



Διαδίκτυο και Εφαρμογές



Εργαστήριο #2

04/05/2022

Δημιουργία Αλληλεπιδραστικού Γραφικού Περιβάλλοντος και (Επικοινωνία με Σχεσιακές Βάσεις)



Εισαγωγή

- Διαδίκτυο και Εφαρμογές
 - Labs 1, 2, 3 – Desktop Εφαρμογές (GUI, DB Connection, Files)
 - Swing, DBs, **Threads**, XML, JSON, Files,
 - Labs 1, 2, 4 – Web Υπηρεσίες & Εφαρμογές
 - **HTML**, **CSS**, **JavaScript**, **AJAX**, JSP, Servlet, **Threads**, XML, JSON, Files, DBs
- Java Memory Allocation
 - Αντικειμενοστραφής Γλώσσα Προγραμματισμού, Υψηλού Επιπέδου
 - **Διαχείριση Μνήμης / Garbage Collector**
 - **Java Heap Size**

Runtime.getRuntime()

 - Memory used to store Objects
 - **Control** Java heap size using Command-Line Parameters
 - -Xms : Initial Java heap size (e.g., -Xms512M)
 - -Xmx : Maximum heap size (e.g., -Xmx1G)



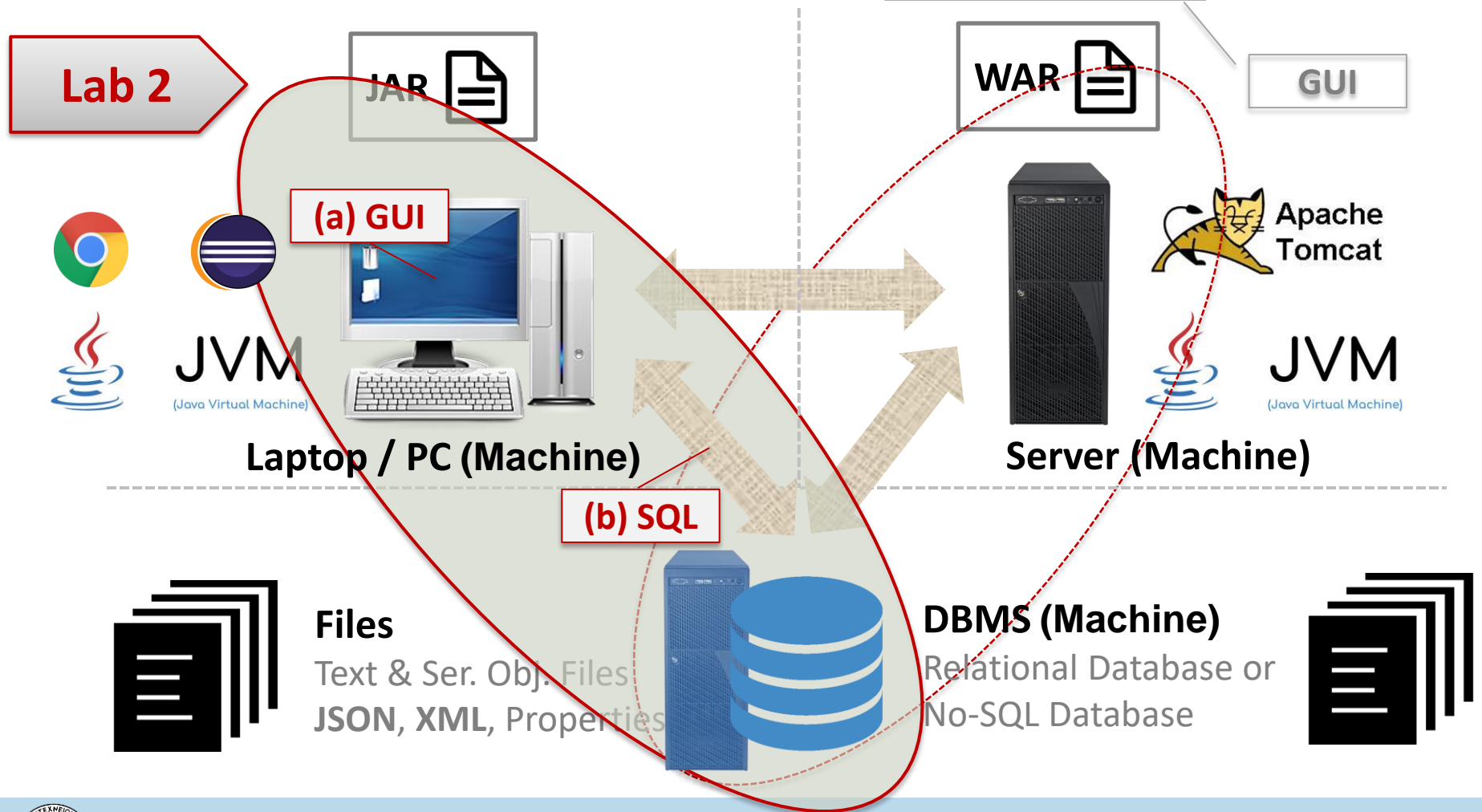
Desktop App. VS Dynamic Web App.

Desktop Applications

Java, Swing (GUI), Text Files, XML, JSON, DB

Dynamic Web App., Web Services

JSP, JavaScript, HTML, CSS, DB, XML, JSON



Development & Testing Tools

Desktop Applications

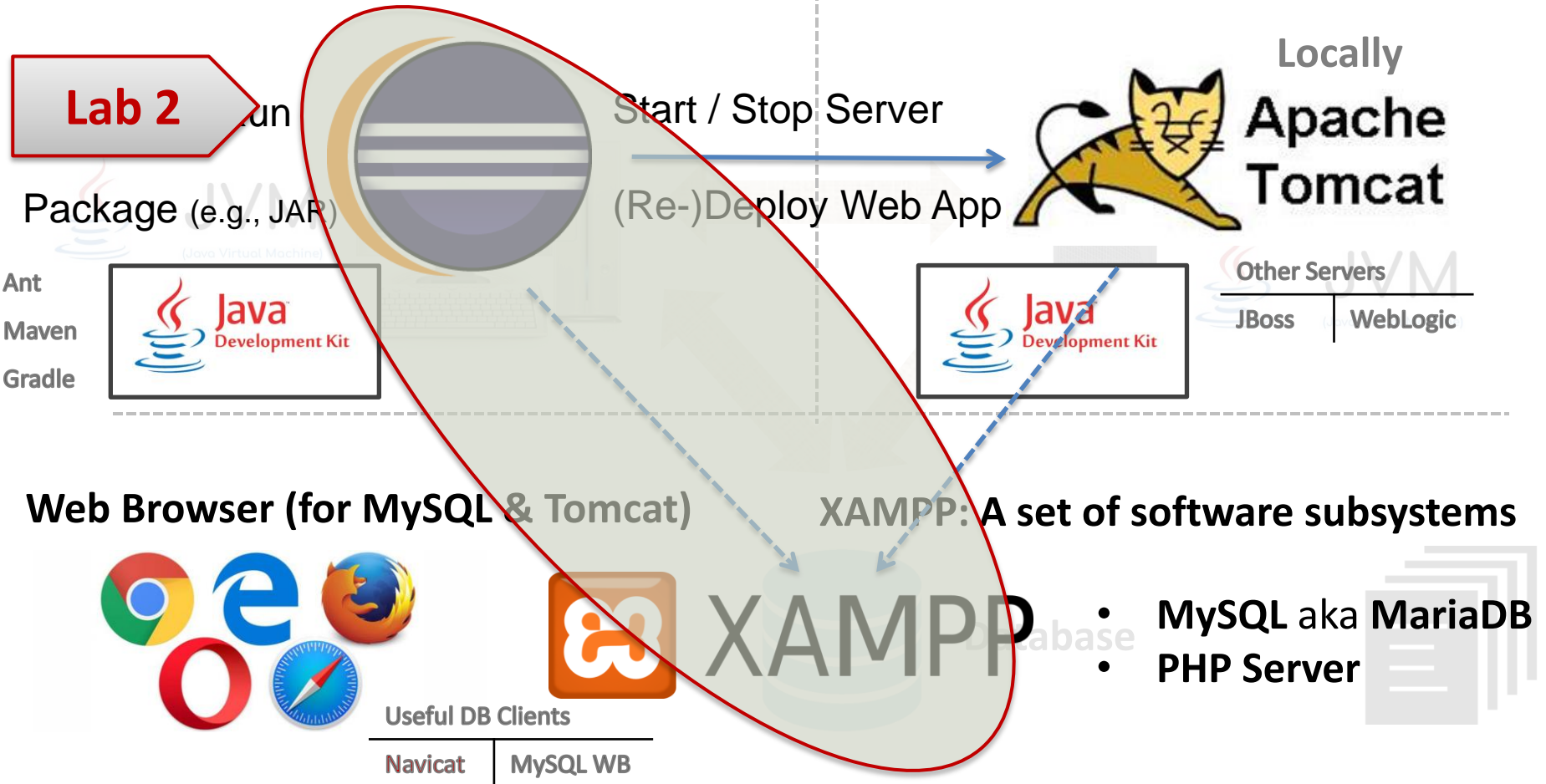
Java, Swing (GUI), Text Files, XML, JSON, DB

Eclipse IDE for Java EE Developers

Dynamic Web App., Web Services

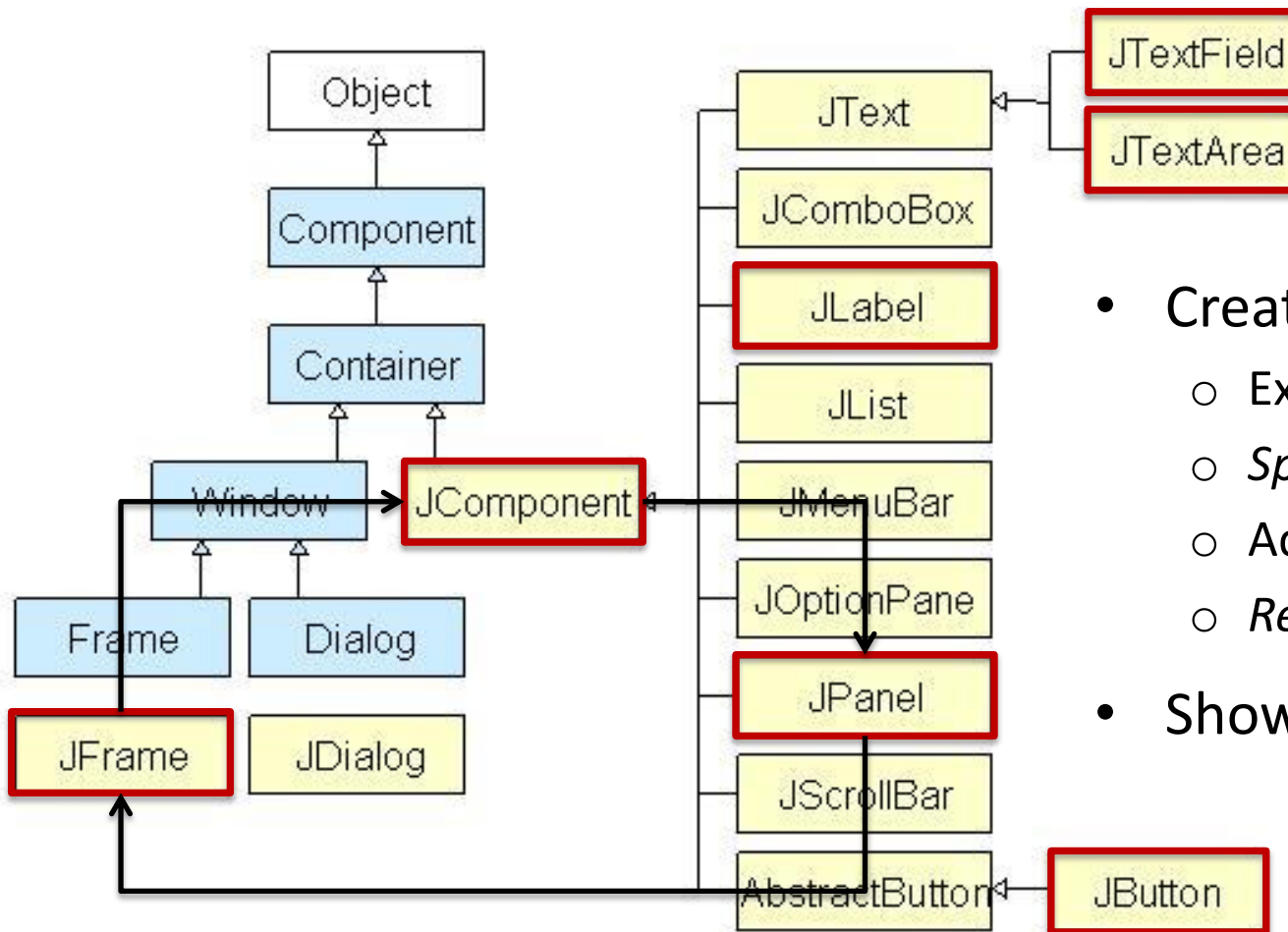
Java, JavaScript, HTML, CSS, DB, XML, JSON

Apache Tomcat Web App. Server



Desktop App - Graphical User Interface (GUI)

- Java Classes Hierarchy



- Create GUI
 - Extend **JFrame**
 - *Specify Layout*
 - Add **JComponents**
 - *Register Action Listeners*
- Show GUI



Relational Database – SQL Query Execution

Register driver

→ Get connection

Create statement

Execute query

→ Close connection



- Java Classes/Interfaces

- Connection
- Statement
- PreparedStatement
- ResultSet

- SQL Query Execution

- Create Statement
 - Get ResultSet
 - Close ResultSet

→ Close Statement



Διαδίκτυο και Εφαρμογές

[Login](#)

Διαδίκτυο και Εφαρμογές

- Γενικά Στοιχεία
- Ανακοινώσεις
- Εκπαιδευτικό Υλικό
- Ασκήσεις- Εργασίες
- Σύνδεσμοι
- Διαλέξεις & Εργαστήρια

Ασκήσεις - Εργασίες



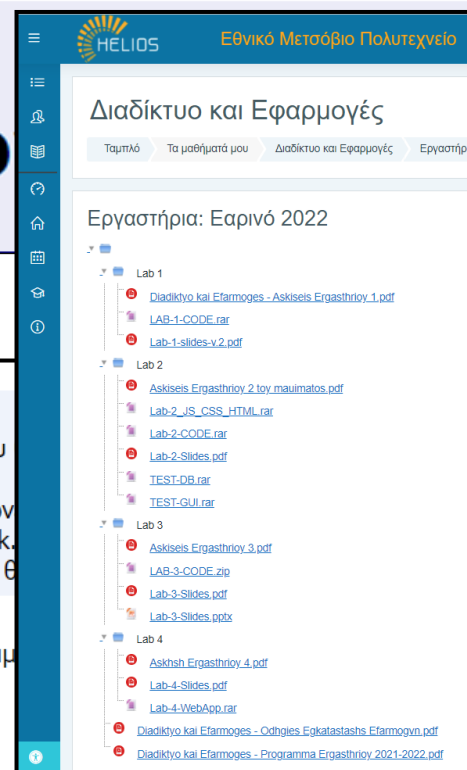
Πρόγραμμα Εργαστηρίου

Στο αρχείο αυτό υπάρχει μια σύντομη περιγραφή των εφαρμογών που καθώς και τις τεχνολογίες που θα χρησιμοποιήσουμε. Η ακριβή ημερομηνία που θα γίνεται το κάθε εργαστήριο θα ανακοινώνεται με τη διάρκεια των διαλέξεων καθώς και στο site του μαθήματος και το slack. Το Webex link που θα χρησιμοποιηθεί για τις εργαστηριακές ασκήσεις θα ανακοινωθεί.



Οδηγίες Εγκατάστασης Εφαρμογών

Στο αρχείο αυτό θα βρείτε οδηγίες για την εγκατάσταση των προγραμμάτων που θα χρησιμοποιούνται στο εργαστήριο για την ανάπτυξη των εφαρμογών.



Import Eclipse Project Instructions

Desktop App
Simple GUI

Lab 1

Εκφώνηση και Κώδικας Εφαρμογών 1ου εργαστηρίου



Ασκήσεις Εργαστηρίου 1



[LAB-1-CODE](#)



[Lab-1-Slides](#)

Create DB & Import Tables/Data Instructions

DB Connection from Java Program

Lab 2



[Test Desktop App GUI](#)



[Test DB](#)



SQL Query Execution from Java App.

- **XAMPP - MySQL**

- Start XAMPP **Control Panel** and **MySQL** (i.e., MariaDB) server and **Apache** (PHP) Server - Their background should become green
- Open a Web Browser and Type <http://localhost/> → Visit **phpMyAdmin**
- Create a DB either from scratch or import an existing one (SQL dump)

- **Connecting to the MySQL Relational Database**

- Create a Java Project

- Download **MySQL connector JAR** File


- Add MySQL connector JAR File to the Application **Classpath**

- Run a Java Program that successfully connects to the Relational DB

- Ensure that DB connection established and Data successfully retrieved from DB
- Release Resources (Connection/Statements/ResultSets) & Data



Lab Software

- JDK
 - **Java 1.7**
 - Annotations (@...) and Generic Types (<T>)
 - Diamond Operator (<>)
 - Lambda expression not supported !
- IDE
 - **Eclipse Kepler (2014)**
 - Can support development using Java 1.7 (?) or earlier
 - Can be linked with an Apache Tomcat v.7 or earlier
- DBMS
 - **XAMPP version 2.5** `Class.forName("com.mysql.jdbc.Driver");`
 - DB Connector: MySQL Connector – Java-5-0.7 

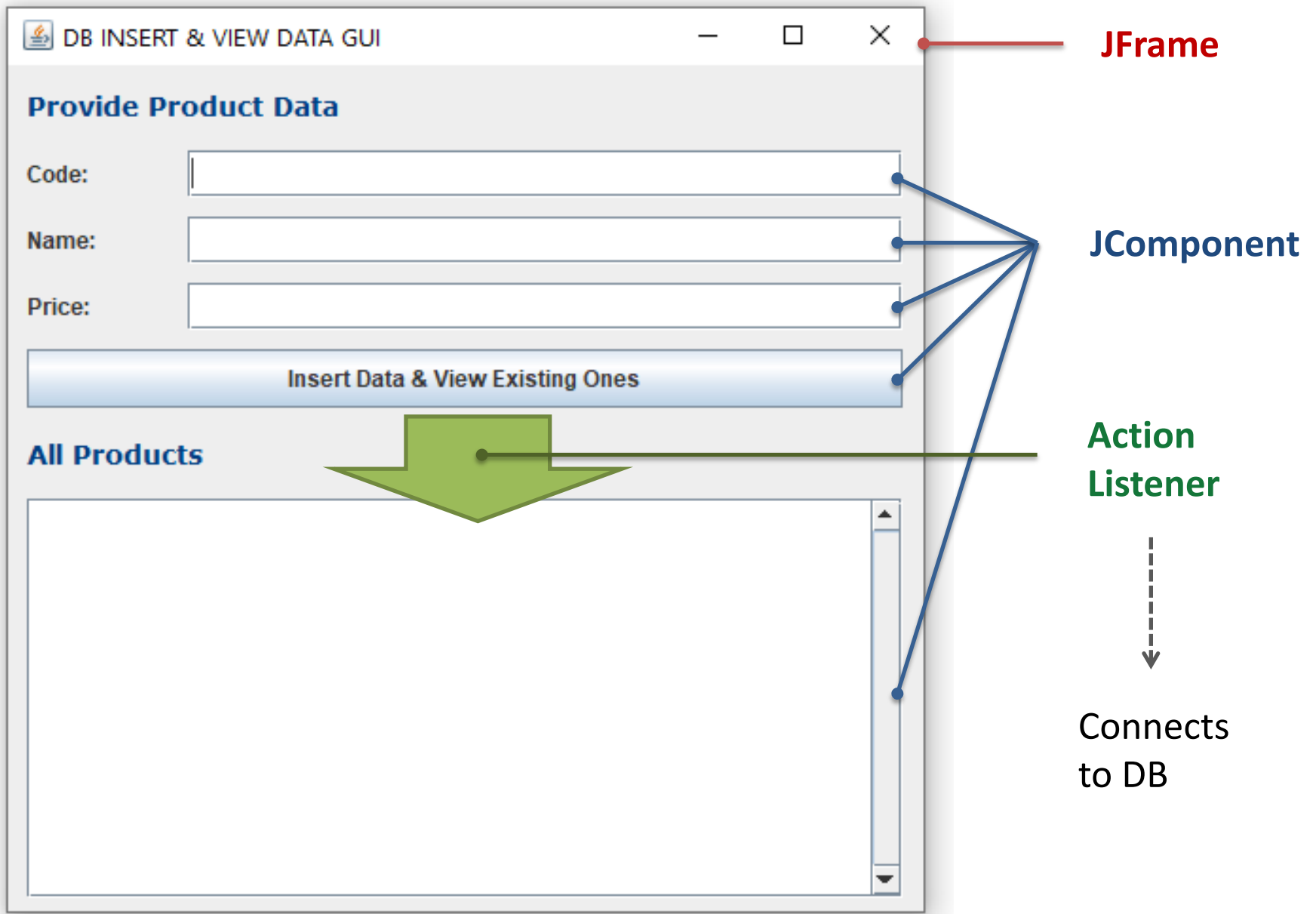
https://osdn.net/projects/sfnet_associationview/downloads/OldFiles/mysql-connector-java-5.0.7-bin.jar/
- Web
 - **Apache Tomcat version 7**
- App Server
 - Download Apache Tomcat **version 7** (from Archives)
 - Servlet definition using Annotations



Άσκηση Εργαστηρίου 1/2

- **Δημιουργία Γραφικού Περιβάλλοντος (Μέρος Α)** μιας Desktop Εφαρμογής που επιτρέπει στον χρήστη να εισάγει τα δεδομένα που αφορούν ένα συγκεκριμένο προϊόν όπως είναι α) ο **κωδικός** του, β) το **όνομά** του και γ) η **τιμή** του και ακολούθως εισάγει τα δεδομένα στην βάση και εμφανίζει τα περιεχόμενα της βάσης στην οθόνη του **(Μέρος Β)**.
 - Desktop Application GUI
 - Windows & JFrames
 - JComponents and Action Listeners
 - Java Features
 - Interfaces Methods **Implementation** and Java Classes **Extension**
 - Data Processing and User Interface Update



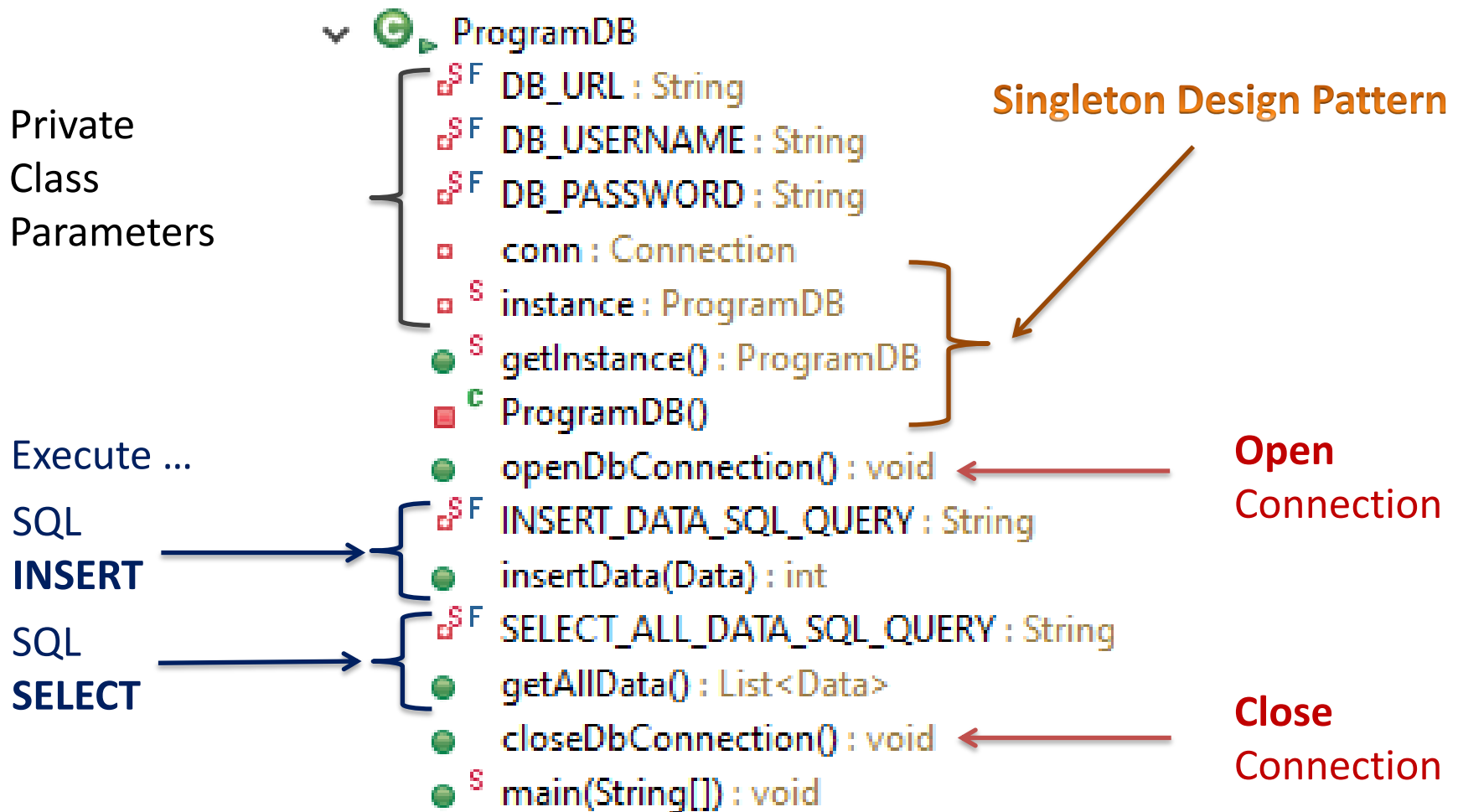


Άσκηση Εργαστηρίου 2/2

- Δημιουργία μιας Desktop Εφαρμογής (Μέρος Β) που επιτρέπει στον χρήστη να διαχειριστεί τα δεδομένα που υπάρχουν σε μια σχεσιακή βάση. Ειδικότερα, η εφαρμογή θα του επιτρέψει:
 - Να εισάγει νέα δεδομένα
 - SQL Type: INSERT
 - Να δει τα δεδομένα που υπάρχουν στην σχεσιακή βάση
 - SQL Type: SELECT
 - Να τροποποιήσει υπάρχοντα δεδομένα
 - SQL Type: UPDATE
 - Να διαγράψει υπάρχοντα δεδομένα
 - SQL Type: DELETE
- «Σύνδεση» με το γραφικό περιβάλλον (Μέρος Α)



Java Class Methods



Homework

Desktop Applications



Java Memory – Exceptions – GUI

- Δημιουργία μιας εφαρμογής που εμφανίζει την **κατάσταση της μνήμης** πριν και μετά την εκτέλεση του ακόλουθου προγράμματος/μεθόδου που αποθηκεύει 10 εκατομμύρια διαφορετικές συμβολοακολουθίες (π.χ. String-1) σε μια λίστα.
 - Να εμφανίζει την Κατάσταση Μνήμης (Max, Current, Used) πριν και μετά το «κάλεσμα» του Garbage Collector σε διάφορα σημεία του προγράμματός σας όπως στην αρχή, στο τέλος πριν και αφού ελευθερώσετε τα resources.
- Δημιουργία μιας Desktop Εφαρμογής που επιστρέφει όλη την πληροφορία που υπάρχει σε ένα **Exception / Throwable** (συμπεριλαμβανομένου των Caused By ..) με την μορφή ενός String.
- **Δημιουργία Γραφικού Περιβάλλοντος** για την λήψη και εκτέλεση εντολών.
 - Για διευκόλυνση του χρήστη η εφαρμογή θα πρέπει να παρουσιάζει τις διαθέσιμες/υποστηριζόμενες εφαρμογές (dropdown list) ή να βοηθάει τον χρήστη κατά την πληκτρολόγηση (auto-complete functionality).
 - Ιδανικά, θα έπρεπε ο χρήστης να μπορεί να ανοίγει/κλείνει αυτό το παράθυρο με το πάτημα ενός (ή συνδυασμού) πλήκτρων.



Java Reflection & Object-Relational Mapping

- Δημιουργία μεθόδου που λαμβάνει ως input α) ένα **SELECT SQL QUERY** και β) μια **Java Class** και αναλαμβάνει να «εκτελέσει» το SQL query και να βάλει τα αποτελέσματα σε μια **List** από αντικείμενα της δοθείσας κλάσης.

```
public static <T> List<T> executeSqlQuery(Connection conn, String sqlQuery, Class<T> cls) {  
  
    final List<T> resList = new ArrayList<T>();  
  
    // TODO: +++  
  
    return resList;  
  
}
```

- Παραδοχές
 - Τα **ονόματα των πεδίων/παραμέτρων** της Java Classes είναι ακριβώς ίδια με αυτά των πεδίων του πίνακα της σχεσιακής βάσης (για λόγους απλότητας, θεωρήστε ότι το SELECT SQL query είναι της μορφής: SELECT * FROM TABLE)
 - Εναλλακτικά θα μπορούσατε να χρησιμοποιήσετε **Annotations**
- Σκεφτείτε τι αλλαγές θα χρειαζόταν να γίνουν έτσι ώστε αντί να επιστρέφει μια List<T> να επιστρέφει έναν Iterator<T>.



Questions

**THANK
YOU!**

- **Site:** <http://ecourses.dbnet.ntua.gr/15373.html>
- **Email:** dkmsgroup@gmail.com
- **Slack:** <https://join.slack.com/t/internetappli-qob4034/signup>
 - Register using your email at mail.ntua.gr

