```java
/*
Noor Haj Dawood - 314997602
Bader Daka - 208219212
Demo Video - https://youtu.be/ARpuI-boCL8
 */
package il.ac.shenkar.assignments;

import java.lang.reflect.Field;
import java.lang.reflect.Method;
import java.util.LinkedList;
import java.util.List;

/**
 * Displays information about a class: Methods / Fields / ParentClass
 * Uses the Class<T> class to get the required information
 */
public class NoorHajDawoodReflectionUtilities implements IReflectionUtilities{

    /**
     * return a list which includes all the methods names in className
     * @param className
     * @return List<String>
     * @throws ClassNotFoundException
     */
    @Override
    public List<String> getNamesOfAllMethods(String className) throws ClassNotFoundException {
        // get Class reference
        Class<?> clazz = Class.forName(className);

        // Get class Methods, iterate on the results and add them to the list
        List<String> list = new LinkedList<String>();
        Method[] methods = clazz.getDeclaredMethods();
        for(Method method: methods) {
            list.add(method.getName());
        }
        return list;
    }

    /**
     * return a list which includes all the fields names in className
     * @param className
     * @return List<String>
     * @throws ClassNotFoundException
     */
    @Override
    public List<String> getNamesOfAllFields(String className) throws ClassNotFoundException{
        // get Class reference
        Class<?> clazz = Class.forName(className);

        // Get class Methods, iterate on the results and add them to the list
        List<String> list = new LinkedList<String>();
        Field[] fields = clazz.getDeclaredFields();
        for(Field filed: fields) {
            list.add(filed.getName());
        }
        return list;
    }

    /**
     * return the name of the parent class of className
     * @param className
     * @return String
     * @throws ClassNotFoundException
     */
    @Override
    public String getNameOfParentClass(String className) throws ClassNotFoundException {
        // get Class reference
        Class<?> clazz = Class.forName(className);

        // get Class and then get the Super class an
```

```
return it's name
        return clazz.getClass().getSuperclass().
getName();
    }
}

```