

Java 7: Что дальше?

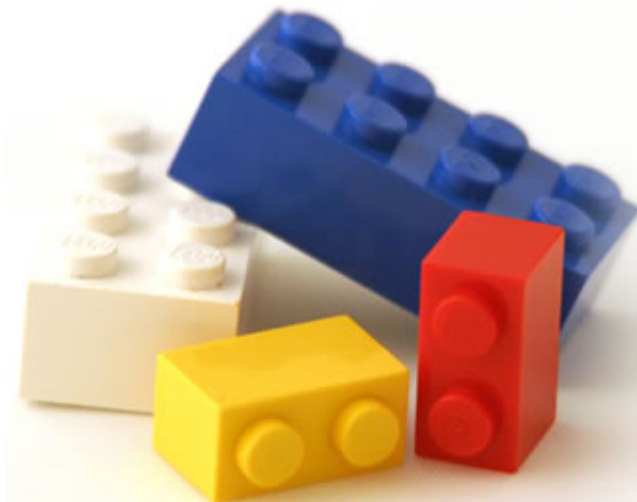
Alex Miller, BEA Systems
Перевод Андрей Гребнев

Java 7 Цели

- Java модульность
 - Супер-пакеты
 - Система модулей Java
 - Ядро Java
- Swing
 - Swing App Framework
 - Связывание бинов
 - Валидация бинов
- Generics и типы
 - Материлизация Generic
 - Типизированные аннотации
 - Быстрое создание экземпляров
- NIO 2
- Поддержка XML
- Поддержка свойств
- Замыкания
- invokedynamic
- Дата & Время API
- Единицы измерения
- JMX
 - JMX 2.0
 - Удаленный коннектор API
- Javadoc

Модульность Java

- Суперпакеты (JSR 294)
- Модульная система Java (JSR 277)
- Ядро Java (по JSR)

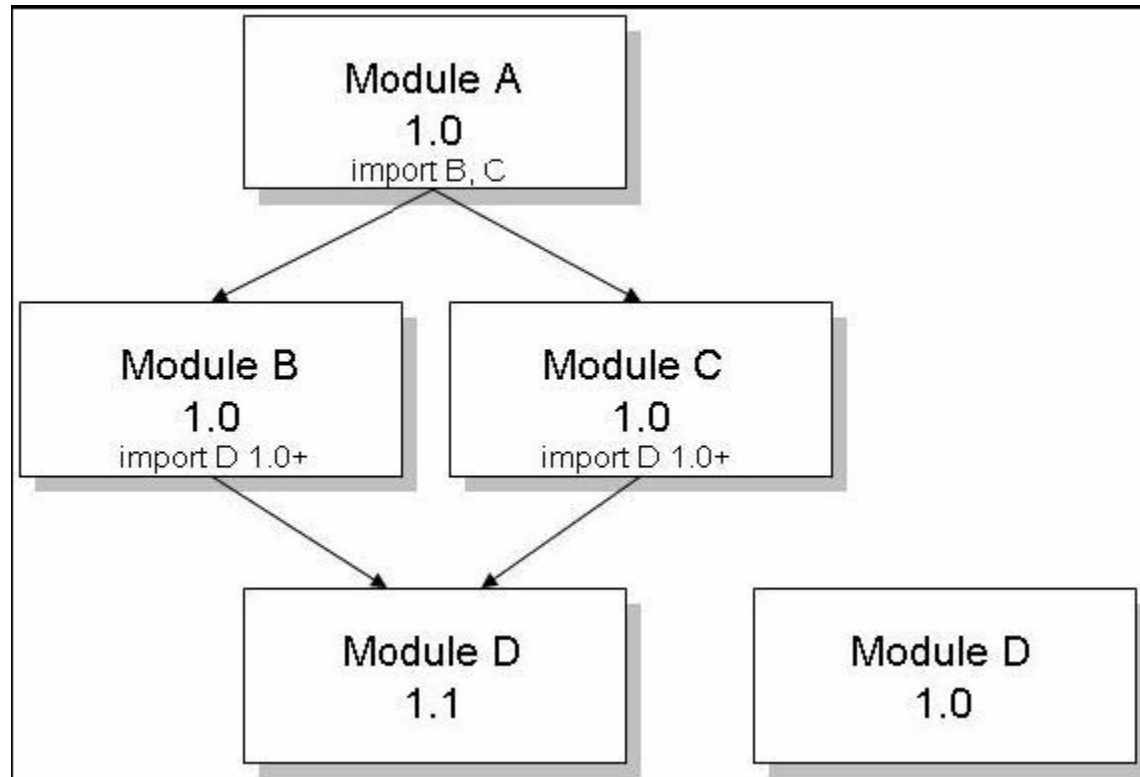


Суперпакеты (JSR 294)

```
src/com/foo/super-package.java:
```

```
superpackage com.foo {  
    member package com.foo.api, com.foo.model,  
    com.foo.util;  
    export com.foo.api; // Сделать api пакета публичным  
}
```

Модульная система Java (JSR 277)



Опред. метаданных модуля

/MODULE-INF/METADATA.module:

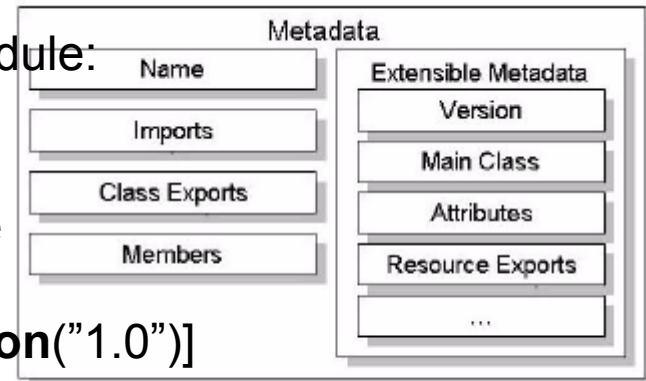
name=com.wombat.webservice

extensible-metadata=[**@Version**("1.0")]

members=[com.wombat.webservice.MainApp,
com.wombat.webservice.PrivateClassA,
com.wombat.webservice.PrivateInterfaceB]

imports=[ImportModule(org.foo.xml,
@VersionConstraint("1.3.0"),
ImportModule(org.foo.soap,
@VersionConstraint("2.0.0+"))]

class-exports=[com.wombat.webservice.MainApp]



The metadata of a module definition.

Ядро Java

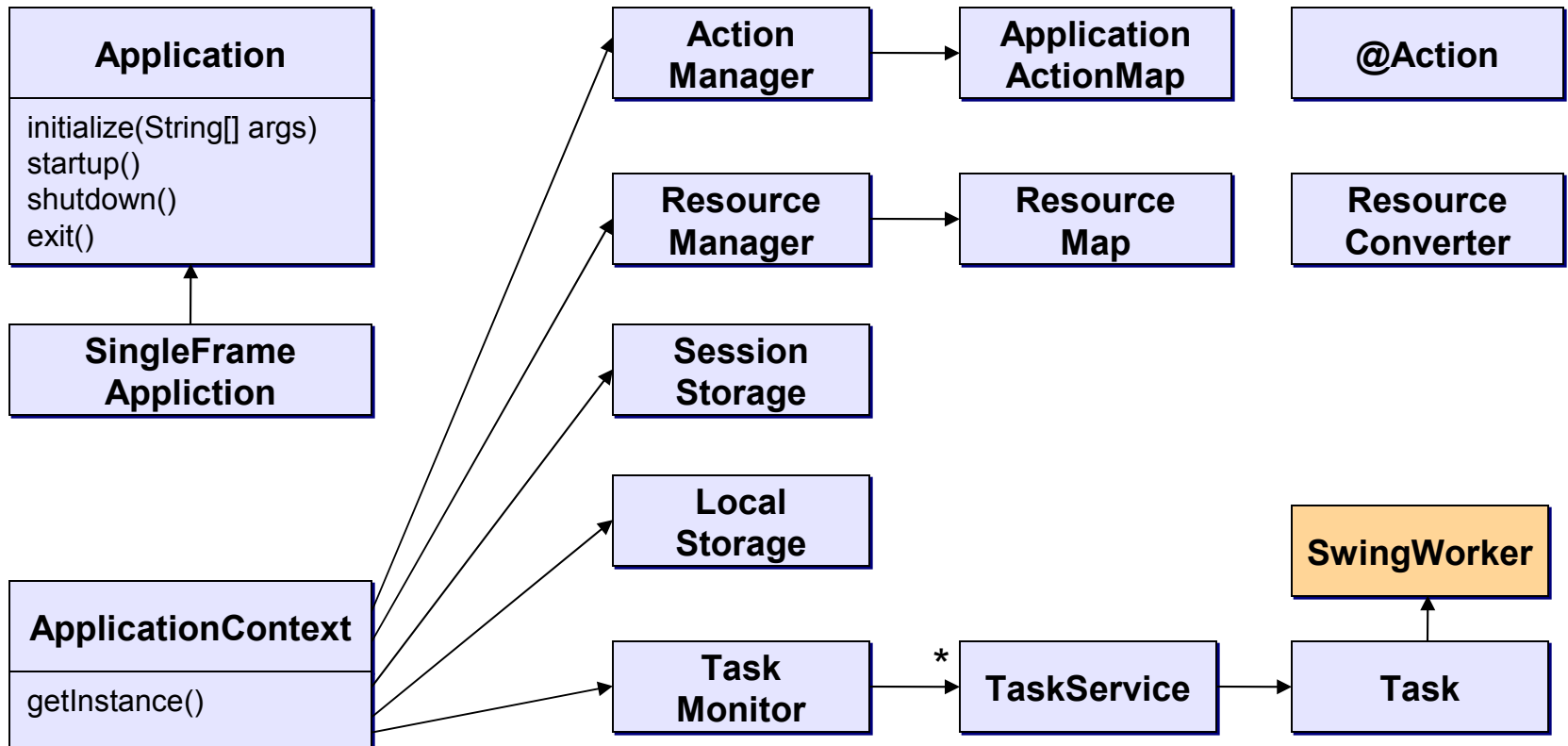
- Реорганизация JRE в ядро + опциональные модули, загружаемые динамически
- Конкуренция с Flex, Silverlight
- Текущий прототип 2.5M

Улучшения Swing

- Swing Application Framework (JSR 296)
- Связывание бинов (JSR 295)
- Валидация бинов (JSR 303)



Swing Application Framework JSR 296



Связывание бинов

JSR 295

- Связывание двух свойств разных бинов:

```
Binding binding = new Binding(source, "sourcePath",  
    target, "targetPath");  
binding.bind();
```

- Один раз связав, изменения источника отображаются на цель
- Можно использовать `BindingConverter` и `BindingValidator`
- Язык выражений JSP (EL) используется

Валидация бинов JSR 303

- Это пример из Hibernate Validator (похожее будет в JSR 303):

```
public class Address {
    @NotNull
    private String line1;
    private String line2;
    private String zip;
    private String state;

    @Length(max = 20)
    @NotNull
    private String country;

    @Range(min = -2, max = 50, message = "Floor out of range")
    public int floor;
}
```

Материализация Generics

```
class NewCollection<class E> extends  
    Collection<E> { ... }
```

```
class NewList<class E> extends  
    NewCollection<E>, List<E> { ... }
```

Получение типа Generic в runtime

Аннотации типов Java

JSR 308

- для аргументов generic типа для параметризации классов:
`Map<@NonNull String, @NotEmpty List<@ReadOnly Document>> files;`
- для аргументов generic типа в generic в вызове метода и конструктора:
`o.<@NonNull String>m("...");`
- для ограничений и шаблонов параметров:
`class Folder<F extends @Existing File> { ... }
Collection<? super @Existing File>`
- для наследования классов:
`class UnmodifiableList<T> implements @ReadOnly List<@ReadOnly T>
{ ... }`
- для вызова исключений:
`void monitorTemperature() throws @Critical TemperatureException { ... }`
- для приведения типов:
`myString = (@NonNull String) myObject;`
- для проверки типов:
`boolean isNonNull = myString instanceof @NonNull String;`
- для создания объектов:
`new @NotEmpty @ReadOnly List<String>(myNonEmptyStringSet)`

Быстрое создание экземпляра

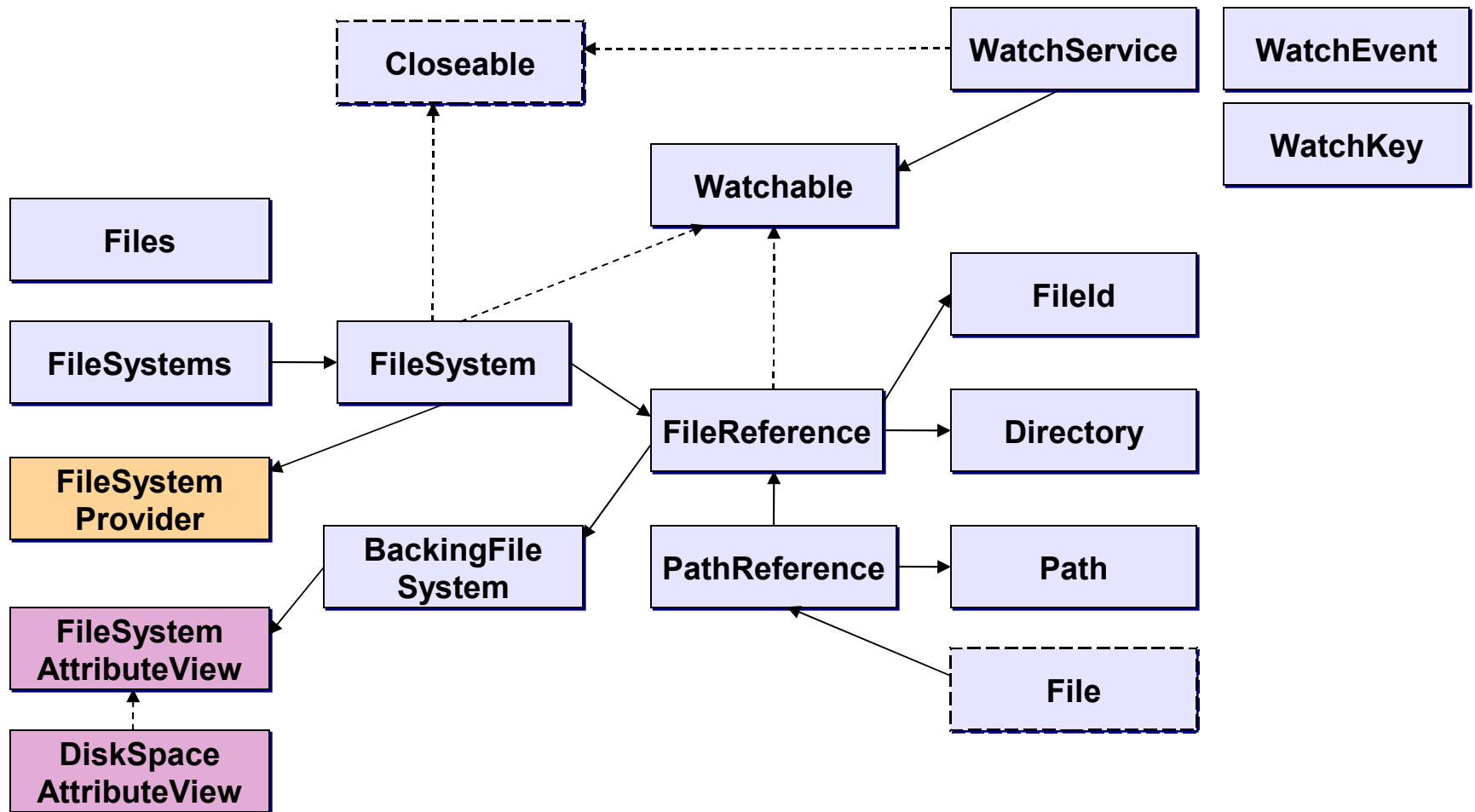
```
public static void main(String[] args) {  
    map := new HashMap<String, Integer>();  
    for(word : args) {  
        freq := map.get(word);  
        map.put(word, (freq==null) ? 1 : freq+1);  
    }  
    System.out.println(map); }  
}
```

```
public static void main(String[] args) {  
    final map = new HashMap<String, Integer>();  
    for(final word : args) {  
        final freq = map.get(word);  
        map.put(word, (freq==null) ? 1 : freq+1);  
    }  
    System.out.println(map); }  
}
```

NIO 2 (JSR 203)

- Основные фиши:
 - Новый интерфейс файловой системы
 - Разрешения
 - Атрибуты файлов
 - Настраиваемая реализация
 - Исключено использование API зависящего от ФС
 - API для асинхронных I/O сокетов и файлов
 - Законченное решение для multicast и пр.

NIO 2 – API ΦC



NIO 2: Копирование файлов

```
PathReference source =  
    PathReference.from("spongebob.jpg");  
PathReference target =  
    PathReference.from("squarepants.jpg");  
int flags = CopyFlag.COPY_ATTRIBUTES |  
            CopyFlag.REPLACE_EXISTING;  
source.copyTo(target, flags);
```

NIO 2 – Разрешение доступа к файлам

```
PathReference ref = PathReference.from("foo.txt");
PosixFileAttributeView view =
    ref.newFileAttributeView(PosixFileAttributeView.class);
PosixFileAttributes attrs = view.readAttributes();
int perms = attrs.getPermissions();

// prints "-rw-r--r-- alice bandits"
System.out.format("%s\t%s\t%s\n",
    PosixFilePermission.toString(perms), attrs.getOwner(),
    attrs.getGroup());

// deny others
perms &= ~OTHERS_READ & ~OTHERS_WRITE & ~OTHERS_EXECUTE;
view.updatePermissions(perms);

// change group
UserPrincipal cops = view.lookupPrincipalByGroupName("cops");
view.updateOwners(null, cops);
```

Поддержка XML: Создание

```
void addReviewer(XML feature,  
                String reviewer,  
                String time) {  
  
    feature.add(<reviewed>  
              <who>{ reviewer }</who>  
              <when>{ time }</when>  
              </reviewed>);  
}
```

Поддержка XML: Код. данных

```
package java.xml;
abstract class DataCoder {
    String encode(Object o);
    String encode(Object o, String type);
    <T> T decode(String data, Class<T> c);
    <T> T decode(String data, Class<T> c, String type);
    boolean supports(Class<?> c);
    boolean supports(Class<?> c, String type);

    static final DataCoder JAVA;
    static final DataCoder XSD;
}
```

Поддержка XML: пример код. данных

```
void addReviewer(XML feature,  
                String reviewer,  
                Timestamp time) {  
  
    DataCoder dc = DataCoder.XSD;  
    feature.add(<reviewed>  
               <who>{ reviewer }</who>  
               <when>{ dc.encode(time) }</when>  
               </reviewed>);  
}
```

Поддержка XML: Навигация

```
void rejectOpenFeatures(XML featureList) {  
    List<XML> fs = featureList.findAll(  
        "feature[state!='approved']");  
  
    for(XML f : fs) {  
        f.get("state").set("rejected");  
    }  
}
```

Поддержка свойств

```
public class Point {  
    property double x;  
    property double y;  
}  
Point p = new Point(1, 2);  
System.out.println(p.x + " " + p.y);
```

```
Point p = new Point();  
p->X = 56;  
p->Y = 87;  
int z = p->X;
```

Замыкание - Абстракция

```
void validateWithMax(Stat[] stats, final int max) throws InvalidStatException {  
    for(Stat stat : stats) {  
        if(! (stat.value() <= max)) {  
            throw new InvalidStatException();  
        }  
    }  
}
```

```
void validateWithMin(Stat[] stats, final int min) throws InvalidStatException {  
    for(Stat stat : stats) {  
        if(! (stat.value() >= min)) {  
            throw new InvalidStatException();  
        }  
    }  
}
```

Замыкание – Внутренний класс

```
void validateWithMax(Stat[] stats, final int max) throws InvalidStatException {  
    validate( stats, new Validator() {  
        public boolean isValid(Stat stat) { return stat.value() <= max; }  
    }); }  
}
```

```
void validateWithMin(Stat[] stats, final int min) throws InvalidStatException {  
    validate( stats, new Validator() {  
        public boolean isValid(Stat stat) { return stat.value() >= min; }  
    }); }  
}
```

```
interface Validator { boolean isValid(Stat stat); }  
void validate(Stat[] stats, Validator validator) {  
    for(Stat stat : stats) {  
        if(! validator.isValid(stat)) { throw new InvalidStatException(); }  
    }  
}
```

Замыкание: предложения

- BGGA – Gilad Bracha, Neal Gafter, James Gosling, Peter von der Ahe
- CICE – “Concise Instance Creation Expressions”
 - Josh Bloch, Bob Lee, Doug Lea
- FCM – “First Class Methods” / JCA “Java Control Abstraction”
 - Stephen Colebourne, Stefan Schulz, Ricky Clarkson

Замыкание - CICE

```
void validateWithMax(Stat[] stats, int max) throws InvalidStatException
{
    validate( stats,
        Validator(Stat stat) {
            return stat.value() <= max;
        });
}
```

- Интерфейс Validator и метод validate() не изменились.

Замыкание - BGGA

```
void validateWithMax(Stat[] stats, int max) throws InvalidStateException {  
    validate(stats, {Stat stat =>  
        stat.value() <= max  
    });  
}
```

```
void validate(Stat[] stats, {Stat stat => boolean} block) {  
    for(Stat stat : stats) {  
        if(! block.invoke(stat)) {  
            throw new InvalidStateException();  
        }  
    }  
}
```

Замыкание - FCM

```
void validateWithMax(Stat[] stats, final int max) throws InvalidStateException {  
    validate( stats,  
        #(boolean(Stat stat)) {  
            return stat.value() <= max;  
        } );  
}
```

```
void validate(Stat[] stats, #(boolean(Stat)) block) {  
    for(Stat stat : stats) {  
        if(! block.invoke(stat)) {  
            throw new InvalidStateException();  
        }  
    }  
}
```

Замыкание – Контроль

```
File file = ...;
FileReader reader = null;
try {
    reader = new FileReader(file);
    reader.read...
} finally {
    if (reader != null) {
        try {
            reader.close();
        } catch (IOException e) {
            // ignore
        }
    }
}
```

Замыкание - JCA

```
File file = ...;  
usingFileReader(FileReader reader : file) {  
    reader.read...  
}
```

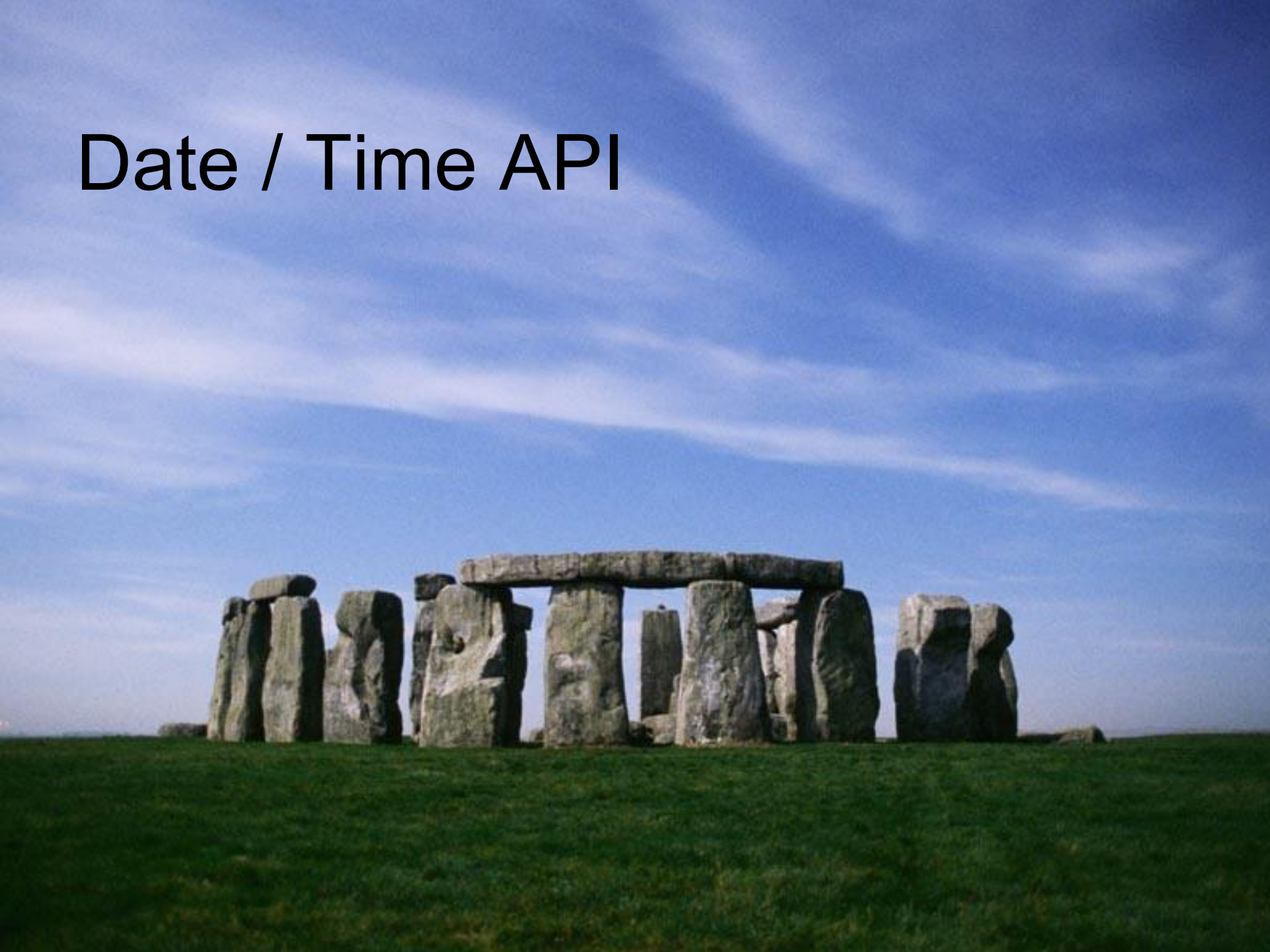
```
public static void usingFileReader(#(void(FileReader)) block : File file) throws  
    IOException {  
    FileReader reader = null;  
    try {  
        reader = new FileReader(file);  
        block.invoke(reader);  
    } finally {  
        if (reader != null) {  
            try {  
                reader.close();  
            } catch (IOException e) { }  
        }  
    }  
}
```

Замыкание - BGGA

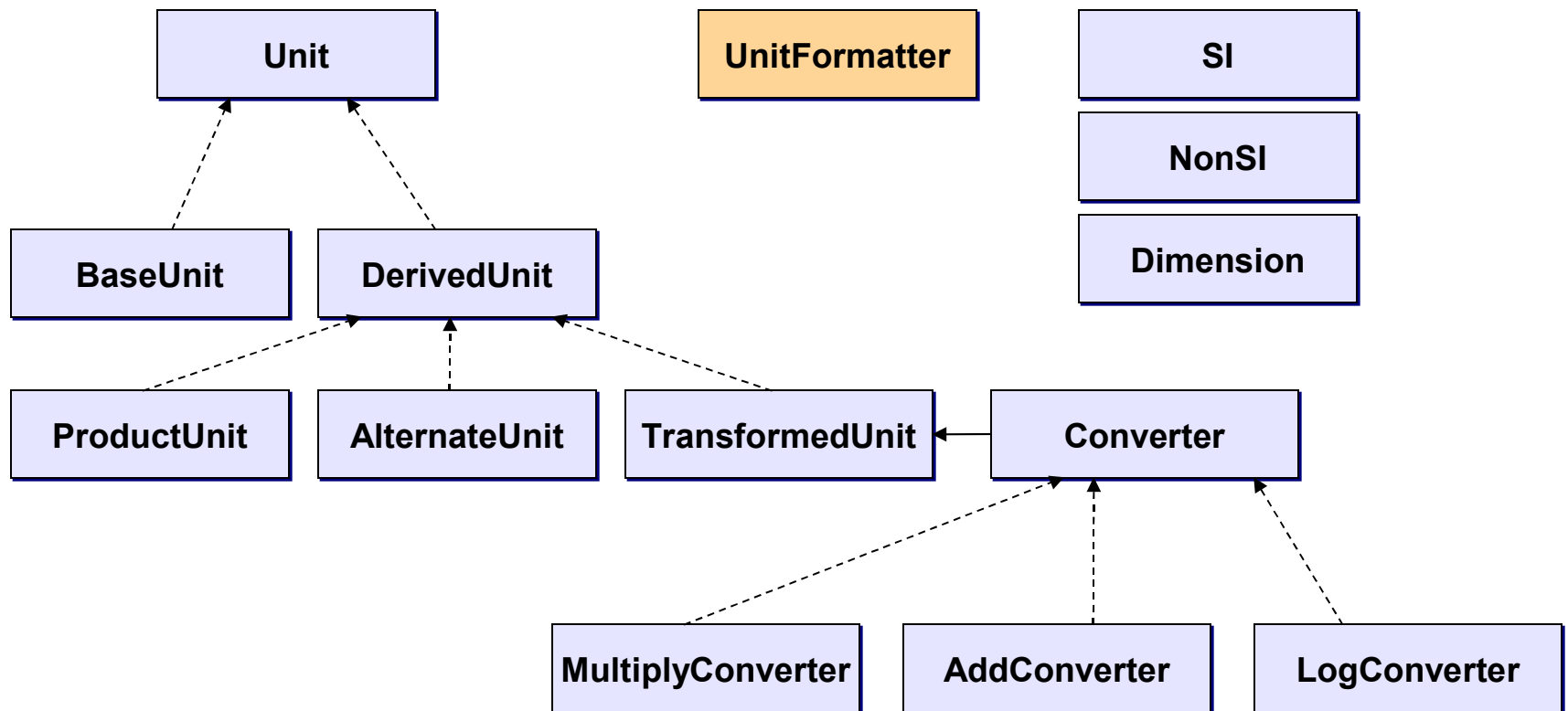
```
File file = ...;
usingFileReader(FileReader reader : file) {
    reader.read...
}
```

```
public static <T,throws E extends Exception>
T usingFileReader(File file, {=>T throws E} block) throws E {
    FileReader reader = null;
    try {
        reader = new FileReader(file);
        return block.invoke(reader);
    } finally {
        if (reader != null) {
            try {
                reader.close();
            } catch (IOException e) { }
        }
    }
}
```

Date / Time API



Размерности (JSR 275)



Размерности: примеры

// Конвертация.

```
KILO(METER) .getConverterTo(MILE) .convert(10)
```

- 6.2137119223733395

// Системные размерности.

```
REVOLUTION.divide(MINUTE) .getSystemUnit()
```

- rad/s

// Проверка

```
ELECTRON_VOLT.isCompatible(WATT.times(HOUR))
```

- true


// Извлечение размерности

```
ELECTRON_VOLT.getDimension()
```

- $[L]^2 \cdot [M] / [T]^2$

JMX

- JMX 2.0 (JSR 255)
 - Модификация с generics
 - Использование аннотаций
 - Теперь открытые Mbeans использовать легче
 - Унификация мониторов для использования не простых типов
 - Каскадируемые MBean сервера
- Коннектор веб-сервисов (JSR 262)



Javadoc (JSR 260)

- Новые теги
- Новая категоризация типов методов, бинов, свойств бинов и пр.
- Новый внешний вид

Miscellaneous

- **BigDecimal operator support**

```
BigDecimal a = ...
```

```
BigDecimal b = ...
```

```
BigDecimal s = a + b;
```

- **Allow String literals in switch case blocks**

```
switch(stooge) {
```

```
    case 'Moe': ...
```

- **Comparison support for enums**

```
JACK < QUEEN
```



Спасибо!

- Блог Alex Miller:

<http://tech.puredanger.com>