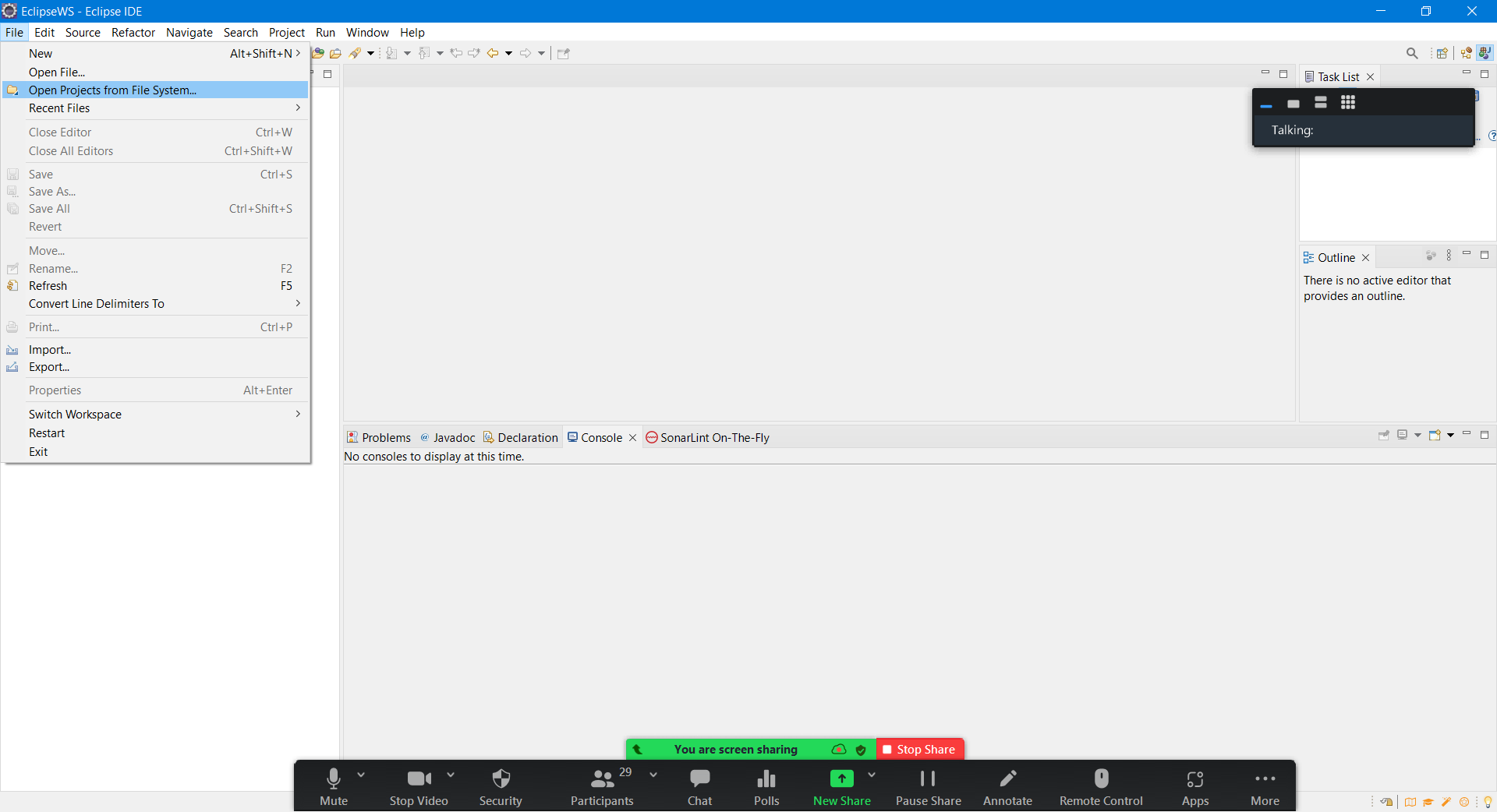
Foundation Project Skeleon code & requirements

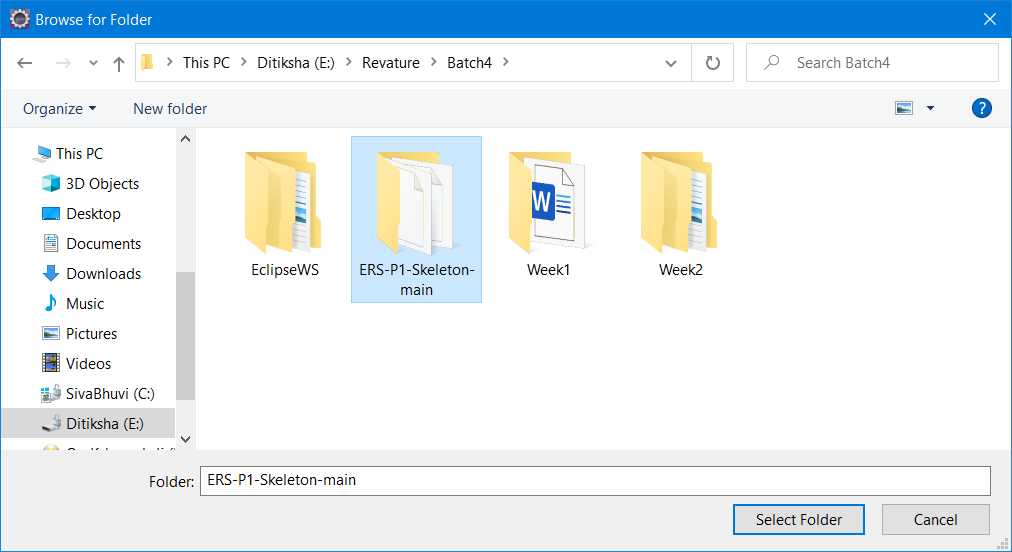


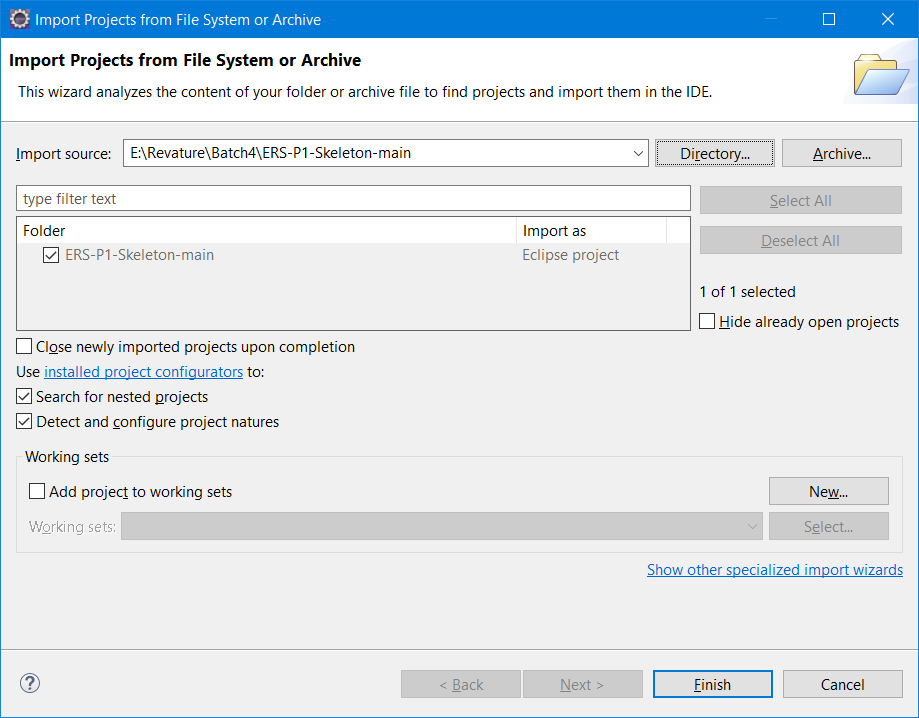
Foundation Project github repo url : <https://github.com/revature-coe/ERS-P1-Skeleton>

Importing/Opening existing project in Eclipse

* Open Eclipse
* Click File-> Open Project from File System







In this week, I will create a similar project.

Leave Request System – LRS

Project Package Structure.

Driver.java present in com.revature package is the Starter class.

Agenda

1. Annotations
2. Stack & Heap
3. Garbage collections
4. Collection API (List, Set, Queue, Map)

Revisit Day 8

1. Creating Custom Exception
2. Getting Input from User using Scanner
3. Abstract Class & Interface
4. Unit Testing using Junit
5. Git commands (init, status, log, push, pull, add, commit)

Annotation – Is nothing but MetaData (Data about data)

Real Time Example – TOC (Table of Contents) page in a book & index page in a book.

In HTML meta tags provide more information about the web page that search engines will utilize to categorize the page.

Generally meta mean Data about data.

In Java, Annotations is introduced in Java version 5 only.

Annotations will start with @ symbol.

Annotations can be added to class, methods, variables, arguments etc.,

Let’s consider JAVA Complete Reference – Is a Book about JAVA

@Test is a Junit annotation

Types of Annotation

1. Built-in Annotation /Pre-defined/System defined - @Deprecated, @SupressWarnings, @Override
2. Custom Annotation / User defined annotation [@interface keyword]
3. Meta Annotation (Annotations used for creating custom annotation) @Retention, -- where to keep the annotation, @Documented, @Target – Where the annotation can be applied

Annotations will provide more information to the JVM.

Stack & Heap are JVM memory area

Stack – is the place where method invocations & reference variable are stored

Heap – is the location in memory where java object will be created and stored

Employee emp = new Employee; (In Stack) emp will be stored,(in Heap) employee object will be stored.

If the stack is full you will get “StackOverflowError”

If the heap is full you will get “OutOfMemoryError”

<https://www.geeksforgeeks.org/stack-vs-heap-memory-allocation/>

Garbage Collection – Is a Java feature - -- Automatic Memory Management

JVM will automatically remove unused objects from heap area periodically or whenever it is necessary.

JVM will decide when to garbage collect using some internal algorithms. It’s unpredictable behavior.

Ways of requesting gc garbage collection

* System.gc()
* Runtime.getRuntime().gc()
* System.runFinalize()

Collection API – Set of classes & interfaces used for handling group of objects. This is called Collection API

Collection API Hierarchy



Iterable means traversing through each element in a group of objects.

Collection API will always works with Objects only not with primitive data types.

Collection is used

* Calculating Interest for a period of 3 months in Bank for all account holders

Collection is a Interface which extends Iterable interface.

There are 3 derived interfaces for Collection

1. List (Allow Duplicates, Maintain insertion Order, Allows null n number of time)
2. Queue (It implements FIFO data structure, Maintain insertion order FIFO = First In First Out
3. Set (Won’t allow duplicates, won’t maintain insertion order, allows null only once, if you add duplicate value it will not give any error it will store only one value not the duplicate value)
4. Collections API will work with Wrapper Classes

Primitives – Non-Objects

Wrapper Class – Objects

Boxing – Is a process of converting primitive data type to it’s corresponding object representation using wrapper class

Java will do the boxing process automatically in collection. So it’s called AutoBoxing

Unboxing – Is the reverse of boxing. Converting Wrapper class Objects to it’s corresponding primitive value.

All the Collection Interfaces & Classes are defined in java.util package.

List will allow duplicate values

List will allow null many times

List will maintain insertion order.

List is a interface, we have ArrayList & LinkedList as implementation classes

Arraylist is similar to primitive array, only difference is it’s dynamically growing nature.

Imp drawback of ArrayList – Arraylist is not suitable for insert or delete in the beginning or middle. The reason is lot of shift left or shift right operation needs to be performed which takes more time so it’s affects the performance of the application.

Arraylist is best suited for inserting at the end and searching the data using index.

<https://www.javatpoint.com/java-linkedlist>

Map is not part of Collection API directly.

Map is used to store the data in key,value pair.

Map is used to store one to one relationships

One person can have only one SSN

One Person can have only one Passport

One person can have only one heart.

Map store the data in key,value pair. In that key should be unique. Value can have duplicates

In Map null is allowed only once for Key & allowed many times for values.

Traversing through map