

Exception handling

1. Xử lý ngoại lệ

1.1. Sửa đoạn chương trình sau để bắt ngoại lệ chia cho 0

```
public class Main {  
  
    /**  
     * @param args the command line arguments  
     */  
    public static void main(String[] args) {  
        System.out.println(3/0);  
    }  
}
```

1.2. Sửa đoạn chương trình sau để bắt ngoại lệ

```
import javax.swing.JOptionPane;  
  
public class Main {  
  
    public static void main(String[] args) {  
  
        // Enter the following values and see what exception is caught.  
        // 0, 4, character, <no value>  
        try {  
  
            String value = JOptionPane.showInputDialog(null, "Enter value:");  
  
            // Nếu giá trị nhập vào không phải là số sẽ xảy ra ngoại lệ  
            NumberFormatException  
            int divisor = Integer.parseInt(value);  
  
            // If the divisor is 0, it will result in ArithmeticException  
            System.out.println(3/divisor);  
  
        } catch ( ??? ) {  
            ???  
        } catch ( ??? ) {  
            ???  
        }  
        System.out.println("After exception.");  
    }  
}
```

1.3. Sửa đoạn chương trình sau để bắt ngoại lệ `ArrayIndexOutOfBoundsException` và các ngoại lệ khác nếu có

```
import java.io.*;

public class ListOfNumbers {

    private int [] arr;

    private static final int SIZE = 10;

    public ListOfNumbers() {

        arr = new int [SIZE];

        for (int i = 0; i < SIZE; i++)

            arr[i]=i;

    }

    public void writeList() {

        PrintWriter out = null;

        try {

            System.out.println("Entering try statement");

            out = new PrintWriter(new FileWriter("OutFile.txt"));

            for (int i = 0; i < (SIZE + 1) ; i++)

                System.out.println("Value at: " + i + " = " + arr[i]);

        } catch (???) {

            ???

        } catch (???) {

            ???

        } finally {

            if (out != null) {

                System.out.println("Finally: Closing PrintWriter");
```

```

        out.close();
    } else {
        System.out.println("Finally: PrintWriter not open");
    }
}

}

public static void main(String[] args) {
    ListOfNumbers lst= new ListOfNumbers();

    lst.writeList();
}
}

```

2. Sửa lại phương thức “cat” sau đây để nó có thể chạy được:

```

public static void cat(File file) {
    RandomAccessFile input = null;
    String line = null;

    try {
        input = new RandomAccessFile(file, "r");
        while ((line = input.readLine()) != null) {
            System.out.println(line);
        }
        return;
    } finally {
        if (input != null) {
            input.close();
        }
    }
}

```

Hướng dẫn:

```

public static void cat(File file) {
    RandomAccessFile input = null;
    String line = null;

    try {
        input = new RandomAccessFile(file, "r");
        while ((line = input.readLine()) != null) {
            System.out.println(line);
        }
        return;
    }
}

```

```

    } catch(FileNotFoundException fnf) {
        System.err.format("File: %s not found%n", file);
    } catch(IOException e) {
        System.err.println(e.toString());
    } finally {
        if (input != null) {
            try {
                input.close();
            } catch(IOException io) {
            }
        }
    }
}

```

3. Bổ sung xử lý ngoại lệ cho chương trình sau

```

public static void test() {

    int a[] = new int[5];

    a[5] = 30 / 0;

}

```

4. Bổ sung câu lệnh ném ra một ngoại lệ `InvalidInput` trong khối `if (number>99)`

```

class InvalidInput {
    ...
};

int factorial(int number) {
    int facto=1;
    if (number > 99) {
        _____

    } else {
        facto = 1;
        int n = 1;
        while (++n <= number) {
            facto *= n;
        }
    }
}

```

5. Sử dụng cuối cùng `{ /*...*/ }` để bắt và xử lý ngoại lệ

```

class InvalidInput ... {
    ...
};

static int factorial(int number) {
    int facto;
    if (number > 99) {
        throw InvalidInput();
    }
}

```

```

        } else {
            facto = 1;
            int n = 1;
            while (++n <= number) {
                facto *= n;
            }
        }
    }

    public static void main(String [] args) {
        int inputX, result;

        // nhập dữ liệu cho inputX
        try {
            result =factorial( inputX);

        } // end of try block
        { /*...*/
        }
    }
}

```

6. Tự tạo một lớp ngoại lệ mới và xây dựng ví dụ minh họa: ném và bắt ngoại lệ này.

7. Tự tạo 2 lớp ngoại lệ mới có quan hệ cha, con và xây dựng ví dụ minh: ném và bắt ngoại lệ cha, con.