Homework 3

Due Oct 14, 2020

20 pts each

What should be turned in?

* This word doc
* Homework3 project in zip

# Problem 1: Logger

Scenario 1: You are given some legacy Java code of a Tank game. Class Tank implements Movable interface. Now we want to log the execution time of invoking move() method to the console.

Code your solution here:

|  |
| --- |
| public class Tank implements Movable {  @Override  public void move() {  long startTime = System.*currentTimeMillis*();  System.*out*.println("Tank moving...");  // code to invoke some fancy graphics interface, it may be time consuming  try {  Thread.*sleep*(new Random().nextInt(10000)); // simulating the executing of some time consuming graphics interface  } catch (InterruptedException e) {  e.printStackTrace();  }  long endTime = System.*currentTimeMillis*();  System.*out*.println("Move took" + (endTime - startTime) + " milliseconds");  } } |

Scenario 2: what if you don’t have access to the source code of Tank class, in other words, this Tank class is from a JAR? In this case, you still can use it (e.g. instantiate this class using keyword new and invoke its methods like tank.move()), but you are not able to modify the source code like what you did in scenario 1. A good way to think about this is: can you modify the code in ArrayList<T> class? No, you can only use it, since it is part of JDK. Feel free to create other class under scenario2 package.

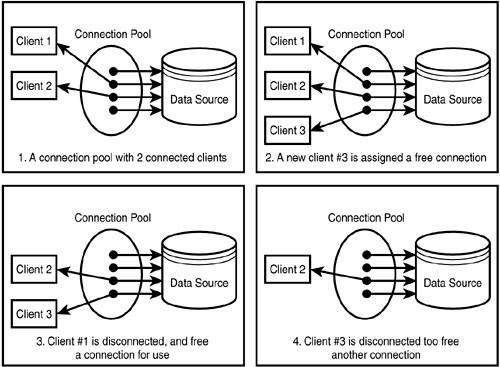
But we still want to log the execution time of invoking move() method to the console.

Code your solution here:

|  |
| --- |
| public class TestTank {  public static void main(String[] args) {  // record move() execution time of a tank  Tank tank = new Tank();  long startTime = System.*currentTimeMillis*();  tank.move();  long endTime = System.*currentTimeMillis*();  System.*out*.println("Move took " + (endTime - startTime) + " milliseconds");  } } |

# Problem 2: Database Connection Pool

As you may know, creating database connections is an expensive operation. As a result, a DB connection pool is used as a cache of database connections maintained so that the connections can be reused when future requests to the database are required. Connection pools are used to enhance the performance of executing commands on a database. For more information about connection pool, please read: <https://stackoverflow.com/questions/4041114/what-is-database-pooling>



Some popular Java connection pooling frameworks are HikariCP, Tomcat JDBC CP, Vibur, Apache Commons DBCP2, C3PO etc…

In this problem, you are asked to implement your own connection pool (you only need to complete DBPool class, don’t change anything in DBPoolTest and BingConnectionImpl).

The connection pool will

* Instantiate some BingConnectionImpl instances based on the initialSize provided by the client program.
* Store the created Connection instances for future use.
* Provide a connection to the client program when asked.
* Return a connection back to the pool when client program is done using this connection.

# Problem3: Health Insurance

Assume you are working on an insurance company’s system. The business flow of handling a claim is: 1. validate the claim; 2. Based on the result, either approve or reject. For now, the insurance company only handles healthy related insurances. Please read the code I provided.

Both the HealthInsuranceSurveyor and ClaimApprovalManager classes work fine and the design for the insurance system appears perfect until a new requirement to process vehicle related insurance claims arises. The company just extend its territory to vehicle insurance. Yeah!

Please extend the code to accommodate this change. Feel free to do whatever you see fit.

# Problem4: Meeting Scheduling

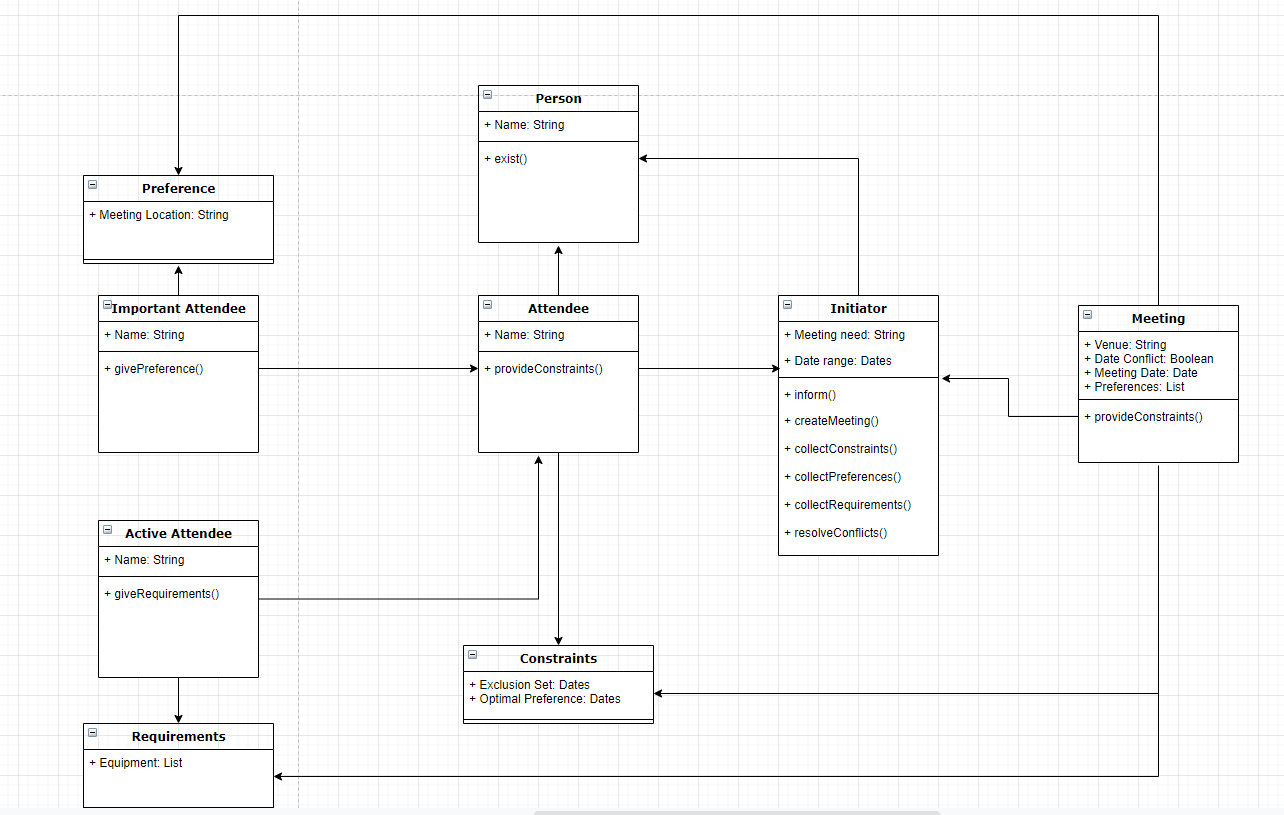
Assume you work for Wonderland Software Services. The marketing department has identified a large potential market for meeting-scheduling software that would exploit Internet-based communication technologies. Scheduling meetings with busy people is generally a nightmare. It is hard to find a date and a place that suit everyone’s constraints; meeting organizers need to pester people to get their availability; other people are unnecessarily inconvenienced by messages that do not concern them; when the meeting is scheduled some constraints have changed in the meantime; new scheduling cycle need to be repeated when no date/location is found in a reasonably short period, and so forth. As a result, meetings tend to be organized poorly and late; important people sometimes do not show up; and there is a significant, unnecessary overhead in the scheduling process.

Meetings are typically scheduled as follows. A meeting initiator informs potential participants about the need for a meeting and specifies a date range within which the meeting should take place, asking people to return their availability constraints within that time interval. Constraints are typically expressed as two sets: an exclusion set specifying dates within the date range when the participant could not attend, and an optional preference set specifying dates within the date range on which the participant would prefer the meeting to take place (a date may refer to a full day or a period in a day). In some cases, the initiator may also ask participants who will play an active role in the meeting for specific requirements regarding the meeting room (e.g. project, laptop, network connection, videoconferencing facilities etc). “Important” participants may optionally be asked to state preferences for meeting locations.

The scheduled meeting date should belong to the stated date range and to none of the exclusion sets; it should ideally belong to as many preference sets as possible. The meeting venue should ideally fit the preferences of important participants. A date conflict occurs when no date can be found outside all exclusion sets. A room conflict occurs when no room can be found, at any date outside all exclusion sets, which meets the room requirements. Conflicts can be resolved in several ways: the initiator may extend the date range, some participant may remove dates from their exclusion set, or some participants my decline the invitation to attend. A new scheduling cycle may thus be required in case of conflict.

Please identify all important class and draw them in a UML class diagram. Here is a link to how to draw UML class diagram. <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-class-diagram/>

Copy your diagram here (you can use draw.io to draw), no need to write any code for this problem.



# Problem5: XML parser and Java Reflection APIs

In this problem, we write the definitions of two objects in XML format. See BeanDefinitions.xml under src directory.

|  |
| --- |
| <beans>  <bean id="harry" class="problem5.domain.Soldier">  <property name="name" value="Harry Potter"></property>  <property name="rank" value="Sergeant"></property>  <property name="age" value="35"></property>  </bean>  <bean id="m1" class="problem5.domain.Tank">  <property name="model" value="M1 Abrams"></property>  <property name="speed" value="45"></property>  </bean>  </beans> |

The first object (aka Java bean) is an instance of Soldier whose name is Harry Potter, rank is Sergeant and age is 35. The second object is a M1 tank with speed 45 miles per hour. Your goal is to write code to parse this XML file and then instantiate the two objects using Java reflection APIs. Some classes have already been written for you. Please complete XmlBeanFactory class. To test your results, run the main method in Main class. (Please only write code in XmlBeanFactory, leave other class unmodified.)

You will find some useful tips on XML parsing here: <https://dom4j.github.io/>

The output shall look like this:

|  |
| --- |
| Operator{name='Harry Potter', rank='Sergeant', age=35}  Tank{model='M1 Abrams', speed=45, driver=null} |