SRM
UNIVERSITY
(Under section 3 of UGC Act 1956)

[IRIS]

APRIL 24, 2017

Smart Test Taker

Presented by:
Alkesh Kumrawat and Aditya Pratap Singh Rajput

Overview

This mini-project in JAVA SE aims to deploy real world Test Taker named “IRIS”. It embodies various parts from registration of new students to providing their performance reports to their respective faculties. Object Oriented Programming have been augmented in the project. Moreover, it’s a future oriented project.

Interrelated classes and interface have been used. Classes for bot conversation, feedback, introduction to the system, interface. Question bank file have been incorporated as a database for the questions to ask from.

Each student attempting a test generates a test report file for the same, for further reference of respective faculties.

About Iris

Iris is a user-friendly substitute of invigilator that helps faculties in taking tests efficiently. It’s a modern approach to class test series taken by institutes. It enriches fair analysis of the students.

Uses special Unicode for making it user friendly conversation.

Object Oriented Programming concepts used are

Multilevel Inheritance-	Feedback extends Conversation, Answers extends Conversation,
Abstraction-	used appropriate methods iris() and get sign()
Encapsulation-	use of getter and setters for taking private data input
Polymorphism-	overloading setters by inherited classes

Java Concepts Used

☑ File Handling

☑ Classes and objects

☑ Exception Handling

☑ Interface

☑ String

☑Wrapper Classes

☑ Date and Time API

☑ Multithreading

Description

Directory View

com.learning.java

Bot/

Conversation (Class)

Elements (Interface)

Feedback (Class)

FeedBack.txt

Introduction (Class)

File_Handling/

Answers (Class)

Student (Class)

Question_Bank/

Question(Class)

Main (Class)

Program Flow Using Main.java class

Main Class

```
package com.learning.java;

import com.learning.java.Bot.Conversations;
import com.learning.java.Bot.Introduction;
import java.io.IOException;

public class Main {

public static void main(String[] args) {

     Introduction introduction = new Introduction();

     Conversations conversations=new Conversations();

       introduction.wishes();

      introduction.intro();

        try {
            conversations.firstConversation();
        } catch (IOException e) {
            e.printStackTrace();
        }
        conversations.instructions();
        conversations.testConversation();
        conversations.afterConversation();
    }
}
```

Description

Firstly, introduction calls wishes() method for greets then it calls intro() methods for introduction of student . In the second part conversation object gives instruction after taking info using intro method using Introductions() . Conversation object uses testconversation()

For taking test using random sets of question paper defined in the files directory. Test answers are taken from student using getter and setter. The answers are written to separate file of students using file handling.

After conversation and test is taken feedback is taken by the student and entered into the feedback text file.