LAB 2

STRUCTURE DIAGRAM: READING, WRITING AND DEBUGGING

LAKSHYA TANDON

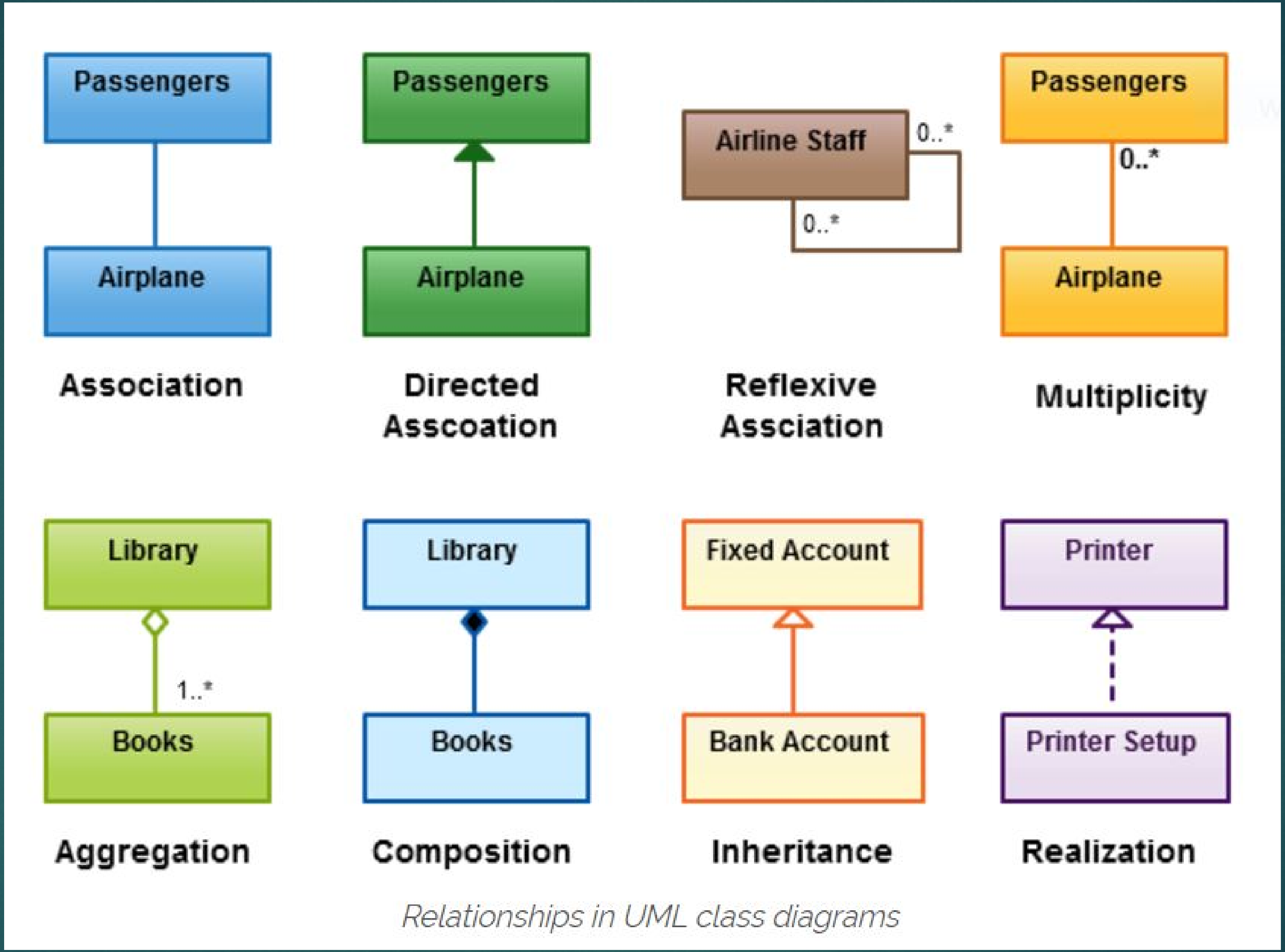
Structure Diagrams

- Shows the breakdown of system to its lowest manageable levels.
- Used to represent classes, relationship between types.
- The components are arranged in form of a tree.

Relationships presented by these diagrams

- **Class Diagram**
  - **Inheritance**
  - **Dependency** – some set of UML elements are dependent in other UML Elements. Also called as supplier client relationship.
  - **Association** – Defines dependency but a much stronger dependency.
  - **Aggregation** - it is an association that represents the part whole or part of relationship. Represented by "has a" relationship. Eg: Professor has a class to teach.
  - **Composition –** used when representing real world whole part relationship. Eg: Engine is a part of car.
  - **Multiplicity** – Many to one or many to may relation.
  - **Reflexive** – Relation into itself

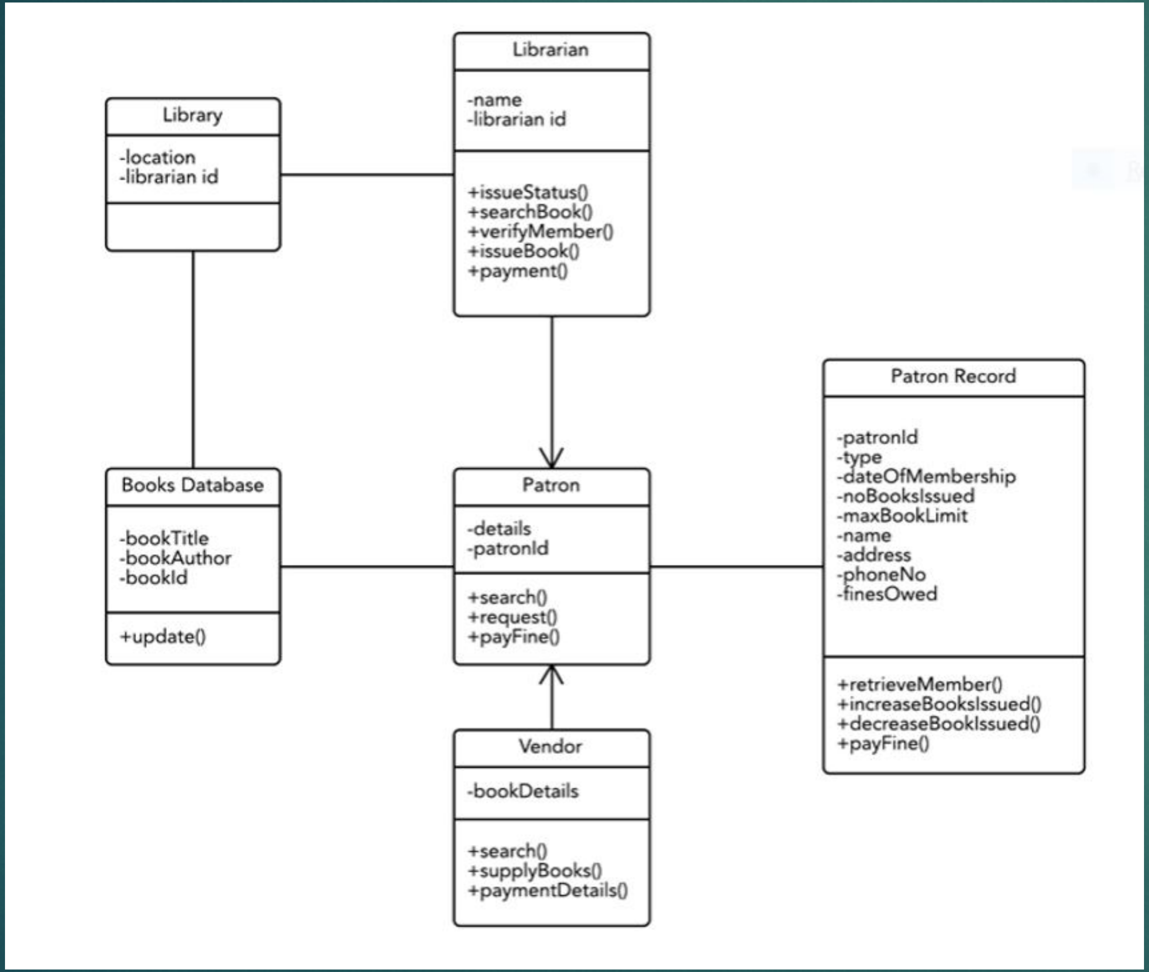
  - These diagrams are represented using Unified Modeling Language.



Relationships in UML class diagrams

Source: http://creately.com/blog/diagrams/class-diagram-relationships

Example of a class diagram

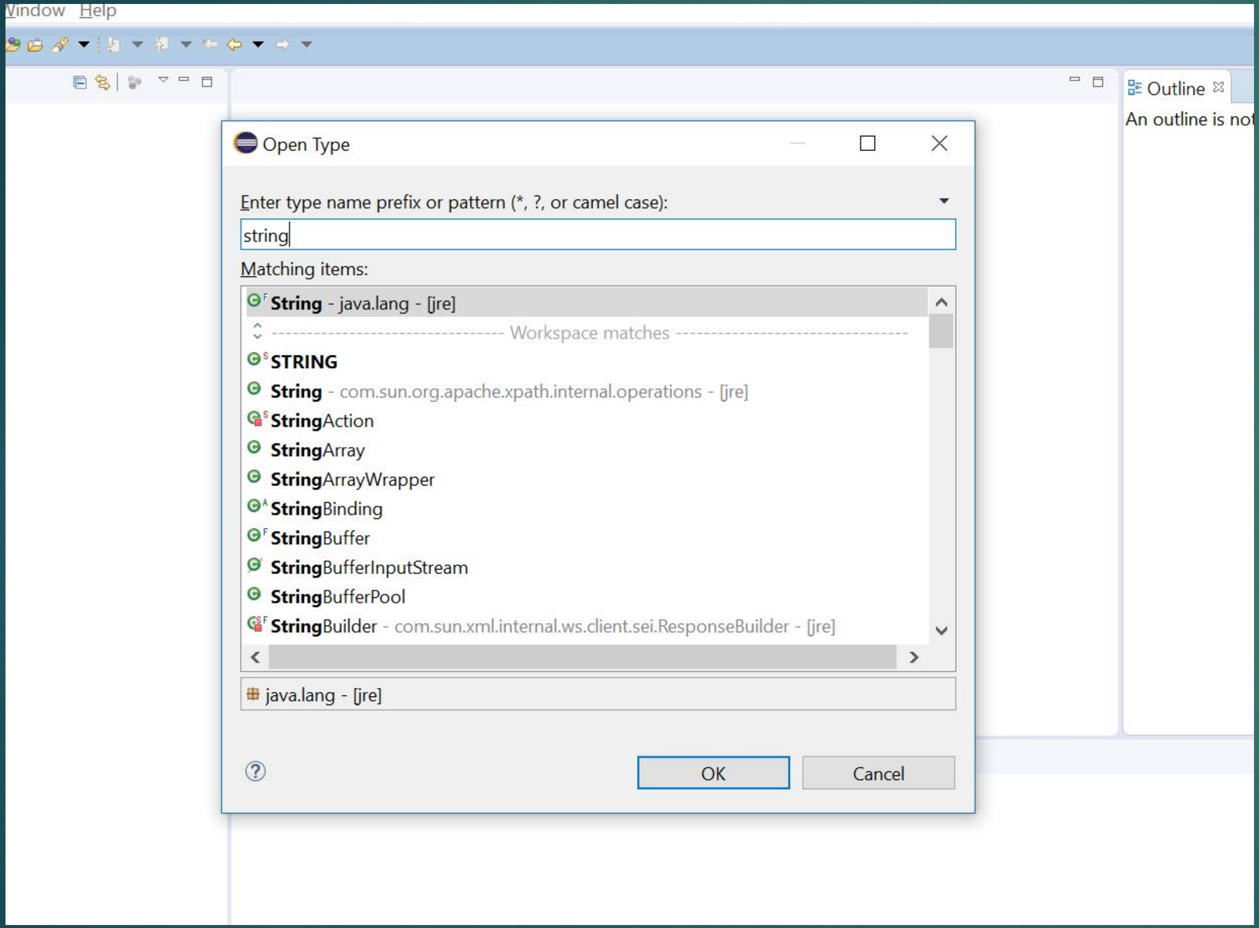


Source: https://www.lucidchart.com/pages/class-diagram-for-library-managemer

Reading code of a class file – java.lang.string

- Open eclipse
- Go to Navigate → Open task
- Or press ctrl+shift+T in windows environment

You'll see this window



Enter type name prefix or pattern (*, ?, or camel case):

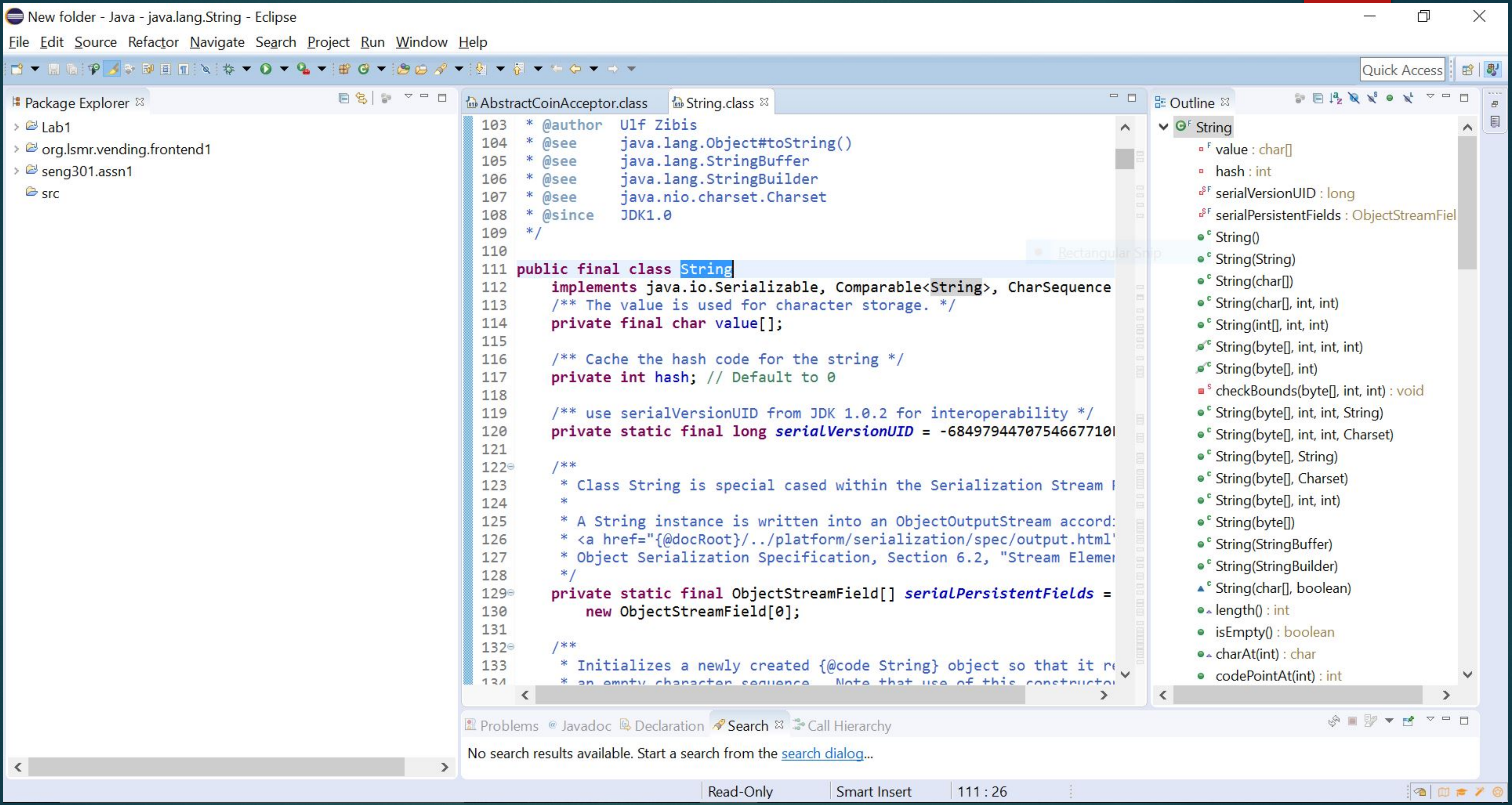
string

Matching items:

String - java.lang - [jre]
---------------------------- Workspace matches ----------------------------
STRING
String - com.sun.org.apache.xpath.internal.operations - [jre]
StringAction
StringArray
StringArrayWrapper
StringBinding
StringBuffer
StringBufferInputStream
StringBufferPool
StringBuilder - com.sun.xml.internal.ws.client.sei.ResponseBuilder - [jre]

java.lang - [jre]

Click OK


New folder - Java - java.lang.String - Eclipse
File Edit Source Refactor Navigate Search Project Run Window Help
Quick Access
Package Explorer
Lab1
org.lsmr.vending.frontend1
seng301.assn1
src
AbstractCoinAcceptor.class
String.class
Outline
Problems
Javadoc
Declaration
Search
Call Hierarchy
Read-Only
Smart Insert
111 : 26

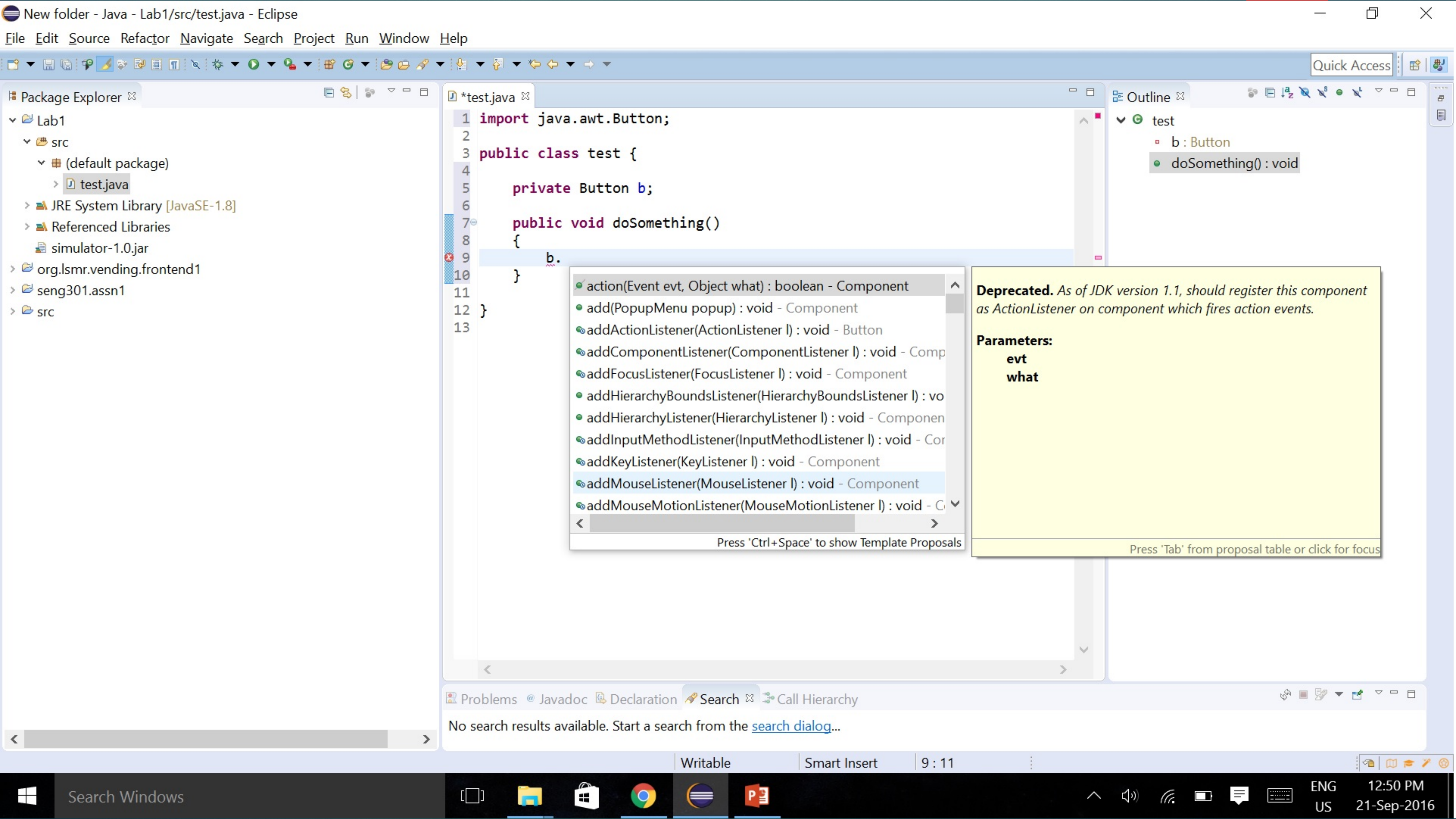
```
 * @author  Ulf Zibis
 * @see     java.lang.Object#toString()
 * @see     java.lang.StringBuffer
 * @see     java.lang.StringBuilder
 * @see     java.nio.charset.Charset
 * @since   JDK1.0
 */

public final class String
    implements java.io.Serializable, Comparable<String>, CharSequence
    /** The value is used for character storage. */
    private final char value[];

    /** Cache the hash code for the string */
    private int hash; // Default to 0

    /** use serialVersionUID from JDK 1.0.2 for interoperability */
    private static final long serialVersionUID = -6849794470754667710

    /**
     * Class String is special cased within the Serialization Stream
     *
     * A String instance is written into an ObjectOutputStream accord
     * <a href="{@docRoot}/../platform/serialization/spec/output.html"
     * Object Serialization Specification, Section 6.2, "Stream Eleme
     */
    private static final ObjectStreamField[] serialPersistentFields =
        new ObjectStreamField[0];

    /**
     * Initializes a newly created {@code String} object so that it r
```

String
- value : char[]
- hash : int
- serialVersionUID : long
- serialPersistentFields : ObjectStreamFiel
- String()
- String(String)
- String(char[])
- String(char[], int, int)
- String(int[], int, int)
- String(byte[], int, int, int)
- String(byte[], int)
- checkBounds(byte[], int, int) : void
- String(byte[], int, int, String)
- String(byte[], int, int, Charset)
- String(byte[], String)
- String(byte[], Charset)
- String(byte[], int, int)
- String(byte[])
- String(StringBuffer)
- String(StringBuilder)
- String(char[], boolean)
- length() : int
- isEmpty() : boolean
- charAt(int) : char
- codePointAt(int) : int

No search results available. Start a search from the search dialog...

Writing Code – Auto Completion

- Ctrl + Space in Windows environment.
- Command + Space in Mac environment.



```
import java.awt.Button;

public class test {

    private Button b;

    public void doSomething()
    {
        b.
    }

}
```

action(Event evt, Object what) : boolean - Component
add(PopupMenu popup) : void - Component
addActionListener(ActionListener l) : void - Button
addComponentListener(ComponentListener l) : void - Comp
addFocusListener(FocusListener l) : void - Component
addHierarchyBoundsListener(HierarchyBoundsListener l) : vo
addHierarchyListener(HierarchyListener l) : void - Componen
addInputMethodListener(InputMethodListener l) : void - Cor
addKeyListener(KeyListener l) : void - Component
addMouseListener(MouseListener l) : void - Component
addMouseMotionListener(MouseMotionListener l) : void - C

Press 'Ctrl+Space' to show Template Proposals

Deprecated. *As of JDK version 1.1, should register this component as ActionListener on component which fires action events.*

Parameters:
evt
what

Press 'Tab' from proposal table or click for focus

No search results available. Start a search from the search dialog...

You can continue with your Lab exercise