平成23年度基盤システム演習A第5回レポート

学籍番号：0312010142
講座名　：澤本研
氏名　　：藤田　拓

目 次

◆課題

1　例外処理

```
// ArgDiv2.java

public class ArgDiv2 {
    public static void main( String [] args ) {

        try {
            int val1 = Integer.parseInt( args[0] );
        }
        catch( ArrayIndexOutOfBoundsException param1 ) {
            System.out.println( "引数無し" );
            return;
        }
        catch( NumberFormatException param2 ) {
            System.out.println( "引数が数字では無い" );
            return;
        }
        catch( Exception param3 ) {
            System.out.println( "その他の例外" );
            param3.printStackTrace();
        }
        finally {
            System.out.println( "finally 内の処理 1" );
        }

        try {
            int val2 = Integer.parseInt( args[1] );
        }
        catch( ArrayIndexOutOfBoundsException param4 ) {
            System.out.println( "引数無し" );
            return;
        }
        catch( NumberFormatException param5 ) {
            System.out.println( "引数が数字では無い" );
            return;
        }
        catch( Exception param7 ) {
            System.out.println( "その他の例外" );
            param7.printStackTrace();
        }
        finally {
            System.out.println( "finally 内の処理 2" );
```

```
        }

        int val1 = Integer.parseInt( args[0] );
        int val2 = Integer.parseInt( args[1] );

        try {
            System.out.println( val1 + " / " + val2 + " = " + val1/val2 );
        }
        catch( ArithmeticException param8 ){
            System.out.println( "ゼロ除算" );
            return;
        }


    }
}
```

2 Thread

```
//MyThread.java

public class MyThread extends Thread {
    char mes;
    long sleeptime;

    public MyThread( char mes, long sleeptime ) {
        this.mes = mes;
        this.sleeptime = sleeptime;
    }

    public void run() {
        for( int i = 0; (char)((int)mes + i ) != '9'+1 && (char)((int)mes + i ) != 'Z'+1
                 && (char)((int)mes + i ) != 'ん'+1; i++ ) {
            try {
                System.out.println( (char)((int)mes + i ) );
            }
            catch( ArrayIndexOutOfBoundsException ia ) {
                break;
            }

            try {
                Thread.sleep( sleeptime );
            }
            catch( InterruptedException is ) {
                is.printStackTrace();
            }
```

```
        }
    }
}

//ThreadTest.java

public class ThreadTest {
    public static void main( String [] args ) {

        Thread x = new MyThread(   'A', 1000 );
        Thread y = new MyThread(   '0', 1500 );
        Thread z = new MyThread(  'あ',  500 );
        x.start();
        y.start();
        z.start();
        System.out.println( "Main thread is over." );
    }
}
```

3 Runnable

```
//MyRunnable.java

public class MyRunnable implements Runnable {
    char mes;
    long sleeptime;

    public MyRunnable( char mes, long sleeptime ) {
        this.mes = mes;
        this.sleeptime = sleeptime;
    }

    public void run() {
        for( int i = 0; (char)((int)mes + i ) != '9'+1 && (char)((int)mes + i ) != 'Z'+1
                 && (char)((int)mes + i ) != 'ん'+1; i++ ) {
            try {
                System.out.println( (char)((int)mes + i ) );
            }
            catch( ArrayIndexOutOfBoundsException ia ) {
                break;
            }

            try {
                Thread.sleep( sleeptime );
            }
            catch( InterruptedException is ) {
```

```
                is.printStackTrace();
            }
        }
    }
}
```

```
//RunnableTest.java

public class RunnableTest {
    public static void main( String [] args ) {
        Runnable cmd1 = new MyRunnable(  'A', 1000 );
        Runnable cmd2 = new MyRunnable(  '0', 3000 );
        Runnable cmd3 = new MyRunnable(  'あ',  500 );

        Thread x = new Thread( cmd1 );
        Thread y = new Thread( cmd2 );
        Thread z = new Thread( cmd3 );

        x.start();
        y.start();
        z.start();
        System.out.println( "Main thread is over." );
    }
}
```

4 同期処理

```
//SyncTest.java

public class SyncTest {
    public static void main( String [] args ) {
        Increment inc = new Increment();
        Thread t1 = new Thread( new ExecIncrement( inc, 'A',  1000 ) );
        Thread t2 = new Thread( new ExecIncrement( inc, '0',  3000 ) );
        Thread t3 = new Thread( new ExecIncrement( inc, 'あ',  500 ) );

        t1.start();
        t2.start();
        t3.start();
    }
}
```

```
//ExecIncrement.java

public class ExecIncrement implements Runnable {
    Increment inc;
    char mes;
```

```
    long sleeptime;

    ExecIncrement( Increment inc, char mes, long sleeptime ) {
        this.inc = inc;
        this.mes = mes;
        this.sleeptime = sleeptime;
    }

    public void run() {
        inc.calc( mes, sleeptime );
    }
}

//Increment.java

public class Increment {
    synchronized void calc( char mes, long sleeptime ) {
        for( int i = 0;  (char)((int)mes + i ) != '9'+1 && (char)((int)mes + i ) != 'Z'+1
                 && (char)((int)mes + i ) != 'ん'+1; i++ ) {
            try {
                System.out.println( (char)((int)mes + i ) );
            }
            catch( ArrayIndexOutOfBoundsException ia ) {
                break;
            }

            try {
                Thread.sleep( sleeptime );
            }
            catch( InterruptedException is ) {
                is.printStackTrace();
            }
        }
    }
}
```