```java
/**
 * @author Caroline Ta
 * @since 05.19.2020
 */

// ---------------------------------------------------------------------------
// Ctrl + Shift + Alt + G = Generate all Javadoc documents
// Ctrl + Shift + Alt + Z = Remove all Javadoc documents

// Ctrl + Shift + A: "Press Alt+Enter to assign a shortcut"

// Collapse javadoc comments: Alt + "-"
// Expand javadoc comments: Alt + "="

// Tools > Generate Javadoc

// Tools > Generate UML Diagram
// ---------------------------------------------------------------------------

package com.company;

import com.company.data.course.*;
import com.company.data.person.*;

import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;

import static java.lang.Boolean.parseBoolean;
import static java.lang.Integer.parseInt;

/**
 * The type Course scheduler main.
 */
public class CourseSchedulerMain {

    /**
     * The entry point of application.
     *
     * @param args the input arguments
     * @throws IOException the io exception
     */
    public static void main(String[] args) throws IOException {

        //---------------------------------------------------------------------
        // CONSTANTS & VARIABLES
        //---------------------------------------------------------------------

        final String WELCOME =
                "\n--------------------------------" +
                        "\nWELCOME TO THE COURSE SCHEDULER" +
                        "\n--------------------------------" +
                        "\n[0] Exit the program" +
                        "\n[1] Log in as an Admin" +
                        "\n[2] Log in as a  Faculty" +
                        "\n[3] Log in as a  Student";

        final String ADMIN =
                "\n--------------------------" +
                        "\nYOU ARE LOGGED IN AS ADMIN" +
                        "\n--------------------------" +
                        "\n[0] Exit - to COURSE SCHEDULER" +
                        "\n[1] View Course Schedule" +
                        "\n[2] View Faculty List" +
                        "\n[3] View Student List" +
                        "\n[4] Output file ScheduledCourseSessions.txt" +
                        "\n[5] Output file UnscheduledCourseSessions.txt" +
                        "\n[6] Output file Faculty.txt" +
                        "\n[7] Output file ScheduledStudents.txt" +
                        "\n[8] Output file UnscheduledStudents.txt";

        final String FACULTY =
                "\n--------------------------" +
                        "\nYOU ARE LOGGED IN AS FACULTY" +
                        "\n--------------------------" +
                        "\n[0] Exit - to COURSE SCHEDULER" +
                        "\n[1] View Course Schedule" +
                        "\n[2] View My Course List" +
                        "\n[3] View My Information" +
                        "\n[4] Assign myself to a Course" +
                        "\n[5] Remove myself from a Course";

        final String STUDENT =
                "\n--------------------------" +
                        "\nYOU ARE LOGGED IN AS STUDENT" +
                        "\n--------------------------" +
                        "\n[0] Exit - to COURSE SCHEDULER" +
                        "\n[1] View Course Schedule" +
                        "\n[2] View My Course List" +
```

```
92                      "\n[3] View My Information" +
93                      "\n[4] Enroll myself to a Course" +
94                      "\n[5] Unenroll myself from a Course";
95
96          // INPUT FILE NAMES
97
98          final String FACULTY_INPUT_FILE = "src\\com\\company\\inputFiles\\info_faculty_input.txt";
99          final String STUDENT_INPUT_FILE = "src\\com\\company\\inputFiles\\info_student_input.txt";
100         final String COURSE_INPUT_FILE = "src\\com\\company\\inputFiles\\info_course_input.txt";
101
102         final String SESSION_1A_INPUT_FILE = "src\\com\\company\\inputFiles\\1A_session_id_input.txt";
103         final String SESSION_1B_INPUT_FILE = "src\\com\\company\\inputFiles\\1B_session_id_input.txt";
104         final String SESSION_1C_INPUT_FILE = "src\\com\\company\\inputFiles\\1C_session_id_input.txt";
105         final String SESSION_1D_INPUT_FILE = "src\\com\\company\\inputFiles\\1D_session_id_input.txt";
106         final String SESSION_4A_INPUT_FILE = "src\\com\\company\\inputFiles\\4A_session_id_input.txt";
107
108         // OUTPUT FILE PATHS
109
110         final String SCHEDULED_COURSE_SESSIONS_OUTPUT = "out\\outputFiles\\ScheduledCourseSessions.txt";
111         final String UNSCHEDULED_COURSE_SESSIONS_OUTPUT = "out\\outputFiles\\UnscheduledCourseSessions.txt";
112         final String FACULTY_OUTPUT = "out\\outputFiles\\Faculty.txt";
113         final String SCHEDULED_STUDENTS_OUTPUT = "out\\outputFiles\\ScheduledStudents.txt";
114         final String UNSCHEDULED_STUDENTS_OUTPUT = "out\\outputFiles\\UnscheduledStudents.txt";
115
116         // FOR PERSON - 10 categories
117
118         String firstName;
119         String middleName;
120         String lastName;
121         String email;
122         String tel;
123         String street;
124         String city;
125         String state;
126         String zip;
127         String id;
128
129         // FOR FACULTIES - 2 categories
130
131         String hireDate;
132         String tenured;
133
134         // FOR STUDENTS - 3 categories
135
136         String dob;
137         String gpa;
138         String dateStart;
139
140         // FOR COURSE
141         String department;
142         String code;
143         String description;
144         String min;
145         String max;
146
147         String line;
148
149         int loginOption = 0;
150         String loginNameTel;
151
152         String fileData;
153         FileOutputStream fos;
154
155         //-------------------------------------------------------------------------
156         // START GENERATE INFORMATION FROM INPUT FILES
157         //-------------------------------------------------------------------------
158
159         PersonDirectory personDirectory = new PersonDirectory();
160         CourseDirectory courseDirectory = new CourseDirectory();
161
162         Scanner input;
163
164         // GENERATE LIST OF FACULTY OBJECTS WITH INFO FROM INPUT FILE
165         try {
166             File facultyFile = new File(FACULTY_INPUT_FILE);
167             input = new Scanner(facultyFile);
168
169             while (input.hasNext()) {
170
171                 line = input.nextLine();
172
173                 String[] lineAr = line.split("_");
174
175                 firstName = lineAr[0];
176                 middleName = lineAr[1];
177                 lastName = lineAr[2];
178                 email = lineAr[3];
179                 tel = lineAr[4];
180                 street = lineAr[5];
181                 city = lineAr[6];
182                 state = lineAr[7];
```

```
                zip = lineAr[8];
                id = lineAr[9];
                hireDate = lineAr[10];
                tenured = lineAr[11];

                Person person = new Faculty(new PersonName(firstName, middleName, lastName), email, tel,
                        new PersonAddress(street, city, state, zip), id, hireDate, parseBoolean(tenured));

                personDirectory.addPerson(person);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        // GENERATE LIST OF STUDENT OBJECTS WITH INFO FROM INPUT FILE
        try {
            File studentFile = new File(STUDENT_INPUT_FILE);
            input = new Scanner(studentFile);

            while (input.hasNext()) {

                line = input.nextLine();

                String[] lineAr = line.split("_");

                firstName = lineAr[0];
                middleName = lineAr[1];
                lastName = lineAr[2];
                email = lineAr[3];
                tel = lineAr[4];
                street = lineAr[5];
                city = lineAr[6];
                state = lineAr[7];
                zip = lineAr[8];
                id = lineAr[9];
                dob = lineAr[10];
                gpa = lineAr[11];
                dateStart = lineAr[12];

                Person person = new Student(new PersonName(firstName, middleName, lastName), email, tel,
                        new PersonAddress(street, city, state, zip), id, dob, gpa, dateStart);

                personDirectory.addPerson(person);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }

        // GENERATE LIST OF COURSE OBJECTS WITH INFO FROM INPUT FILE
        try {
            File courseFile = new File(COURSE_INPUT_FILE);
            input = new Scanner(courseFile);
            Course course;

            while (input.hasNext()) {

                line = input.nextLine();

                String[] lineAr = line.split("_");

                department = lineAr[0];
                code = lineAr[1];
                description = lineAr[2];
                id = lineAr[3];
                min = lineAr[4];
                max = lineAr[5];

                course = new Course(department, code, description, id, parseInt(min), parseInt(max));

                courseDirectory.addCourse(course);
            }

            // GENERATE LIST OF SESSIONS FOR EACH COURSE
            courseDirectory.generateSession(courseDirectory.getCourseList().get(0).getCourseId(), SESSION_1A_INPUT_FILE);
            courseDirectory.generateSession(courseDirectory.getCourseList().get(1).getCourseId(), SESSION_1B_INPUT_FILE);
            courseDirectory.generateSession(courseDirectory.getCourseList().get(2).getCourseId(), SESSION_1C_INPUT_FILE);
            courseDirectory.generateSession(courseDirectory.getCourseList().get(3).getCourseId(), SESSION_1D_INPUT_FILE);
            courseDirectory.generateSession(courseDirectory.getCourseList().get(4).getCourseId(), SESSION_4A_INPUT_FILE);

        } catch (Exception e) {
            e.printStackTrace();
        }

        //-------------------------------------------------------------------
        // START COURSE SCHEDULER PROGRAM
        //-------------------------------------------------------------------

        input = new Scanner(System.in);
        System.out.println(WELCOME);
        System.out.print("Option: ");
        loginOption = input.nextInt();
```

```
274
275         while (loginOption != 0) {
276             switch (loginOption) {
277                 //-------------------------------------------------------------
278                 // ADMIN LOGIN
279                 //-------------------------------------------------------------
280                 case 1 -> {
281                     System.out.println(ADMIN);
282                     System.out.print("Option: ");
283                     loginOption = input.nextInt();
284                     while (loginOption != 0) {
285                         switch (loginOption) {
286                             case 1: // View Course Session
287                                 System.out.println(courseDirectory.printCourseWSession());
288                                 break;
289
290                             case 2: // View Faculty List
291                                 System.out.println(personDirectory.getFacultyAsString());
292                                 break;
293
294                             case 3: // View Student List
295                                 System.out.println(personDirectory.getStudentAsString());
296                                 break;
297
298                             case 4: // ScheduledCourseSessions.txt
299                                 try {
300                                     fileData = courseDirectory.getScheduledCourseSessionsOutputData();
301
302                                     fos = new FileOutputStream(SCHEDULED_COURSE_SESSIONS_OUTPUT);
303                                     fos.write(fileData.getBytes());
304                                     fos.flush();
305                                     fos.close();
306
307                                     System.out.println("\nOutputting file ScheduledCourseSessions.txt...");
308                                 } catch (Exception e) {
309                                     e.printStackTrace();
310                                 }
311                                 break;
312
313                             case 5: // UnscheduledCourseSessions.txt
314                                 try {
315                                     fileData = courseDirectory.getUnscheduledCourseSessionsOutputData();
316
317                                     fos = new FileOutputStream(UNSCHEDULED_COURSE_SESSIONS_OUTPUT);
318                                     fos.write(fileData.getBytes());
319                                     fos.flush();
320                                     fos.close();
321
322                                     System.out.println("\nOutputting file UnscheduledCourseSessions.txt...");
323                                 } catch (Exception e) {
324                                     e.printStackTrace();
325                                 }
326                                 break;
327
328                             case 6: // Faculty.txt
329                                 try {
330                                     fileData = personDirectory.getFacultyOutputData();
331
332                                     fos = new FileOutputStream(FACULTY_OUTPUT);
333                                     fos.write(fileData.getBytes());
334                                     fos.flush();
335                                     fos.close();
336
337                                     System.out.println("\nOutputting file Faculty.txt...");
338                                 } catch (Exception e) {
339                                     e.printStackTrace();
340                                 }
341                                 break;
342
343                             case 7: // ScheduledStudents.txt
344                                 try {
345                                     fileData = personDirectory.getScheduledStudentsOutputData();
346
347                                     fos = new FileOutputStream(SCHEDULED_STUDENTS_OUTPUT);
348                                     fos.write(fileData.getBytes());
349                                     fos.flush();
350                                     fos.close();
351
352                                     System.out.println("\nOutputting file ScheduledStudents.txt...");
353                                 } catch (Exception e) {
354                                     e.printStackTrace();
355                                 }
356                                 break;
357
358                             case 8: // UnscheduledStudents.txt
359                                 try {
360                                     fileData = personDirectory.getUnscheduledStudentsOutputData();
361
362                                     fos = new FileOutputStream(UNSCHEDULED_STUDENTS_OUTPUT);
363                                     fos.write(fileData.getBytes());
364                                     fos.flush();
```

```
                                    fos.close();

                                    System.out.println("\nOutputting file UnscheduledStudents.txt...");
                                } catch (Exception e) {
                                    e.printStackTrace();
                                }
                                break;
                            }
                            System.out.println(ADMIN);
                            System.out.print("Option: ");
                            loginOption = input.nextInt();
                        }
                    }

                    //-------------------------------------------------------------
                    // FACULTY LOGIN
                    //-------------------------------------------------------------
                    // TEST INPUT: Zeph Prissy Spalding 480586
                    // Zeph_Prissy_Spalding_zps@gmail.com_480586_KLM_Mission Viejo_CA_92692_F001_01/01/2010_true
                    case 2 -> {
                        System.out.println();
                        System.out.println("Enter your full name and phone number as the following format");
                        System.out.print("(FirstName MiddleName LastName 123456): ");
                        firstName = input.next();
                        middleName = input.next();
                        lastName = input.next();
                        tel = input.next();
                        System.out.println(FACULTY);
                        System.out.print("Option: ");
                        loginOption = input.nextInt();
                        while (loginOption != 0) {
                            switch (loginOption) {
                                case 1: // View Course Schedule
                                    System.out.println(courseDirectory.printCourseWSession());
                                    break;

                                case 2: // View My Course List
                                    System.out.println(((Faculty) personDirectory.findPerson(tel)).getFacultyCourseList());
                                    break;

                                case 3: // View My Information
                                    System.out.println(personDirectory.findPerson(tel).toString());
                                    break;

                                case 4: // Assign myself to a Course
                                    System.out.print("\nEnter the Id of the Course you would like to teach: ");
                                    String courseIdAdd = input.next().toUpperCase();
                                    Course courseToAdd = courseDirectory.findCourse(courseIdAdd);

                                    ((Faculty) personDirectory.findPerson(tel)).addCourse(courseToAdd);

                                    System.out.println();
                                    System.out.println("----------------------------------");
                                    System.out.println("SESSIONS AVAILABLE FOR THIS COURSE");
                                    System.out.println("----------------------------------");
                                    for (int i = 0; i < courseToAdd.sessionList.size(); i++) {
                                        System.out.print(courseToAdd.sessionList.get(i).toString());
                                    }

                                    System.out.print("\nEnter the Id of the Session you would like to teach: ");
                                    String sessionIdAdd = input.next().toUpperCase();
                                    Session sessionToAdd = courseDirectory.findCourse(courseIdAdd).findSession(sessionIdAdd);

                                    sessionToAdd.setTeacher((Faculty) personDirectory.findPerson(tel));

                                    System.out.println();
                                    System.out.println("----------------------------------");
                                    System.out.println("SESSIONS FOR THIS COURSE IS UPDATED");
                                    System.out.println("----------------------------------");
                                    for (int i = 0; i < courseToAdd.sessionList.size(); i++) {
                                        System.out.print(courseToAdd.sessionList.get(i).toString());
                                    }
                                    break;

                                case 5: // Remove myself from a Course
                                    System.out.print("\nEnter the Id of the Course you would like to quit: ");
                                    String courseIdRemove = input.next().toUpperCase();
                                    Course courseToRemove = courseDirectory.findCourse(courseIdRemove);

                                    ((Faculty) personDirectory.findPerson(tel)).removeCourse(courseToRemove);

                                    System.out.println();
                                    System.out.println("----------------------------------");
                                    System.out.println("SESSIONS AVAILABLE FOR THIS COURSE");
                                    System.out.println("----------------------------------");
                                    for (int i = 0; i < courseToRemove.sessionList.size(); i++) {
                                        System.out.print(courseToRemove.sessionList.get(i).toString());
                                    }

                                    System.out.print("\nEnter the Id of the Session you would like to quit: ");
                                    String sessionIdRemove = input.next().toUpperCase();
```

```
                                Session sessionToRemove = courseDirectory.findCourse(courseIdRemove).findSession(sessionIdRemove);

                                sessionToRemove.unsetTeacher();

                                System.out.println();
                                System.out.println("-----------------------------------");
                                System.out.println("SESSIONS FOR THIS COURSE IS UPDATED");
                                System.out.println("-----------------------------------");
                                for (int i = 0; i < courseToRemove.sessionList.size(); i++) {
                                    System.out.print(courseToRemove.sessionList.get(i).toString());
                                }
                                break;
                            default:
                        }
                        System.out.println(FACULTY);
                        System.out.print("Option: ");
                        loginOption = input.nextInt();
                    }
                }
                //-------------------------------------------------------------
                // STUDENT LOGIN
                //-------------------------------------------------------------
                // TEST INPUT: Ruthie Bernadine Hanley 182575
                // TEST INPUT: Camille Alfred Emmett 681690
                // Ruthie_Bernadine_Hanley_rbh@gmail.com_182575_JKL_Mission Viejo_CA_92692_S010_10/01/1998_4.0_10/20/2016
                // Camille_Alfred_Emmett_cae@gmail.com_681690_IJK_Mission Viejo_CA_92692_S009_09/01/1998_3.9_09/20/2016
                case 3 -> {
                    System.out.println();
                    System.out.println("Enter your full name and phone number as the following format");
                    System.out.print("(FirstName MiddleName LastName 123456): ");
                    firstName = input.next();
                    middleName = input.next();
                    lastName = input.next();
                    tel = input.next();
                    System.out.println(STUDENT);
                    System.out.print("Option: ");
                    loginOption = input.nextInt();
                    while (loginOption != 0) {
                        switch (loginOption) {
                            case 1: // View Course Schedule
                                System.out.println(courseDirectory.printCourseWSession());
                                break;

                            case 2: // View My Course List
                                System.out.println(((Student) personDirectory.findPerson(tel)).getCourseScheduledList());
                                break;

                            case 3: // View My Information
                                System.out.println(personDirectory.findPerson(tel).toString());
                                break;

                            case 4: // Enroll myself to a Course
                                System.out.print("\nEnter the Id of the Course you would like to enroll: ");
                                String courseIdAdd = input.next().toUpperCase();
                                Course courseToAdd = courseDirectory.findCourse(courseIdAdd);

                                // ADDING COURSE HERE
                                ((Student) personDirectory.findPerson(tel)).addCourse(courseToAdd);

                                System.out.println();
                                System.out.println("-----------------------------------");
                                System.out.println("SESSIONS AVAILABLE FOR THIS COURSE");
                                System.out.println("-----------------------------------");
                                for (int i = 0; i < courseToAdd.sessionList.size(); i++) {
                                    System.out.print(courseToAdd.sessionList.get(i).toString());
                                }

                                System.out.print("\nEnter the Id of the Session you would like to enroll: ");
                                String sessionIdAdd = input.next().toUpperCase();
                                Session sessionToAdd = courseDirectory.findCourse(courseIdAdd).findSession(sessionIdAdd);

                                // ADDING SESSION HERE
                                sessionToAdd.addStudent((Student) personDirectory.findPerson(tel));

                                // UPDATE COURSE WITH SESSION + STUDENT
                                courseDirectory.findCourse(courseIdAdd).enrollStudent(((Student) personDirectory.findPerson(tel
)), sessionToAdd);

                                System.out.println();
                                System.out.println("-----------------------------------");
                                System.out.println("SESSIONS FOR THIS COURSE IS UPDATED");
                                System.out.println("-----------------------------------");
                                for (int i = 0; i < courseToAdd.sessionList.size(); i++) {
                                    System.out.print(courseToAdd.sessionList.get(i).toString());
                                }
                                break;

                            case 5: // Unenroll myself from a Course
                                System.out.print("\nEnter the Id of the Course you would like to enroll: ");
                                String courseIdRemove = input.next().toUpperCase();
                                Course courseToRemove = courseDirectory.findCourse(courseIdRemove);
```

```
546
547                                 // REMOVE COURSE HERE
548                                 ((Student) personDirectory.findPerson(tel)).removeCourse(courseToRemove);
549
550                                 System.out.println();
551                                 System.out.println("-----------------------------------");
552                                 System.out.println("SESSIONS AVAILABLE FOR THIS COURSE");
553                                 System.out.println("-----------------------------------");
554                                 for (int i = 0; i < courseToRemove.sessionList.size(); i++) {
555                                     System.out.print(courseToRemove.sessionList.get(i).toString());
556                                 }
557
558                                 System.out.print("\nEnter the Id of the Session you would like to enroll: ");
559                                 String sessionIdRemove = input.next().toUpperCase();
560                                 Session sessionToRemove = courseDirectory.findCourse(courseIdRemove).findSession(sessionIdRemove);
561
562                                 // REMOVE SESSION HERE
563                                 sessionToRemove.removeStudent((Student) personDirectory.findPerson(tel));
564
565                                 System.out.println();
566                                 System.out.println("-----------------------------------");
567                                 System.out.println("SESSIONS FOR THIS COURSE IS UPDATED");
568                                 System.out.println("-----------------------------------");
569                                 for (int i = 0; i < courseToRemove.sessionList.size(); i++) {
570                                     System.out.print(courseToRemove.sessionList.get(i).toString());
571                                 }
572                                 break;
573                         }
574
575                         System.out.println(STUDENT);
576                         System.out.print("Option: ");
577                         loginOption = input.nextInt();
578                     }
579                 }
580             }
581
582             System.out.println(WELCOME);
583             System.out.print("Option: ");
584             loginOption = input.nextInt();
585         } // Program running Switch statements
586     }
587 }
588
```