

Логиирование

# Немножко истории

Изначально в Java не было своего логгера, что привело к написанию нескольких независимых логгеров. Самым распространенным из них стал log4j.

Спустя несколько лет, в Java все же был добавлен свой логгер, но его функциональность была гораздо беднее и большого распространения он не получил.

Факт, как говорится, на лицо – в Java есть официальный логгер, но все сообщество Java-программистов предпочитает пользоваться другим.

# Процесс логирования

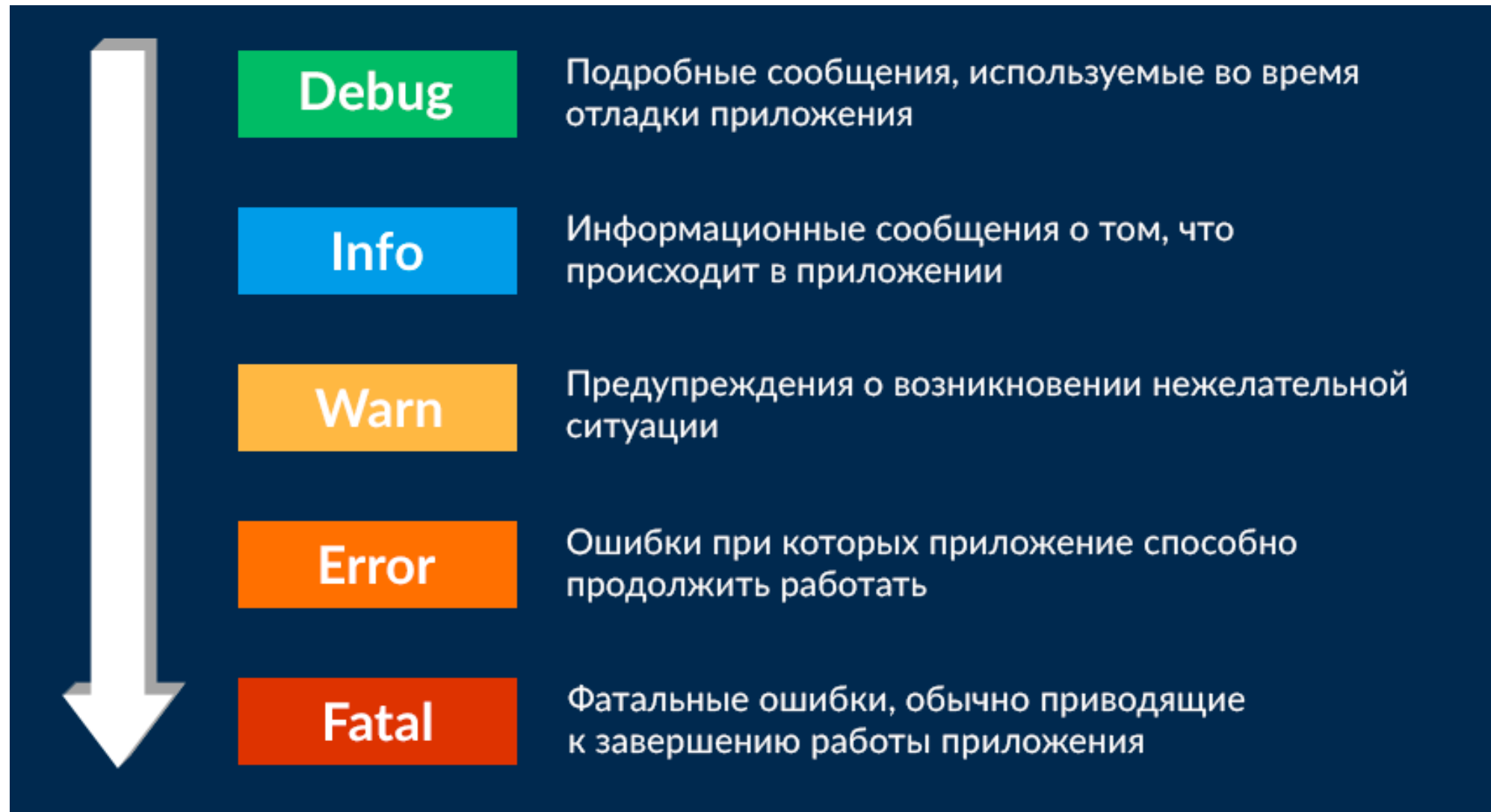
- **Первая часть** – это сбор информации.
- **Вторая часть** – это фильтрование собранной информации.
- **Третья часть** – это запись отобранной информации.

# Сбор информации

```
class Manager
{
    private static final Logger logger = LoggerFactory.getLogger(Manager.class);

    public boolean processTask(Task task)
    {
        logger.debug("processTask id = " + task.getId());
        try
        {
            task.start();
            task.progress();
            task.complete();
            return true;
        }
        catch (Exception e)
        {
            logger.error("Unknown error", e);
            return false;
        }
    }
}
```

# Степень важности



# Как работают уровни логирования

Если выставить уровень логирования в WARN, то все сообщения, менее важные, чем WARN будут отброшены: TRACE, DEBUG, INFO.

Если выставить уровень фильтрации в FATAL, то будут отброшены даже ERROR'ы.

# log4j.properties – запись в консоль

# Root logger option

log4j.rootLogger=INFO, stdout

# Direct log messages **to** stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd  
HH:mm:ss}

# log4j.properties – запись в файл

# Root logger option

log4j.rootLogger=INFO, file

# Direct log messages **to** a log file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=C:\\\\logging.log

log4j.appender.file.MaxFileSize=1MB

log4j.appender.file.MaxBackupIndex=1

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern= %-5p %c{1}:%L - %m%n



# log4j.properties – запись в консоль и файл

# Root logger option

log4j.rootLogger=INFO, file, stdout

# Direct log messages **to** a log file

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=C:\\\\logging.log log4j.appender.file.MaxFileSize=1MB

log4j.appender.file.MaxBackupIndex=1

log4j.appender.file.threshold=**DEBUG**

log4j.appender.file.layout=org.apache.log4j.PatternLayout

log4j.appender.file.layout.ConversionPattern= %-5p %c{1}:%L - %m%n

# Direct log messages **to** stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.Target=System.out

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%d{yyyy-MM-dd HH:mm:ss}

# log4j2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<Configuration status="INFO">
  <Appenders>
    <Console name="ConsoleAppender" target="SYSTEM_OUT">
      <PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n" />
    </Console>
    <File name="FileAppender" fileName="application-${date:yyyyMMdd}.log" immediateFlush="false" append="true">
      <PatternLayout pattern="%d{yyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>
    </File>
  </Appenders>
  <Loggers>
    <Root level="debug">
      <AppenderRef ref="ConsoleAppender"/>
      <AppenderRef ref="FileAppender"/>
    </Root>
  </Loggers>
</Configuration>
```