

Логирование в Java приложениях

Слайды и примеры кода

<https://github.com/khremmin/logging-samples>

Что было в курсе "Основы языка Java"

- `e.printStackTrace()`
- `System.out.println()`, `System.err.println()`
- `FileOutputStream`, `PrintWriter` ...

Проблемы при использовании этих подходов

- Логирование в файл
- Ротация логов
- Многопоточность
- Разбиение на уровни

A close-up shot from the movie Inception showing Leonardo DiCaprio and Matt Damon. DiCaprio is on the left, looking slightly to the right with a serious expression. Damon is on the right, seen in profile, looking towards DiCaprio. The background is dark and out of focus, suggesting an indoor setting with some light sources.

WE NEED TO GO

DEEPER

Конец лихих 90ых – начало 00

- Java молода и непорочна
- У каждого свой велосипед для логирования
- Зарождается log4j

Конец лихих 90ых – начало 00. log4j

- Логгер, категория
- Appenders
- Layout
- Log level
- Иерархичность категорий

log4j породил:

- log4cxx
- log4net
- log4php
- logging в Python 2.x

Конец лихих 90ых – начало 00

- [JSR47](#)
- IBM
- Graham Hamilton
- IBM logging toolkit
- JUL – java.util.logging
- JCL - Jakarta Commons Logging (Apache Commons Logging)
- SLF4J
- Logback (<http://logback.qos.ch/>)

Logback. Зависимости

- slf4j-api
- logback-core
- logback-classic

Logback. Hello, world!

```
package ru.miet.logback;

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class App {
    private static Logger logger = LoggerFactory.getLogger(App.class);

    public static void main(String[] args) {

        logger.trace("Trace test");

        logger.debug("Debug test");

        logger.info("Info test");

        logger.warn("Warn test");

        logger.error("Error test");

        try {
            Integer.parseInt("anything");
        } catch (NumberFormatException e) {
            logger.error(e.getMessage(), e);
        }
    }
}
```

Logback. Hello, world!

```
08:41:18.801 [main] DEBUG ru.miet.logback.App - Debug test
08:41:18.808 [main] INFO ru.miet.logback.App - Info test
08:41:18.808 [main] WARN ru.miet.logback.App - Warn test
08:41:18.808 [main] ERROR ru.miet.logback.App - Error test
08:41:18.811 [main] ERROR ru.miet.logback.App - For input string: "anything"
java.lang.NumberFormatException: For input string: "anything"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:580)
    at java.lang.Integer.parseInt(Integer.java:615)
    at ru.miet.logback.App.main(App.java:23)
    at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
    at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62)
    at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
    at java.lang.reflect.Method.invoke(Method.java:498)
    at com.intellij.rt.execution.application.AppMain.main(AppMain.java:144)

Process finished with exit code 0
```

Logback. Hello, world!

```
<configuration>
```

```
  <appender name="console" class="ch.qos.logback.core.ConsoleAppender">
```

```
    <encoder>
```

```
      <pattern>%d{dd.MM.yyyy HH:mm:ss.SSS} [%thread] %-5level %logger{36} -  
%msg%n</pattern>
```

```
    </encoder>
```

```
  </appender>
```

```
  <root level="trace">
```

```
    <appender-ref ref="console" />
```

```
  </root>
```

```
</configuration>
```

Logback. Hello, world!

```
01.06.2016 08:46:22.755 [main] TRACE ru.miet.logback.App - Trace test
01.06.2016 08:46:22.760 [main] DEBUG ru.miet.logback.App - Debug test
01.06.2016 08:46:22.760 [main] INFO ru.miet.logback.App - Info test
01.06.2016 08:46:22.761 [main] WARN ru.miet.logback.App - Warn test
01.06.2016 08:46:22.761 [main] ERROR ru.miet.logback.App - Error test
01.06.2016 08:46:22.763 [main] ERROR ru.miet.logback.App - For input string: "anything"
java.lang.NumberFormatException: For input string: "anything"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:580)
    at java.lang.Integer.parseInt(Integer.java:615)
    at ru.miet.logback.App.main(App.java:23) <5 internal calls>
+
Process finished with exit code 0
```

Logback. Уровни лога

| Уровень запроса | Уровень, указанный в настройке | | | | | |
|-----------------|--------------------------------|-------|------|------|-------|-----|
| | TRACE | DEBUG | INFO | WARN | ERROR | OFF |
| TRACE | + | - | - | - | - | - |
| DEBUG | + | + | - | - | - | - |
| INFO | + | + | + | - | - | - |
| WARN | + | + | + | + | - | - |
| ERROR | + | + | + | + | + | - |

Logback. Appenders

- ConsoleAppender
- FileAppender
- RollingFileAppender
- DBAppender
- ServerSocketAppender
- SSLServerSocketAppender
- SMTPAppender

Logback. Layouts

- XMLLayout
- HTMLLayout
- PatternLayout
- EchoLayout
- TTLLayout

Logback. Layouts

Log session start time Wed Jun 01 09:19:13 MSK 2016

| Date | Level | Thread | Logger | Message |
|---|--------------|--------|---------------------------|------------------------------|
| 2016-06-01 09:19:13.401 | TRACE | main | ru.miet.logback.App | Trace test |
| 2016-06-01 09:19:13.412 | DEBUG | main | ru.miet.logback.App | Debug test |
| 2016-06-01 09:19:13.413 | INFO | main | ru.miet.logback.App | Info test |
| 2016-06-01 09:19:13.413 | WARN | main | ru.miet.logback.App | Warn test |
| 2016-06-01 09:19:13.414 | ERROR | main | ru.miet.logback.App | Error test |
| 2016-06-01 09:19:13.422 | ERROR | main | ru.miet.logback.App | For input string: "anything" |
| java.lang.NumberFormatException: For input string: "anything" at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65) at java.lang.Integer.parseInt(Integer.java:580) at java.lang.Integer.parseInt(Integer.java:615) at ru.miet.logback.App.main(App.java:23) at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method) at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:62) at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43) at java.lang.reflect.Method.invoke(Method.java:498) at com.intellij.rt.execution.application.AppMain.main(AppMain.java:144) | | | | |
| 2016-06-01 09:19:13.425 | INFO | main | ru.miet.logback.test.Smth | Info test from Smth |
| 2016-06-01 09:19:13.426 | WARN | main | ru.miet.logback.test.Smth | Warn test from Smth |
| 2016-06-01 09:19:13.427 | ERROR | main | ru.miet.logback.test.Smth | Error test from Smth |

Best practices

Правильное объявление логера

```
public class App {  
    private static final Logger logger = LogManager.getLogger(App.class);  
  
    public static void main(String[] args) {  
        ...  
    }  
}
```

Используйте правильный уровень

- Error – фатальная ошибка
- Warn – любое исключение
- Info – обычное информирование
- Debug, trace – только для программиста.
- Возможны проблемы с производительностью

Избегайте любых операций над строками

- Плохо

logger.debug("User with id = " + userId + " is not found");

- Плохо

logger.debug(String.format("User with id = %d is not found", userId));

- Хорошо

logger.debug("User with id = {} is not found", userId);

Логи всегда на английском языке

- Плохо

logger.debug("Пользователь не найден");

- Плохо, но жить можно

logger.debug("Pol'zovatel' ne nayden.");

- Хорошо

logger.debug("User is not found");

Не теряйте stack trace

- Плохо

```
try {  
    Connection connection = getConnection();  
} catch (SQLException e) {  
    logger.error("SQL error");  
}
```

- Хорошо

```
try {  
    Connection connection = getConnection();  
} catch (SQLException e) {  
    logger.error("SQL error", e); // logger.error(e.getMessage(), e);  
}
```


Избыточное логирование

- Плохо

```
logger.error("SQL error");
```

```
throw new SQLException("Can't connect to DB.");
```

Не используйте дефолтные настройки

- Настройка в файл, либо другое хранилище
- Настройка ротации, если используется логирование в файл
- Настройка адекватного паттерна

Логирование входа/выхода

```
public long getSmth(String firstParam, String secondParam) {  
    logger.debug("First param = {}, Second param = {}", firstParam,  
secondParam);  
    ...  
    long smth = ...  
    logger.debug("Return smth = {}", smth);  
    return smth;  
}
```

Только trace или debug

- <https://habrahabr.ru/post/113145/>
- <https://jcp.org/en/jsr/detail?id=47>
- <http://logging.apache.org/log4j/2.x/>
- <http://slf4j.org/>
- <http://logback.qos.ch/>
- <http://logback.qos.ch/manual/layouts.html#conversionWord>