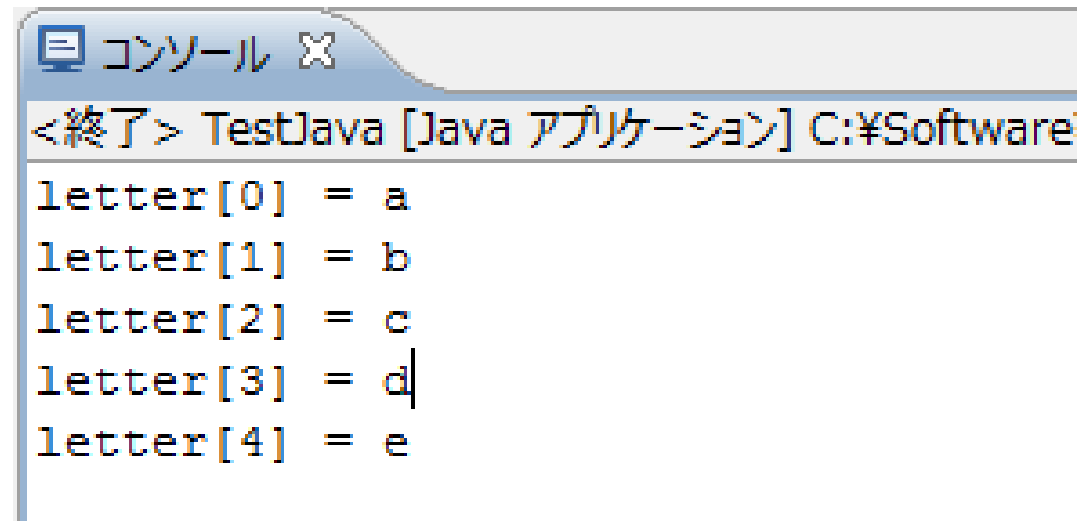


1EAT - 1BITE

文字列の分割

指定した文字で文字列を分割する

```
String input = JOptionPane.showInputDialog("カンマ区切りで入力");  
String[] letter = input.split(",");  
for(int i=0; i<letter.length; i++)  
    System.out.println("letter["+i+"] = "+letter[i]);
```



重複しない 乱数の生成

listに0~9を格納する

listの順序を無作為に入れ替える

listをansの配列に3つ格納する

```
Random.java
package k3;

import java.util.ArrayList;
import java.util.Collections;

public class Random {
    public static void main(String[] args) {
        new Random().start();
    }

    void start() {
        ArrayList<Integer> list = new ArrayList<Integer>();
        for (int i = 0; i < 9; i++)
            list.add(i);
        Collections.shuffle(list);

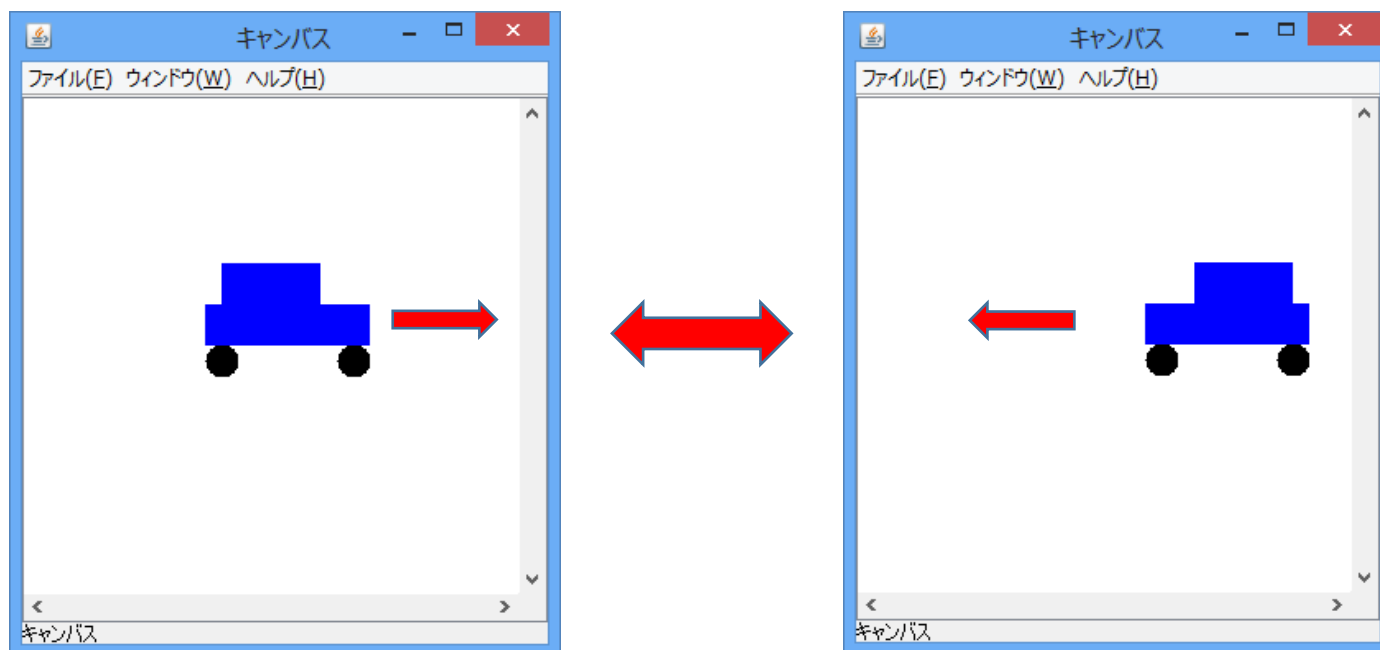
        int[] ans = new int[3];
        for (int i = 0; i < ans.length; i++) {
            ans[i] = list.get(i);
            System.out.print(ans[i]);
        }
    }
}
```

コンソール

<終了> TestJava [Java アプリケーション] C:\Software\Java\jre-1.7.0_17\bin\javaw.exe (2014/06/27)

演習2 MoveCar

キャンバスに車を描写し、横に移動するプログラムを作成せよ。
また、車がキャンバスの端に来たら反転し、移動するようにせよ。



キャンバスの機能

- `Canvas.show();` キャンバスを表示する
- `Canvas.setColor(red, green, blue);` 色を設定する
- `Canvas.fillOval(x, y, width, height);` 色付きの楕円を描く
- `Canvas.fillRect(x, y, width, height);` 色付きの長方形を描く
- `Canvas.clear();` キャンバスの内容を消去する
- `Canvas.waitForCountdown(milliseconds);` ミリ秒待機する

```
11+ void start() {  
22  
23     // キャンバスに車を描写する  
24+ void draw(Car car) {  
36  
37     // 車を移動させる  
38+ void move(Car car) {  
42  
43     // 車の方向を変える  
44+ void direct(Car car) {  
48  
49+ Car createCar(int x, int y, int w, int h, int t, int d, int vx, int vy) {  
61 }  
62  
63 class Car{  
64     int x, y;           // 車の位置  
65     int width, height, tire; // 車の大きさ  
66     int dir;            // 車の向き  
67     int vx, vy;         // 車の移動量  
68 }
```

演習1 NumerOn の答え

```
12 void start() {
13     int size = 3;
14     int[] ans = getAns(size);
15     System.out.println("input | E | B ");
16     System.out.println("-----+---+---");
17
18     while (true) {
19         String input = JOptionPane.showInputDialog("数を入力");
20         String[] num = input.split(",");
21         System.out.println(input + " | "
22                             + EAT(size, num, ans) + " | " + BITE(size, num, ans));
23         if (EAT(size, num, ans) == size)
24             break;
25     }
26     JOptionPane.showMessageDialog(null, "正解");
27 }
28
29 int[] getAns(int size) {
30     ArrayList<Integer> list = new ArrayList<Integer>();
31     for (int i = 0; i < 9; i++)
32         list.add(i);
33     Collections.shuffle(list);
34
35     int[] ans = new int[size];
36     for (int i = 0; i < size; i++)
37         ans[i] = list.get(i);
38     return ans;
39 }
```

```
44⊖ int EAT(int size, String[] num, int[] ans) {  
45     int eat = 0;  
46     for (int i = 0; i < size; i++)  
47         if (ans[i] == Integer.parseInt(num[i]))  
48             eat++;  
49     return eat;  
50 }  
51  
52⊖ int BITE(int size, String[] num, int[] ans) {  
53     int bite = 0;  
54     for (int i = 0; i < size; i++)  
55         for (int j = 0; j < size; j++)  
56             if (ans[i] == Integer.parseInt(num[j]) && i != j)  
57                 bite++;  
58     return bite;  
59 }  
60  
61 }
```

演習2 MoveCar の答え

```
11- void start() {
12     Car car = createCar(100,200,100,50,20,1,5,0);
13     Canvas.show();
14     while(true) {
15         Canvas.clear();
16         draw(car);
17         Canvas.waitForCountdown(100);
18         direct(car);
19         move(car);
20     }
21 }
22
23- void draw(Car car) {
24     Canvas.setColor(0, 0, 255);
25     if(car.dir > 0)
26         Canvas.fillRect(car.x+car.width*0.1, car.y, car.width*0.6, car.height/2);
27     else
28         Canvas.fillRect(car.x+car.width*0.3, car.y, car.width*0.6, car.height/2);
29     Canvas.fillRect(car.x, car.y+car.height*0.5, car.width, car.height/2);
30     Canvas.setColor(0, 0, 0);
31     Canvas.fillOval(car.x, car.y+car.height-1, car.tire, car.tire);
32     Canvas.fillOval(car.x+car.width-car.tire, car.y+car.height-1, car.tire, car.tire);
33
34 }
35
36- void move(Car car) {
37     car.x += car.vx * car.dir;
38     car.y += car.vy * car.dir;
39 }
```

```
41 void direct(Car car){
42     if(car.x < 0) car.dir = 1;
43     else if (car.x > 500 - car.width) car.dir = -1;
44 }
45
46 Car createCar(int x, int y, int w, int h, int t, int d, int vx, int vy){
47     Car car = new Car();
48     car.x = x;
49     car.y = y;
50     car.width = w;
51     car.height = h;
52     car.tire = t;
53     car.dir = d;
54     car.vx = vx;
55     car.vy = vy;
56     return car;
57 }
58 }
59
60 class Car{
61     int x, y;           // 車の位置
62     int width, height, tire; // 車の大きさ
63     int dir;           // 車の向き
64     int vx, vy;        // 車の移動量
65 }
```

発展 例外処理

NumerOnをプレイ中，間違えて4つの数字を入力したり，aという文字を入力してしまうことがありますよね？

当然エラーとなり，プログラムは終了...

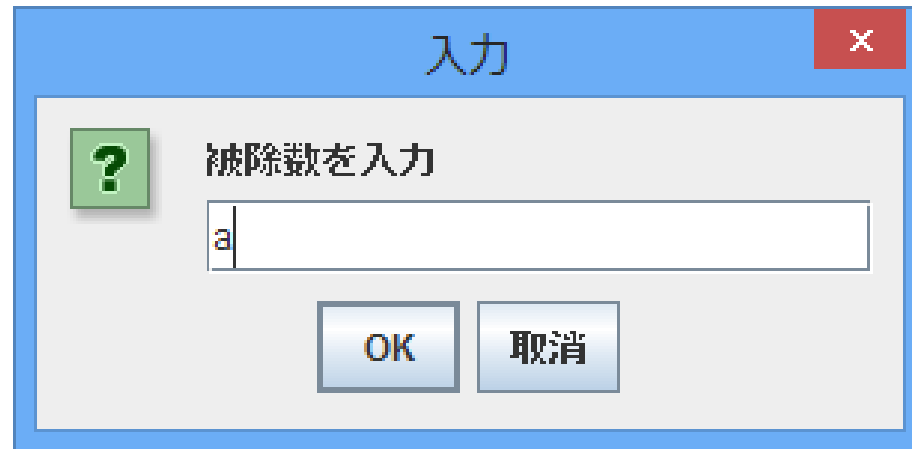
1EAT-2BITEのときに間違えると非常に腹が立つ、(´Д｀)ノ

そこで，ユーザーが間違えて入力してもいいように例外処理をしてみよう

例として、次のプログラムを実行してみよう

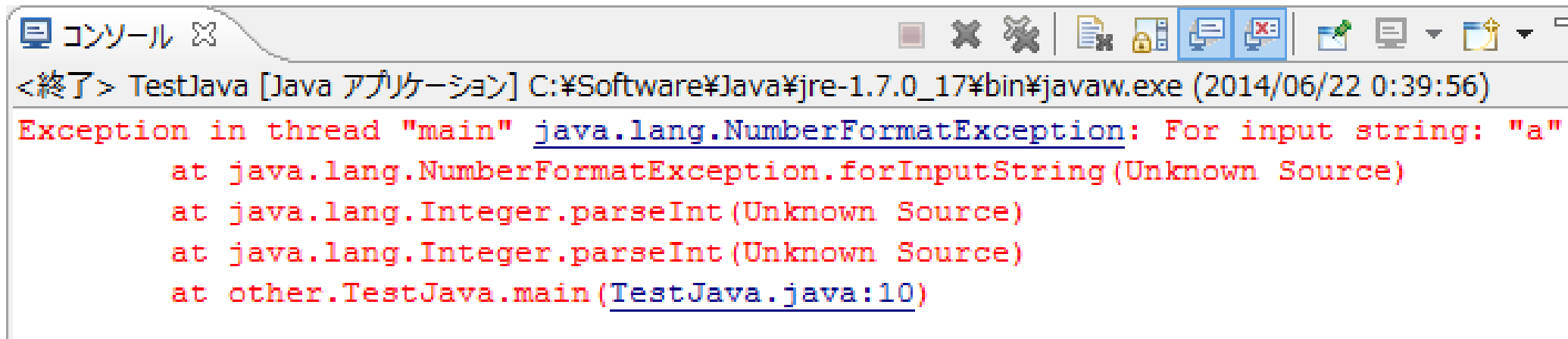
```
8 public class TestJava {  
9     public static void main(String[] args) {  
10         int num1 = Integer.parseInt(  
11             JOptionPane.showInputDialog("被除数を入力"));  
12         int num2 = Integer.parseInt(  
13             JOptionPane.showInputDialog("除数を入力"));  
14         System.out.println(num1 / num2);  
15     }  
16 }
```

ここで



と入力すると,

下のようなエラーメッセージが表示されて、プログラムが中断します
これは、aという文字列が数値型に変換できないのでエラーがおこります



```
コンソール ✕
<終了> TestJava [Java アプリケーション] C:\Software\Java\jre-1.7.0_17\bin\javaw.exe (2014/06/22 0:39:56)
Exception in thread "main" java.lang.NumberFormatException: For input string: "a"
    at java.lang.NumberFormatException.forInputString(Unknown Source)
    at java.lang.Integer.parseInt(Unknown Source)
    at java.lang.Integer.parseInt(Unknown Source)
    at other.TestJava.main(TestJava.java:10)
```

例外を処理するには？

try-catch文を使います

では、数字でない文字を入力してしまった時のために、例のプログラムを try-catch文を使って書き換えてみましょう

```
8 public class TestJava {
9     public static void main(String[] args) {
10         try{
11             int num1 = Integer.parseInt(
12                 JOptionPane.showInputDialog("被除数を入力"));
13             int num2 = Integer.parseInt(
14                 JOptionPane.showInputDialog("除数を入力"));
15             System.out.println(num1 / num2);
16         }catch(NumberFormatException e){
17             System.out.println("数値型変換エラー: "+e);
18             main(args);
19         }
20     }
21 }
```



数値型に変換できないとき、
Catch文の命令が処理される

これで、例のプログラムは数値型に変換できない文字についての例外処理ができました

しかし、これではまだ不完全です

それは、除数に0を入力してしまったときに対しての例外処理をしていません

では、算術の例外処理をしましょう

```
8 public class TestJava {
9     public static void main(String[] args) {
10         try{
11             int num1 = Integer.parseInt(
12                 JOptionPane.showInputDialog("被除数を入力"));
13             int num2 = Integer.parseInt(
14                 JOptionPane.showInputDialog("除数を入力"));
15             System.out.println(num1 / num2);
16         }catch (NumberFormatException e){
17             System.out.println("変換エラー : "+e);
18             main(args);
19         }catch ( ) {
20             System.out.println("算術エラー : "+e);
21             main(args);
22         }
23     }
24 }
```

この□の中には、どのような命令をかけばいいでしょうか
ヒント:ゼロの割り算をすると、どのようなエラーメッセージがでるでしょうか

try-catch文の注意点

try-catch は、必ずふたつの文を書いて正常に動作します
片方だけでは、正しく動作しません

また、ひとつのtry文には、複数のcatch文が書けます

余力のある人に

NumerOnや、今まで書いたプログラムの例外処理を試してみましょう