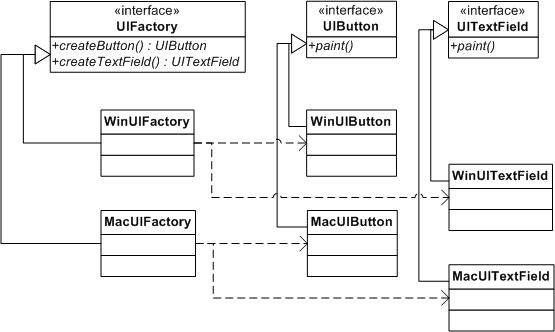
1. Write a program printing out salary and bonus of employees in Fsoft. There are 2 kinds of employees: developer and project manager. An employee has information: lastname, firstname, birthday, startWorkingDay, TaxCode, salary. Company name of employee is common and cannot be changed. The lunch bonus is 600k/month for every employee, it can be changed. A developer has a main programming language. A manager manages a list of developers. User can list person information (lastname, firstname...) of developers that are managed by a manager. A developer was paid 10% salary as the performance bonus. A manager as paid 10% salary + a number of managed employees\*100k as the performance bonus. Every month, an employee will be paid a total of: salary, lunch bonus, performance bonus.
2. Build a UI toolkit that supports multiple OSs (implementation). Motivation:
   1. Not require much change to make an application target a particular OS.
   2. Easy to add new OS support into the toolkit.
   3. Expectation: using **Abstract Factory Pattern**.



1. Build a Bank class that will notify Customer class and StakeHolder class when there are any changes in the interest rate using Observer pattern.
2. Create OracleJDBCService class with the method name “execute” which only returns “This is OracleJDBCService”.

Create MySqlJDBCService class with the method name “execute” which only returns “This is MySqlJDBCService”.

Create a Service Locator to do the following features:

* 1. Lookup the correct service by only inputting service name.
  2. Print out the result.

1. Create PersonA class
   1. Create a Counter class that will increase 1 for each time accessing.
   2. Create PersonA class which extends Thread and override run() method to access the counter and increase 1.
   3. Create PersonB class which implements Runnable and override run() method to access the counter and increase 1.
   4. Create a main class with the following features:
      1. Access the Counter class as singleton pattern.
      2. Start PersonA and PersonB thread and loop them 100 times.
      3. Print out the counter at the end of threads.
      4. What will happen if calling as following: PersonA.run()?
2. Create Person Class with the following features:
   1. Properties: id, name, address, age
   2. Override equals/hashcode using ‘name’ property.

Create a main Program with the following features:

1. Create HashSet list to store a list of Persons (generate a list of 50 persons in the hash with random name, address, age).
2. Using both Comparator and Comparable to sort the Person list in DESC of ‘name’ property and print the name out.
3. Get the top 10 of Person list and print it out.
4. Get the top 10 of Persons who are ‘age’ > 10 and print it out.
5. Create a list of Persons using HashMap and ConcurrentHashMap.

Generate a list of 50 persons in the hash with random name, address, age.

Implement the following features:

1. Using “for each” to print out the list of persons. Add/delete one item directly to the hashmap list and concurrenthashmap list. See what will happen and report it.
2. Using Iterator to print out the list of persons. Add/delete one item directly to the hashmap list and concurrenthashmap list. See what will happen and report it.
3. Measure performance of adding/deleting/Printing out person list when using ConcurrentHashMap and Collections.synchronizedMap(hashMap).
4. Create a list of Persons using ArrayList.

Generate a list of 50 persons in the hash with random name, address, age.

Implement the following features:

1. Using Iterator and IteratorList to print out the list.
2. Using IteratorList to print out the list from the last item to fist item.
3. Add one more item into the list that using IteratorList (not using list.add(xxx), using iteratorList.add(xxx))and see what happen and report it.
4. Print out the list with ‘name’ property is sorted DESC.
5. Create a Counter class with a private method named “helloWorldReflection”. That method will only print out the message: “Welcome to reflection API”.

Create a ReflectionTests class to inspect helloWorldReflection method and execute it.