

# פיתוח מערכות תוכנה מבוססות Java

---

## תצורה (Configuration)

אוהד ברזילי

# System Configuration

---

שליטה על התצורה (configuration) של תוכנית היא היכולת לבצע התאמות לתוכנית לאחר סיום תהליך הפיתוח

- פרמטריזציה

ההתאמות ישפיעו על אופן הריצה (mode) של התוכנית

לתוכנות שונות יש התאמות שונות שנרצה לבצע

דרכי ההתאמה, המנגנונים והשיטות לבקרת תצורה בתוכנה מודרנית רבות ומגוונות

# תצורה של מה?

---

העברת פרמטרים לתוכנית

של הפלטפורמה:

חומרה

מערכות הפעלה

דפדפנים

טלפונים ניידים

Enterprise Software

Application Servers

Databases

של התהליכים הארגוניים (שקף הבא)

# תוכנות אירגוניות (Enterprise Software)

---

התאמה והטמעה

Amdocs vs. SAP

התוכנה נכתבת פעמיים

# שיטות לבקרת תצורה

---

- כתיבת קוד ידני
  - "קרא קובץ myapp.conf"
- שימוש בשיטות של שפת התכנות
  - Annotations
  - java.util.Properties
- טיפול בבקרת התצורה ע"י הפלטפורמה
  - למסגרות עבודה רבות קיימים תקנים המגדירים את הפרמטרים ברי השינוי ואת הדרכים לשלוט בהם
  - לדוגמא: Web Servers, Hibernate
  - XML , Annotations
- שימוש בכלים ייעודיים לתצורה
  - Scripts, ANT

# בקרת תצורה ברמת עץ הפיתוח

---

CVS Branch

Merging and Branch

מה קורה כשמוצאים באג בחלק המשותף? ■

CVS vs. GIT

Product Line vs. Configurable Product

# בקרת תצורה בזמן התקנה

---

□ אוטומטי או ידני

■ Windows של CD

□ קבצים מקומיים או שימוש ב Registry

□ תכנות הצהרתי (declarative programming)

■ XML

■ Annotation (לא קשור לקונפיגורציה)

# בקרת תצורה בשלב הפיתוח

---

בשלב הבדיקות

#DEBUG ■

שימוש בתסריטים:

ANT, make, shell scripts ■

sed, awk, perl ■

java.util.prefs

java.util.Properties

java.util.PropertyResourceBundle ■



# בקרת תצורה בשלב איטגרציה/בדיקות

---

סימולטורים

Images

ghost

# תצורת הפרויקט

---

## □ Maven

- Featuring program-build capabilities
- high-level project management tools
- project-oriented approach
- project's dependencies

## ■ Nice introduction:

- <http://www.alphacsp.com/Events/Ultimate-IDE/Presentations/The%20Maven2%20Revolution.pdf>

# A POM Example

---

```
<project>
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.alphacsp</groupId>
  <artifactId>my-app</artifactId>
  <packaging>jar</packaging>
  <version>1.0-SNAPSHOT</version>
  <dependencies>
    <dependency>
      <groupId>junit</groupId>
      <artifactId>junit</artifactId>
      <version>3.8.1</version>
      <scope>test</scope>
    </dependency>
  </dependencies>
</project>
```

- All the information required for the build in a single file

# What is Ant?

---

- Ant is an open source project
  - <http://ant.apache.org/>
  - Java-based build tool
  - Used on any Java 1.1+ system
  - Command line driven
  - Integrated with IDEs
  - Uses XML files to describe the build process

# Ant's Design Goals

---

- Simple to use
- Easy to understand
- Can be extended

# Ant Motivation

---

- ❑ Build, test and deploy Java Projects
- ❑ Workflow engine
- ❑ Web publishing engine
- ❑ Support .Net and other languages
- ❑ Automation

# Ant common tasks

---

- Archive Tasks, Audit/Coverage Tasks, Compile Tasks, Deployment Tasks, Documentation Tasks, EJB Tasks, Execution Tasks, File Tasks, Java2 Extensions Tasks, Logging Tasks, Mail Tasks, Miscellaneous Tasks, .NET Tasks, Pre-process Tasks, Property Tasks, Remote Tasks, SCM Tasks, Testing Tasks, Visual Age for Java Tasks
  
- User defined tasks
  
- Nice tutorial:
  - <http://supportweb.cs.bham.ac.uk/docs/tutorials/docs/system/build/tutorials/ant/ant.html>

# Real Build File...

```
<?xml version="1.0" ?>
<project name="MYPROJECT" default="compile" >

  <target name="init">
    <mkdir dir="build/classes" />
    <mkdir dir="bin" />
  </target>

  <target name="compile" depends="init">
    <javac srcdir="src" destdir="build/classes" />
  </target>

  <target name="archive" depends="compile">
    <jar destfile="bin/myproject.jar"
        basedir="build/classes" />
  </target>
</project>
```



# Real Build File...

```
<target name="execute" depends="compile">
  <java
    classname="FirstClass"
    classpath="build/classes">
    <arg value="first argument passed"/>
    <arg value="second argument passed"/>
  </java>
</target>

<target name="clean" depends="init">
  <delete dir="build" />
  <delete dir="bin" />
</target>

</project>
```

# איך נחליף את שפת המערכת?

---

i18n

<http://www.google.com/transconsole>

# DHCP

---

- ❑ The Dynamic Host Configuration Protocol (DHCP)
- ❑ Automates the assignment of IP addresses, subnet masks, default gateway, and other IP parameters
- ❑ When a DHCP-configured machine boots up, it sends a query requesting necessary information from a DHCP server
- ❑ The DHCP server manages a pool of IP addresses and also has information about client configuration parameters such as the default gateway, the domain name, the DNS servers, other servers such as time servers, and so forth

# plugins

---

Eclipse Plugins

Facebook Plugins

טיענת מחלקות דינאמית:

Class loaders and Reflection

# הורשה וקונפיגורציה

---

□ לא ברמת המערכת אלא ברמת הרכיב

Factory Design Pattern □