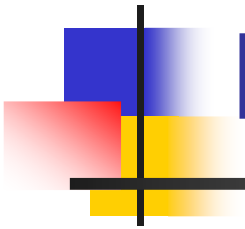
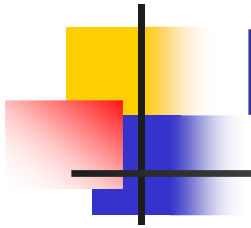


Programowanie w języku Java

Wykład 1: Wprowadzenie

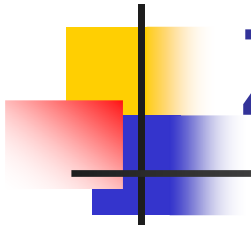
http://eclipse.elektron.pk.edu.pl/~sdeniziak/

Hasło: java2012



Literatura

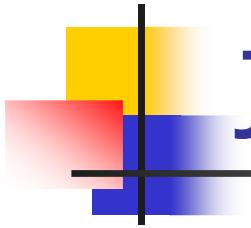
- Bruce Eckel, „Thinking in Java – edycja polska”, wydanie. 4, Helion
- Cay S. Horstmann, Gary Cornell, *Java 2. Podstawy,* Helion, 2003.
- Cay S. Horstmann, Gary Cornell, *Java 2. Techniki zaawansowane,* Helion, 2005.



Zakres wykładu

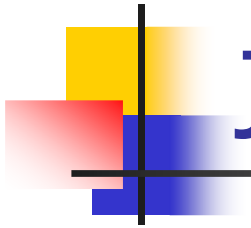
- Przegląd i uzupełnienie konstrukcji języka Java:
  - Programowanie współbieżne
  - Programowanie rozproszone
  - Grafika
- Java Micro Edition
- Java Standard Edition
- Java Enterprise Edition

Podstawowe konstrukcje języka Java (1)



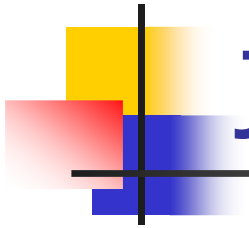
- Klasy
  - pola i metody
  - konstruktory
  - kwalifikatory dostępu: public, private, protected
  - pola i metody statyczne
  - dziedziczenie
  - klasy, pola, metody, parametry ostateczne

Podstawowe konstrukcje języka Java (2)



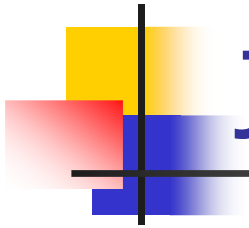
- Interfejsy
  - wielodziedziczenie
  - polimorfizm

Podstawowe konstrukcje języka Java (3)



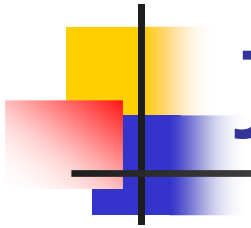
- Klasy kolekcji
  - listy
  - zbiory
  - mapy
  - kolejki

Podstawowe konstrukcje języka Java (4)



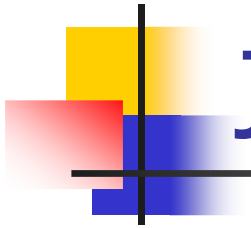
- Wyjątki
  - fraza try catch finally
  - hierarchia wyjątków
  - propagacja wyjątków

Podstawowe konstrukcje języka Java (5)



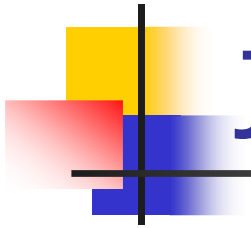
- Strumienie We/Wy
  - strumienie podstawowe
  - strumienie „opakowujące”
  - serializacja obiektów

Podstawowe konstrukcje języka Java (6)



- JFC/Swing, Aplety
  - GUI
  - Aplety: init(), start(), stop(), destroy()

Podstawowe konstrukcje języka Java,cd



- Wątki współbieżne

Nowe konstrukcje JDK 5.0 (1)

- Typy uogólnione

```
static void expurgate(Collection c) {
   for (Iterator i = c.iterator(); i.hasNext(); )
     if (((String) i.next()).length() == 4)
         i.remove();
}
```

JDK5.0

```
static void expurgate(Collection <String> c) {
  for (Iterator <String> i = c.iterator(); i.hasNext(); )
       if (i.next().length() == 4)
      i.remove();
}
```

<?> - dowolna klasa

<? extends Klasa> - dowolna podklasa

Sparametryzowane metody:

```
public static <T, S extends T>
   void copy(List<T> dest,
         List<S> src) {...}
```

Nowe konstrukcje JDK 5.0 (2)

- Instrukcja For-Each

```
void cancelAll(Collection<TimerTask> c) {
   for (Iterator<TimerTask> i = c.iterator(); i.hasNext(); )
                  i.next().cancel();
}
```

```
void cancelAll(Collection<TimerTask> c) {
      for (TimerTask t : c)
          t.cancel();
}
```

for each TimerTask t in c

Nowe konstrukcje JDK 5.0 (3)

- Autoboxing

```
import java.util.*;
public class Frequency {
 public static void main(String[] args) {
 Map<String, Integer> m = new TreeMap<String, Integer>();
  for (String word : args) {
       Integer freq = m.get(word);
      m.put(word, (freq == null ? 1 : freq + 1));
   }
 System.out.println(m);
 }
}
```

Nowe konstrukcje JDK 5.0 (4)

- Typ wyliczeniowy

```
enum Day { SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY,
SATURDAY }

for (Day d : EnumSet.range(Day.MONDAY, Day.FRIDAY))
        System.out.println(d);

public enum Operation {
        PLUS, MINUS, TIMES, DIVIDE;
        double eval(double x, double y){
        switch(this) {
        case PLUS: return x + y;
        case MINUS: return x - y;
        case TIMES: return x * y;
        case DIVIDE: return x / y;
        } throw new AssertionError("Unknown op: " + this); }
}
```

Nowe konstrukcje JDK 5.0 (5)

- Zmienna liczba parametrów

```
public static String format(String pattern, Object... arguments);
```

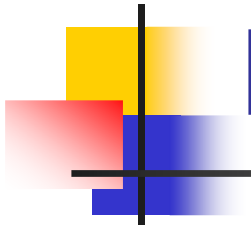
Tablica lub lista argumentów (tylko jako ostatni argument!)

Nowe konstrukcje JDK 5.0 (6)

- Statyczne importowanie

```
double r = Math.cos(Math.PI * theta);
```

```
import static java.lang.Math.*;
double r = cos(PI * theta);
```

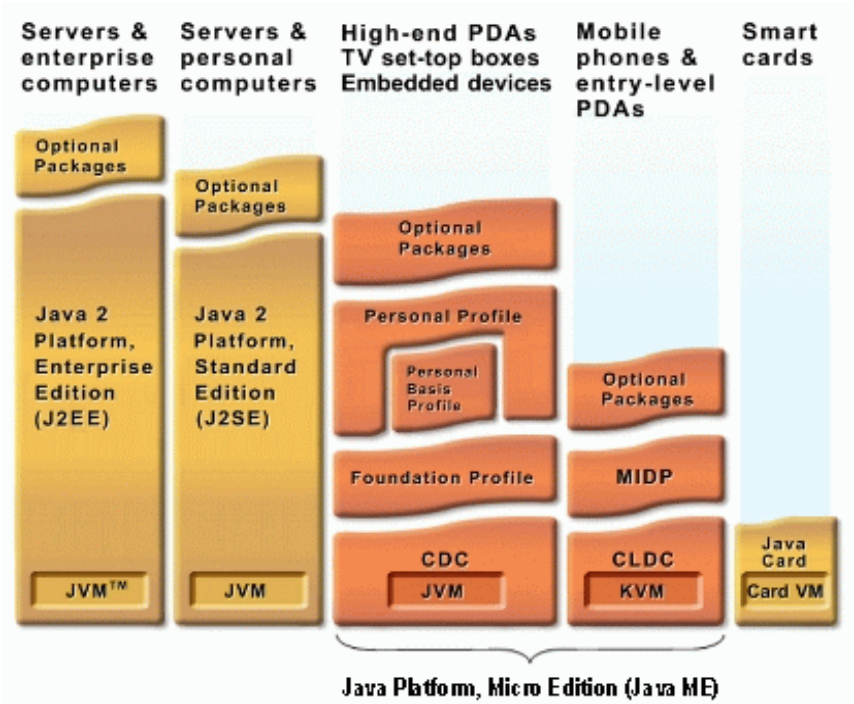


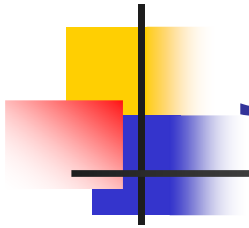
Programy w języku Java

- Aplety
- Aplikacje
- Inne:
  - Serwlety
  - Midlety
  - Xlety

Środowiska programowania

- Java Platform, Micro Edition
- Java Platform, Standard Edition
- Java Platform, Enterprise Edition

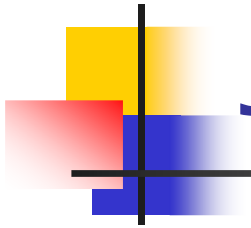




JME: Konfiguracje

Standard dla pewnych grup urządzeń

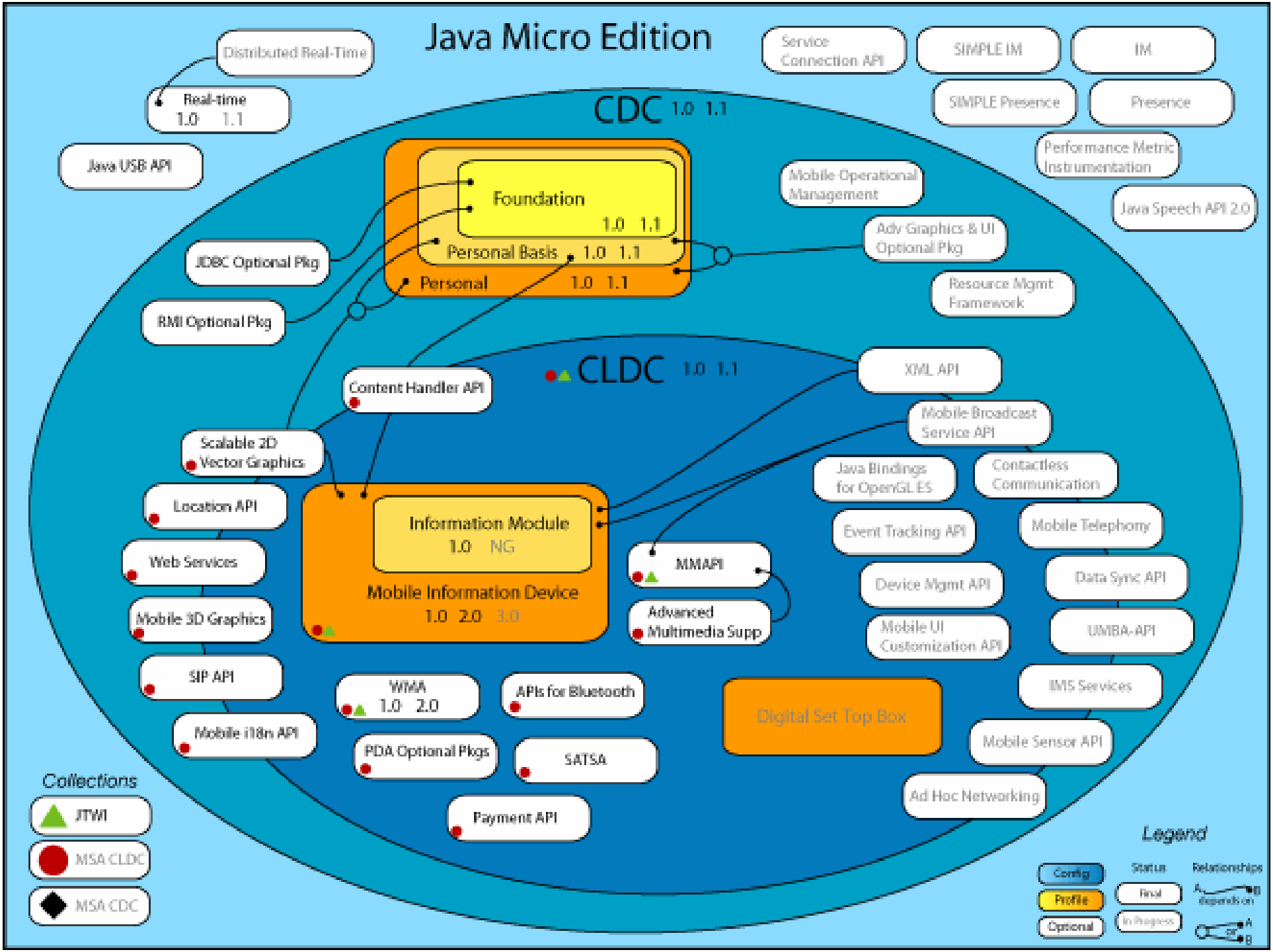
- Connected Limited Devices Configuration (CLDC)
  - procesor 16-bitowy, 192kB RAM
  - telefony komórkowe, notesy elektroniczne, itp.
- Connected Device Configuration (CDC)
  - procesor 32-bitowy, 2MB RAM
  - palmtopy, smartfony

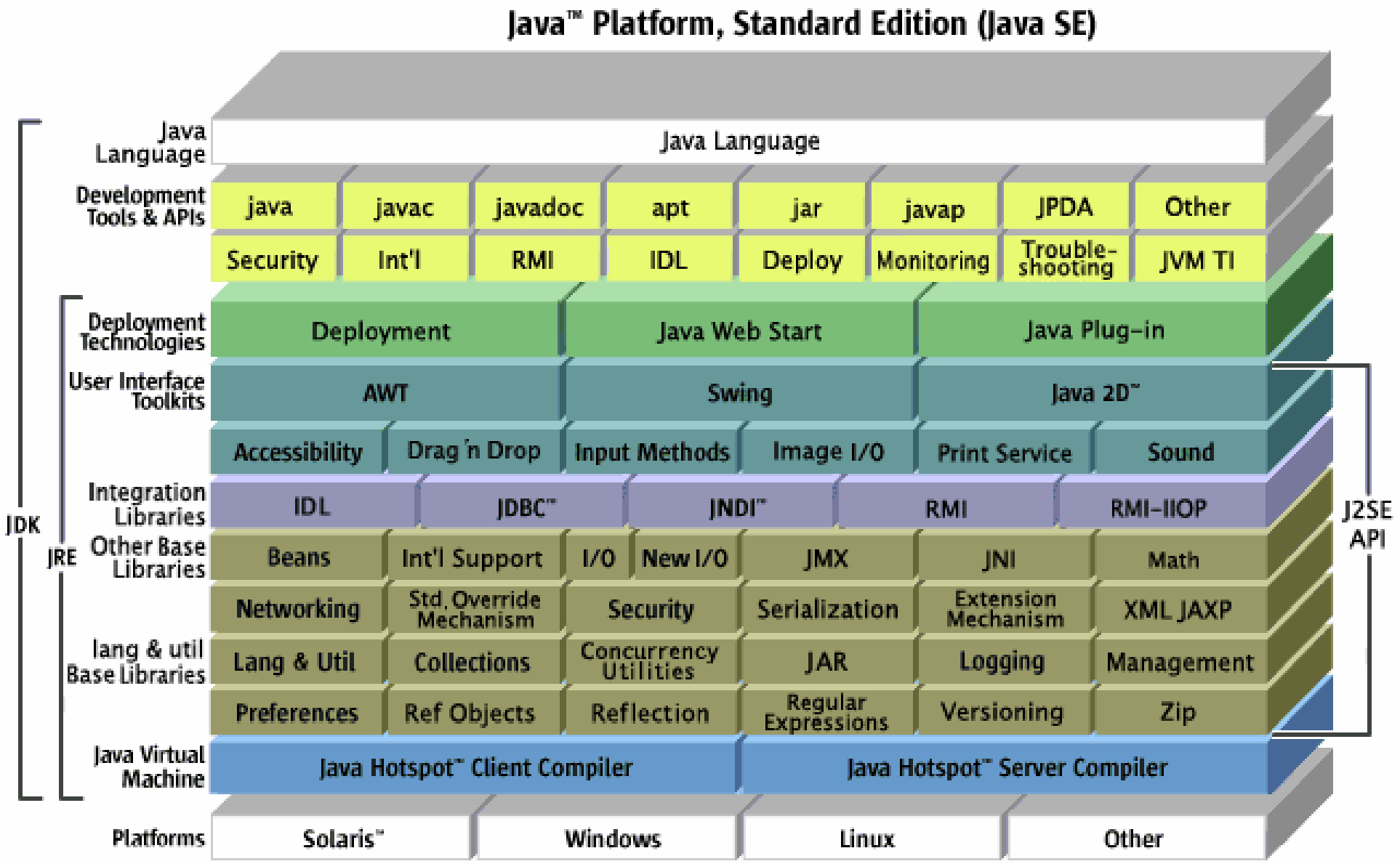


JME: Profile

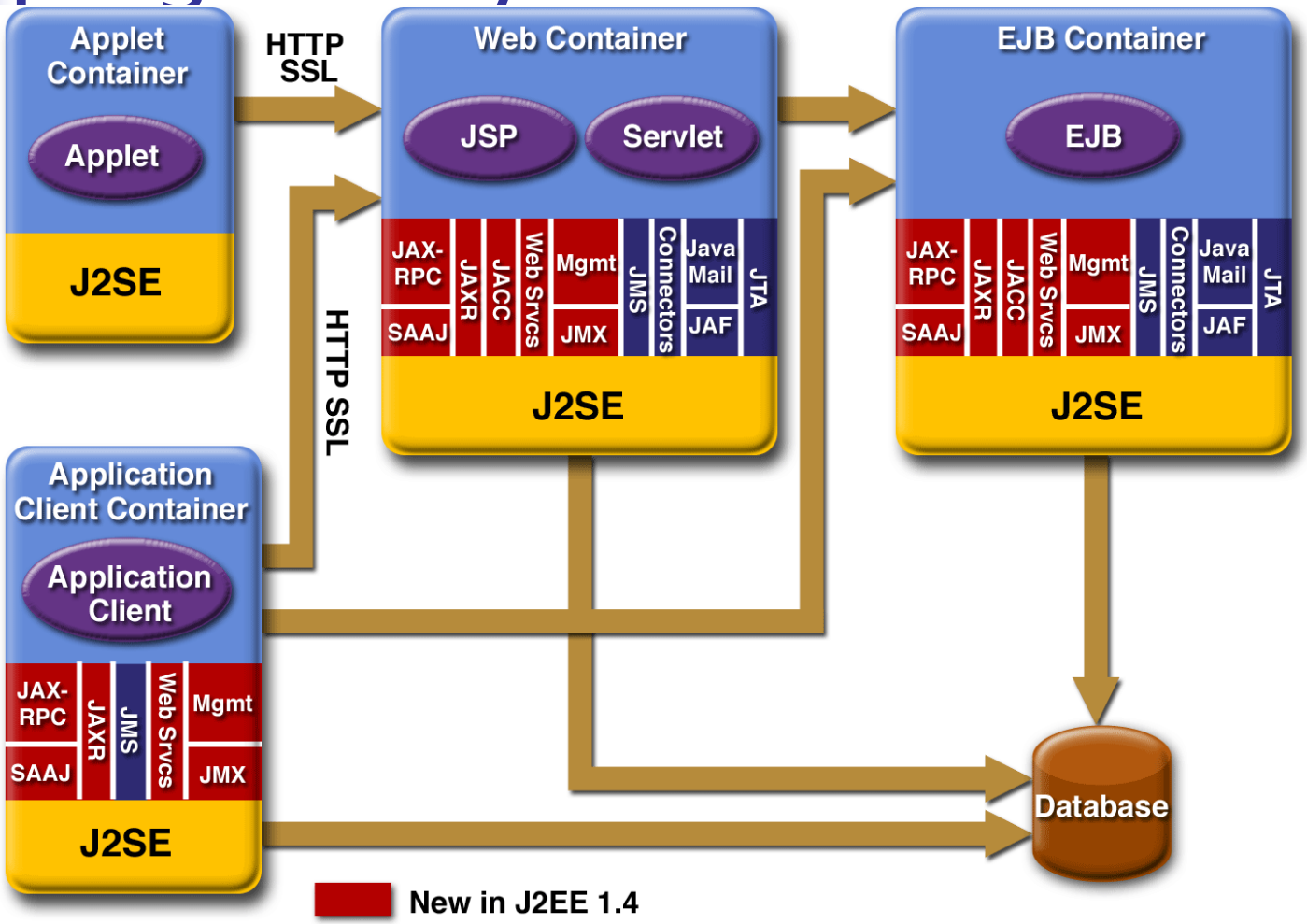
Standard API dla pewnej grupy urządzeń w ramach danej konfiguracji:

- **MIDP (Mobile Information Device Profile) – CLDC**
  - **uproszczone GUI (dla LCD), midlety**
- **Foundation Profile – CDC**
  - **bez GUI, standardowe klasy Javy**
- **Personal Basis Profile – CDC**
  - **xlet**
- **Personal Profile – CDC**
  - **aplety i AWT**


Java Micro Edition
Distributed Real-Time
Real-time
1.0 1.1
Java USB API
Service Connection API
SIMPLE IM
IM
SIMPLE Presence
Presence
Performance Metric Instrumentation
Java Speech API 2.0
CDC 1.0 1.1
Foundation
1.0 1.1
Personal Basis 1.0 1.1
Personal 1.0 1.1
Mobile Operational Management
Adv Graphics & UI Optional Pkg
Resource Mgmt Framework
JDBC Optional Pkg
RMI Optional Pkg
CLDC 1.0 1.1
Content Handler API
XML API
Mobile Broadcast Service API
Scalable 2D Vector Graphics
Java Bindings for OpenGL ES
Contactless Communication
Location API
Information Module
1.0 NG
Mobile Information Device
1.0 2.0 3.0
Event Tracking API
Mobile Telephony
Web Services
MMAPI
Device Mgmt API
Data Sync API
Mobile 3D Graphics
Advanced Multimedia Supp
Mobile UI Customization API
UMBA-API
SIP API
WMA
1.0 2.0
APIs for Bluetooth
Digital Set Top Box
IMS Services
Mobile i18n API
PDA Optional Pkgs
SATSA
Mobile Sensor API
Ad Hoc Networking
Payment API
Collections
JTWI
MSA CLDC
MSA CDC
Legend
Config
Profile
Optional
Status
Final
In Progress
Relationships
A depends on B
A or B

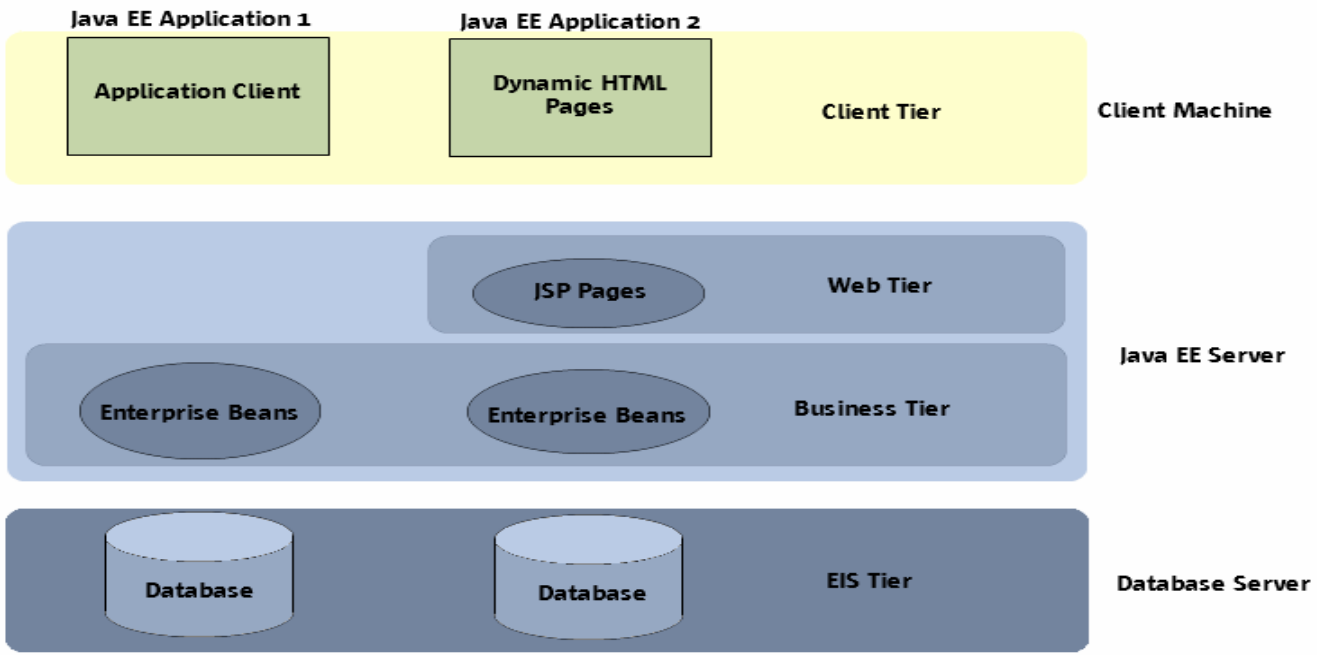

Java™ Platform, Standard Edition (Java SE)
Java Language
Java Language
Development Tools & APIs
java
javac
javadoc
apt
jar
javap
JPDA
Other
Security
Int'l
RMI
IDL
Deploy
Monitoring
Trouble-shooting
JVM TI
Deployment Technologies
Deployment
Java Web Start
Java Plug-in
User Interface Toolkits
AWT
Swing
Java 2D™
Accessibility
Drag 'n Drop
Input Methods
Image I/O
Print Service
Sound
Integration Libraries
IDL
JDBC™
JNDI™
RMI
RMI-IIOP
Other Base Libraries
Beans
Int'l Support
I/O
New I/O
JMX
JNI
Math
Networking
Std. Override Mechanism
Security
Serialization
Extension Mechanism
XML JAXP
lang & util Base Libraries
Lang & Util
Collections
Concurrency Utilities
JAR
Logging
Management
Preferences
Ref Objects
Reflection
Regular Expressions
Versioning
Zip
Java Virtual Machine
Java Hotspot™ Client Compiler
Java Hotspot™ Server Compiler
Platforms
Solaris™
Windows
Linux
Other
JDK
JRE
J2SE API

J2EE –środowisko programistyczne



Standard J2EE

Java 2 Enterprise Edition



Koniec