```
 1 package control;
 2
 3 import adt.ArrList;
 4 import adt.DoublyLinkedList;
 5 import adt.LinkedListInterface;
 6 import adt.ListInterface;
 7 import boundary.CourseMaintenanceUI;
 8 import boundary.CourseProgramMaintenanceUI;
 9 import dao.DAO;
10 import dao.Initializer;
11 import entity.Course;
12 import entity.CourseCodeComparator;
13 import entity.CourseProgram;
14 import entity.Program;
15 import utility.Command;
16 import utility.MessageUI;
17 import utility.Search;
18
19 /**
20  *
21  * @author Chew Lip Sin
22  */
23 public class CourseProgramMaintenance {
24
25     private static final DAO<CourseProgram> cpDAO = new DAO<>();
26     private final CourseProgramMaintenanceUI coursePUI = new
CourseProgramMaintenanceUI();
27     private final Search search = new Search();
28     private final CourseMaintenanceUI courseUI = new CourseMaintenanceUI();
29     private final Initializer in = new Initializer();
30     private final CourseCodeComparator cCodeC = new CourseCodeComparator();
31
32     public void runCourseProgramMaintenance(ListInterface<Course> courseList,
ListInterface<Program> programList) {
33         LinkedListInterface<CourseProgram> courseProgramList =
cpDAO.dLLRetrieveFromFile("courseProgram.dat");
34         int choice;
35         do {
36             choice = coursePUI.getMenuChoice();
37             switch (choice) {
38                 case 0:
39                     break;
40                 case 1:
41                     addCourseProgram(courseProgramList, courseList, programList);
42                     break;
43                 case 2:
```

```
 44                      removeCourseProgram(courseProgramList, courseList, programList);
 45                      break;
 46                  default:
 47                      MessageUI.displayInvalidChoiceMessage();
 48
 49              }
 50          } while (choice != 0);
 51
 52      }
 53
 54      private void addCourseProgram(LinkedListInterface<CourseProgram> courseProgramList,
ListInterface<Course> courseList, ListInterface<Program> programList) {
 55          ArrList.insertionSort(courseList, cCodeC, "asc");
 56          int choice;
 57          do {
 58              courseProgramList = cpDAO.dLLRetrieveFromFile("courseProgram.dat");
 59              courseUI.listAllCourses(courseList);
 60              choice = coursePUI.getCourseChoices(courseList);
 61              if (choice != 0) {
 62                  Command.cls();
 63                  coursePUI.addCourseProgramUI(courseProgramList, courseList, programList,
choice);
 64              }
 65          } while (choice != 0);
 66
 67      }
 68
 69      private void removeCourseProgram(LinkedListInterface<CourseProgram>
courseProgramList, ListInterface<Course> courseList, ListInterface<Program> programList) {
 70          ArrList.insertionSort(courseList, cCodeC,"asc");
 71          int choice;
 72          do {
 73              courseProgramList = cpDAO.dLLRetrieveFromFile("courseProgram.dat");
 74              courseUI.listAllCourses(courseList);
 75              choice = coursePUI.getCourseChoices(courseList);
 76              if (choice != 0) {
 77                  Command.cls();
 78                  coursePUI.removeCourseProgramUI(courseProgramList, courseList,
programList, choice);
 79              }
 80          } while (choice != 0);
 81      }
 82
 83      public void deleteCourse(Course course, LinkedListInterface<CourseProgram>
courseProgramList) {
 84          LinkedListInterface<CourseProgram> cp1 = new DoublyLinkedList<>();
```

```
 85          for (int i = 0; i < courseProgramList.sizeOf(); i++) {
 86              if (courseProgramList.get(i).getCourseCode().equals(course.getCourseCode()))
{
 87                  cp1.add(courseProgramList.get(i));
 88              }
 89          }
 90          for (int i = 0; i < cp1.sizeOf(); i++) {
 91              courseProgramList.remove(cp1.get(i));
 92          }
 93          cpDAO.saveToFile(courseProgramList, "courseProgram.dat");
 94      }
 95
 96      public void modifyCourse(Course course, Course oldCourse,
LinkedListInterface<CourseProgram> courseProgramList) {
 97          LinkedListInterface<CourseProgram> cp1 = new DoublyLinkedList<>();
 98          for (int i = 0; i < courseProgramList.sizeOf(); i++) {
 99              if
(courseProgramList.get(i).getCourseCode().equals(oldCourse.getCourseCode())) {
100                  cp1.add(new CourseProgram(course.getCourseCode(),
courseProgramList.get(i).getProgramCode(),
101                          courseProgramList.get(i).isIsElective()));
102                  for (int j = 0; j < cp1.sizeOf(); j++) {
103                      courseProgramList.set(i, cp1.get(j));
104                  }
105              }
106          }
107
108          cpDAO.saveToFile(courseProgramList, "courseProgram.dat");
109      }
110 }
```