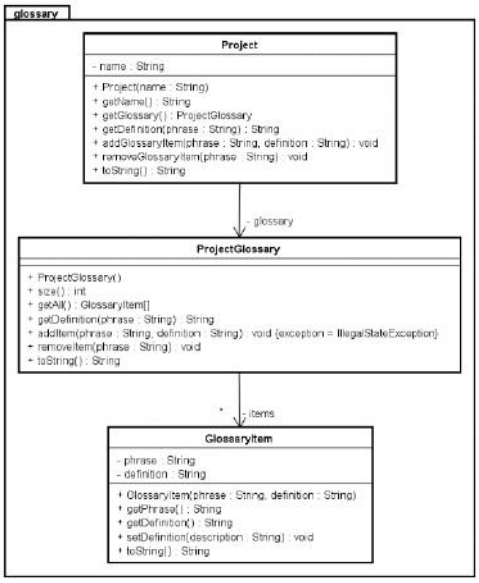
Exercises: Project Glossary



The system presented in the UML class diagram (and as source code at the end of this document) represent an IT project with a project glossary.

A Project has a name and a glossary where the glossary contains a list of items with a word or phrase and the corresponding definition, e.g.

Phrase: "User"

Definition: "*End user in form of a doctor or a nurse.*"

Example use (main method)

```
Project project1 = new Project("Project 1");
project1.addGlossaryItem("Client", "The client part of "
                      + a "client/server application.");
project1.addGlossaryItem("User", "End user in form of a doctor or a nurse.");
project1.addGlossaryItem("Account", "A location on the server application storing username, "
                                  + "password and phone number.");
System.out.println("Project 1: Definition for Client: " + project1.getDefinition("Client"));
System.out.println(project1);


/* OUTPUT:
Project 1: Definition for Client: The client part of a client/server application.
Project: Project 1
- Client: "The client part of a client/server application."
- User: "End user in form of a doctor or a nurse."
- Account: "A location on the server application storing username, password and phone number."
*/
```

Exercise: Project Glossary (Singleton)

Your exercise is to convert the class `ProjectGlossary` into a Singleton (such that you have global access to a project glossary and each project created now shares the same project glossary).

- Implement the Singleton version of the system
- Draw a class digram for your solution

Exercise: Project Glossary (Multiton)

Your exercise is now to convert the class `ProjectGlossary` into a Multiton with a String key representing the language, e.g. “uk” for a project using an english project glossary and “dk” for a project using a danish project glossary.

- Implement the Multiton version of the system
- Draw a class digram for your solution

Source code for class `GlossaryItem`:

```
package glossary;

public class GlossaryItem
{
  private String phrase;
  private String definition;

  public GlossaryItem(String phrase, String definition)
  {
    if (phrase == null || phrase.isEmpty())
    {
      throw new IllegalArgumentException("Empty phrase");
    }
    if (definition == null || definition.isEmpty())
    {
      definition = "[No definition]";
    }
    this.phrase = phrase;
    this.definition = definition;
  }

  public String getPhrase()
  {
    return phrase;
  }

  public String getDefinition()
  {
    return definition;
  }

  public void setDefinition(String description)
  {
    this.definition = description;
  }

  @Override public String toString()
  {
    return String.format("%s: \"%s\"", phrase, definition);
  }
}
```

Source code for class `ProjectGlossary`:

```
package glossary;

import java.util.ArrayList;
import java.util.List;

public class ProjectGlossary
{
  private List<GlossaryItem> items;

  public ProjectGlossary()
  {
    this.items = new ArrayList<>();
  }

  public int size()
  {
    return items.size();
  }

  public GlossaryItem[] getAll()
  {
    GlossaryItem[] array = new GlossaryItem[items.size()];
    return items.toArray(array);
  }
```

```
  public String getDefinition(String phrase)
  {
    for (GlossaryItem item : items)
    {
      if (item.getPhrase().equalsIgnoreCase(phrase))
      {
        return item.getDefinition();
      }
    }
    return null;
  }

  public void addItem(String phrase, String definition)
  {
    if (getDefinition(phrase) != null)
    {
      throw new IllegalStateException(
          "Glossary phrase already exist: " + phrase);
    }
    items.add(new GlossaryItem(phrase, definition));
  }

  public void removeItem(String phrase)
  {
    items.remove(new GlossaryItem(phrase, getDefinition(phrase)));
  }

  public String toString()
  {
    String s = "";
    if (items.size() == 0)
    {
      s += "[Empty]";
    }
    for (int i = 0; i < items.size(); i++)
    {
      s += "- " + items.get(i);
      if (i < items.size() - 1)
      {
        s += "\n";
      }
    }
    return s;
  }
}
```

Source code for class `Project`:

```
package glossary;

public class Project
{
  private String name;
  private ProjectGlossary glossary;

  public Project(String name)
  {
    this.name = name;
    this.glossary = new ProjectGlossary();
  }

  public String getName()
  {
    return name;
  }

  public ProjectGlossary getGlossary()
  {
    return glossary;
```

```
  }

  public String getDefinition(String phrase)
  {
    return glossary.getDefinition(phrase);
  }

  public void addGlossaryItem(String phrase, String definition)
  {
    glossary.addItem(phrase, definition);
  }

  public void removeGlossaryItem(String phrase)
  {
    glossary.removeItem(phrase);
  }

  @Override public String toString()
  {
    String s = "Project: " + name;
    if (glossary.size() > 0)
    {
      s += "\n" + glossary;
    }
    else
    {
      s += " [No glossary]";
    }
    return s;
  }
}
```

A test program:

```
import glossary.GlossaryItem;
import glossary.Project;

public class MainForProjectGlossary
{
  public static void main(String[] args)
  {
    Project project1 = new Project("Project 1");

    project1.addGlossaryItem("Client", "The client part of a client/server "
                                     + "application.");
    project1.addGlossaryItem("User", "End user in form of a doctor or a nurse.");
    project1.addGlossaryItem("Account", "A location on the server application "
              + "storing username, password and phone number.");

    System.out.println("Project 1: Client: " + project1.getDefinition("Client"));
    System.out.println(project1);

    // Danish:
    Project project2 = new Project("Project 2");

    try
    {
      project2.addGlossaryItem("Client",
          "Det program der som en del af en Client/Server applikation bliver "
        + "installeret på computere til læger og sygeplejesker.");
    }
    catch (IllegalStateException e) // Using the same phrase as in Project 1
    {
      System.out.println("Error: " + e.getMessage());
    }
    project2.addGlossaryItem("Bruger", "Bruger af systemet - her en læge "
                           + "eller sygeplejeske.");
    project2.addGlossaryItem("Konto", "Et sted på en server med oplysninger "
                           + "om brugernavn, kodeord og telefonnummer.");
```

```
    System.out.println("Project 2: Client: " + project2.getDefinition("Client"));
    System.out.println(project2);


    // A new project with the project glossary as in project 1
    Project project3 = new Project("Project 3");
    GlossaryItem[] glossaryItems = project1.getGlossary().getAll();
    for (int i=0; i<glossaryItems.length; i++)
    {
      project3.addGlossaryItem(glossaryItems[i].getPhrase(),
                               glossaryItems[i].getDefinition());
    }
    System.out.println(project3);

  }
}
```