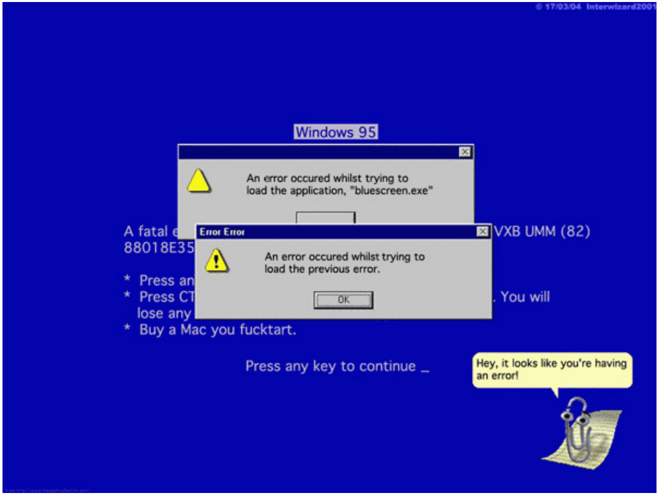
Обработка ошибок, исключения

Алексей Владыкин



© 17/03/04 Interwizard2001

Windows 95

An error occured whilst trying to
load the application, "bluescreen.exe"

Error Error

An error occured whilst trying to
load the previous error.

OK

A fatal e VXB UMM (82)
88018E35

* Press an
* Press CT . You will
lose any
* Buy a Mac you fucktart.

Press any key to continue _

Hey, it looks like you're having
an error!

```java
public interface Calculator {

    double calculate(String expr);

}
```

```java
public class CalculatorImpl
        implements Calculator {

    @Override
    public double calculate(String expr) {
        // ...
        System.exit(1);
        // ...
    }

}
```

```java
public class CalculatorImpl
        implements Calculator {

    @Override
    public double calculate(String expr) {
        // ...
        return Double.NaN;
        // ...
    }

}
```

```java
public class CalculatorImpl
        implements Calculator {

    @Override
    public Result calculate(String expr) {
        // ...
        return Result.error();
        // ...
    }
}
```

```java
public class CalculatorImpl
        implements Calculator {

    private boolean error;

    @Override
    public double calculate(String expr) {
        // ...
        error = true;
        return 0;
        // ...
    }

    public boolean isError() {
        return error;
    }
}
```

```java
Object nullRef = null;

// java.lang.NullPointerException
nullRef.toString();
```

```
java.lang.NullPointerException
    at org.stepic.java.exception.Test.baz(Test.java:19)
    at org.stepic.java.exception.Test.bar(Test.java:14)
    at org.stepic.java.exception.Test.foo(Test.java:10)
    at org.stepic.java.exception.Test.main(Test.java:6)
```

```java
int[] array = {1, 2, 3};

// java.lang.ArrayIndexOutOfBoundsException
array[10];
```

```java
// java.io.FileNotFoundException
new FileInputStream("not_existing_file");
```

java.lang.Throwable

```java
throw new IllegalStateException(
        "Invalid user. " +
        "Please replace user and continue.");
```

```java
package java.lang;

public class Throwable {

    public String getMessage() { /*...*/ }

    public void printStackTrace() { /*...*/ }

    public StackTraceElement[] getStackTrace() { /*...*/ }

    public Throwable getCause() { /*...*/ }

    public Throwable[] getSuppressed() { /*...*/ }

    // ...
}
```

Классификация исключений

- Исключительные ситуации в JVM
  `java.lang.Error`
- Исключительные ситуации в пользовательском коде
  - Проверяемые (checked)
    `java.lang.Exception`
  - Непроверяемые (unchecked)
    `java.lang.RuntimeException`

java.lang.Error

- java.lang.OutOfMemoryError
- java.lang.NoClassDefFoundError
- java.lang.VerifyError

java.lang.Exception

```java
import java.io.IOException;

public class ExceptionDemo {

    public void someMethod() {
        // ...
        throw new IOException("Failed to read file");
        // ...
    }
}
```

java.lang.RuntimeException

- java.lang.NullPointerException
- java.lang.ArrayIndexOutOfBoundsException
- java.lang.ArithmeticException

Собственное исключение

```java
public class CalculatorException extends RuntimeException {

    public CalculatorException(String message) {
        super(message);
    }

    public CalculatorException(String message, Throwable cause) {
        super(message, cause);
    }
}
```

```java
public class CalculatorImpl
        implements Calculator {

    @Override
    public double calculate(String expr) {
        // ...
        throw new CalculatorException(
                "Unsupported operator found");
        // ...
    }
}
```

Перехват исключения: `try-catch`

```
for (;;) {
    System.out.print("Enter expression: ");
    String expr = readUserInput();
    if (expr == null || "exit".equalsIgnoreCase(expr)) {
        break;
    }
    try {
        double result = calculator.calculate(expr);
        System.out.println("Result: " + result);
    } catch (CalculatorException e) {
        System.out.print("Bad expression: " + e.getMessage());
        System.out.print("Please try again: ");
    }
}
```

Перехват нескольких исключений

```
try {
    // ...
} catch (FirstException e) {
    e.printStackTrace();
} catch (SecondException e) {
    e.printStackTrace();
}

// since Java 7 can be replaced with:
try {
    // ...
} catch (FirstException | SecondException e) {
    e.printStackTrace();
}
```

finally

```
InputStream is = new FileInputStream("a.txt");
try {
    readFromInputStream(is);
} finally {
    is.close();
}
```

finally

```
InputStream is = new FileInputStream("a.txt");
try {
    readFromInputStream(is);
} finally {
    try {
        is.close();
    } catch (IOException e) {
        // ignore
    }
}
```

try с ресурсами

```
try (InputStream is =
        new FileInputStream("a.txt")) {
    readFromInputSteam(is);
}
```

try с ресурсами

```
InputStream is = new FileInputStream("a.txt");
try {
    readFromInputStream(is);
} catch (Throwable t) {
    try {
        is.close();
    } catch (Throwable t2) {
        t.addSuppressed(t2);
    }
    throw t;
}
is.close();
```

```java
package java.lang;

public interface AutoCloseable {

    void close() throws Exception;
}
```

Обработка исключения

```
try {
    // code throwing MyException
} catch (MyException e) {
    // ???
}
```

Плохой пример

```java
String string;
try {
    string = object.toString();
} catch (NullPointerException e) {
    string = "null";
}
System.out.println(string);
```

Хороший пример

```
String string = object == null
        ? "null"
        : object.toString();

System.out.println(string);
```

```java
package org.stepic.java.logging;

import java.util.logging.*;

public class LogDemo {

    private static final Logger LOGGER =
      Logger.getLogger(LogDemo.class.getName());

    // ...

}
```

```java
LOGGER.log(Level.INFO, "I'm logging");

// SEVERE, WARNING, INFO,
// CONFIG, FINE, FINER, FINEST


LOGGER.warning("We have a problem!");
LOGGER.log(Level.FINEST,
        "Current value of x is " + x);
```

```java
LOGGER.log(Level.FINEST,
        "Current value of x is {0}", x);
```

```java
LOGGER.log(Level.FINEST,
        "Point coordinates are ({0}, {1})",
        new Object[] {x, y});
```

```java
LOGGER.log(Level.SEVERE,
        "Unexpected exception", e);
```

`java.util.logging.Handler`

- Обработчик сообщения
  Определяет, куда будет записано сообщение

- `java.util.logging.ConsoleHandler`
- `java.util.logging.FileHandler`
- `java.util.logging.SocketHandler`

java.util.logging.Formatter

- Определяет формат вывода

- `java.util.logging.SimpleFormatter`
- `java.util.logging.XMLFormatter`