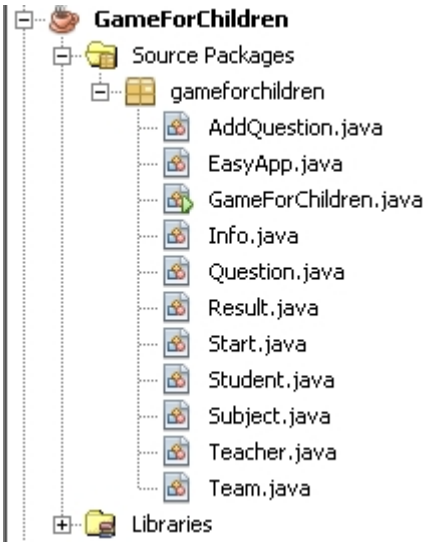
## Appendix

The main structure of my program:




```
1 package gameforchildren;
2
3  /**
4  * @date xx
5  * Game for children
6  * IDE - NetBeans
7  */
8 import java.awt.*;
9
10 public class GameForChildren extends EasyApp {
11
12     public static void main(String[] args) {
13
14         new GameForChildren();
15
16     }
17     Label Title = addLabel("Quiz Game for PYP Students", 70, 40, 500, 60, this);
18     Button Teacher = addButton("For Teacher", 60, 110, 120, 60, this);
19     Button Student = addButton("For Student", 220, 110, 120, 60, this);
20     Button Exit = addButton("Exit", 140, 200, 120, 60, this);
21     Label CopyRight = addLabel("Copyright © ESIBCS", 240, 280, 200, 50, this);
22
23     public GameForChildren() {
24
25         setTitle("Quiz For PYP Students");
26         Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
27         Title.setFont(new Font("Arial", 0, 20));
28         Teacher.setFont(new Font("Arial", 0, 15));
29         Student.setFont(new Font("Arial", 0, 15));
30         Exit.setFont(new Font("Arial", 0, 15));
31         setBounds(50, 50, 400, 350);
32     }
```

```
33
34    public void actions(Object source, String command) {
35        if (source == Teacher) {
36            new Teacher();
37        }
38        if (source == Student) {
39            new Student();
40        }
41        if (source == Exit) {
42            System.exit(0);
43        }
44    }
45 }
```

```
1 package gameforchildren;
2
3 import java.awt.*;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Teacher extends EasyApp {
8
9    Label Title = addLabel("Teacher's Section", 50, 50, 400, 50, this);
10    Button Math = addButton("Math", 100, 100, 100, 100, this);
11    Button CS = addButton("CS", 400, 100, 100, 100, this);
12    Button Art = addButton("Art", 100, 250, 100, 100, this);
13    Button Music = addButton("Music", 400, 250, 100, 100, this);
14    Button UOI = addButton("UOI", 250, 175, 100, 100, this);
15    Button Close = addButton("Close", 60, 380, 80, 40, this);
16    Button Exit = addButton("Exit", 150, 380, 80, 40, this);
17
18    public Teacher() {
19
20        setTitle("Teacher's Section");
21        Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
22        Title.setFont(new Font("Arial", 0, 30));
23        Math.setFont(new Font("Arial", 0, 20));
24        Art.setFont(new Font("Arial", 0, 20));
25        CS.setFont(new Font("Arial", 0, 20));
26        UOI.setFont(new Font("Arial", 0, 20));
27        Music.setFont(new Font("Arial", 0, 20));
28        setBounds(50, 50, 600, 450);
29    }
30    static String subject;
31
32    public void actions(Object source, String command) {
33
34        if (source == Math) {
35            subject = "Math";
36            SubjectChoise();
37        }
38        if (source == CS) {
39            subject = "CS";
40            SubjectChoise();
41        }
42        if (source == Art) {
43            subject = "Art";
44            SubjectChoise();
45        }
46        if (source == Music) {
47            subject = "Music";
48            SubjectChoise();
49        }
50        if (source == UOI) {
51            subject = "UOI";
52            SubjectChoise();
53        }
54        if (source == Close) {
55            dispose();
```

```
56        }
57        if (source == Exit) {
58            System.exit(0);
59        }
60    }
61
62    void SubjectChoise() {
63
64        // .txt file should be created for each subject
65        try {
66            RandomAccessFile subjectFile = new RandomAccessFile(subject + ".txt", "rw");
67            if (subjectFile.length() == 0) {
68                subjectFile.writeBytes(subject + " Questions:\n");
69            }
70        } catch (IOException e) {
71            e.getMessage();
72        }
73        new Subject();
74    }
75 }
```

```
 1 package gameforchildren;
 2
 3 import java.awt.*;
 4 import java.io.IOException;
 5 import java.io.RandomAccessFile;
 6
 7 public class Subject extends EasyApp {
 8
 9    Label Title = addLabel("Subject " + Teacher.subject, 40, 50, 150, 40, this);
10    Button Add = addButton("Add Question", 50, 100, 100, 40, this);
11    Button Edit = addButton("Edit Question", 50, 150, 100, 40, this);
12    Button Delete = addButton("Delete Question", 50, 200, 100, 40, this);
13    Button Close = addButton("Close", 50, 280, 100, 40, this);
14    Button Exit = addButton("Exit", 50, 330, 100, 40, this);
15    Label ListL = addLabel("List of Questions", 220, 60, 150, 30, this);
16    Button ListB = addButton("List", 430, 45, 70, 40, this);
17    List List = addList(Teacher.subject + " questions list:", 220, 110, 280, 260, this);
18
19    public Subject() {
20
21       setTitle("Subject - " + Teacher.subject);
22       Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
23       Title.setFont(new Font("Arial", 0, 20));
24       setBounds(80, 140, 540, 420);
25    }
26
27    public void actions(Object source, String command) {
28
29       if (source == Add) {
30          new AddQuestion();
31       }
32       if (source == Delete) {
33          Delete();
34       }
35       if (source == ListB) {
36          ListButton();
37       }
38       if (source == Edit) {
39          EditQuestion();
40       }
41
42       if (source == Close) {
43          dispose();
44       }
45       if (source == Exit) {
46          System.exit(0);
47       }
48    }
49    String[] arrayQuestions = new String[25];  // maximum 25 tasks will be written in one subject
                                                    file
50    int k;
51
52    public void arrayWord() {
53       k = 0;
54       try {
```

```
55          RandomAccessFile subjectFile = new RandomAccessFile(Teacher.subject + ".txt", "rw");
56          subjectFile.seek(0);
57          subjectFile.readLine(); //to read title line
58
59          while (subjectFile.length() != subjectFile.getFilePointer()) {
60             arrayQuestions[k] = subjectFile.readLine();
61             k++;
62          }
63       } catch (IOException e) {
64          e.getMessage();
65       }
66    }
67
68    void ListButton() {
69       List.removeAll();
70       List.add(Teacher.subject + " questions list:\n");
71       //read questions/lines from the appropriate file
72       try {
73          RandomAccessFile subjectFile = new RandomAccessFile(Teacher.subject + ".txt", "rw");
74          subjectFile.seek(0);
75          subjectFile.readLine();
76          while (subjectFile.getFilePointer() != subjectFile.length()) {
77             List.add(subjectFile.readLine());
78          }
79       } catch (IOException e) {
80          e.getMessage();
81       }
82    }
83
84    //reads file records/lines and saves in the array
85    public void Delete() {
86       String questionDel = List.getSelectedItem();   //question, we need to delete from list
87       arrayWord();
88
89       for (int j = 0; j < k; j++) {     //delete word from array
90          if (arrayQuestions[j].equals(questionDel)) {
91             for (int i = j; i < k - 1; i++) {
92                arrayQuestions[i] = arrayQuestions[i + 1];
93             }
94             arrayQuestions[k - 1] = "";
95          }
96       }
97
98       try {  //write array elements in file again - sutable task is already deleted
99          RandomAccessFile subjectFile = new RandomAccessFile(Teacher.subject + ".txt", "rw");
100         subjectFile.seek(0);
101         subjectFile.setLength(0);
102         subjectFile.writeBytes(Teacher.subject + " Questions:\n");
103         for (int z = 0; z < k - 1; z++) {
104            subjectFile.writeBytes(arrayQuestions[z] + "\n");
105         }
106      } catch (IOException e) {
107         e.getMessage();
108      }
109   }
```

```
110
111    void EditQuestion() {
112        String questionEdit = List.getSelectedItem();   //question, we need to Edit
113        Delete();  // first we delete the selected question
114        // and then edit and save in the file again with AddQuestion method
115        new AddQuestion(questionEdit);
116    }
117 }
```

```
 1 package gameforchildren;
 2
 3 import java.awt.*;
 4 import java.io.IOException;
 5 import java.io.RandomAccessFile;
 6
 7 public class AddQuestion extends EasyApp {
 8
 9    Label Title = addLabel("Adding Questions - subject " + Teacher.subject, 50, 50, 400, 40, this);
10     TextField Question = addTextField("", 50, 90, 300, 30, this);
11     Label WriteQuestion = addLabel("Question", 50, 75, 300, 10, this);
12     TextField AnswerA = addTextField("", 50, 170, 200, 30, this);
13     Label WriteAnswerA = addLabel("Answer A", 50, 155, 200, 10, this);
14     TextField AnswerB = addTextField("", 300, 170, 200, 30, this);
15     Label WriteAnswerB = addLabel("Answer B", 300, 155, 200, 10, this);
16     TextField AnswerC = addTextField("", 50, 250, 200, 30, this);
17     Label WriteAnswerC = addLabel("Answer C", 50, 235, 200, 10, this);
18     TextField AnswerD = addTextField("", 300, 250, 200, 30, this);
19     Label WriteAnswerD = addLabel("Answer D", 300, 235, 200, 10, this);
20     Label CorrectAnswerL = addLabel("Choose correct answer: (A, B, C or D)", 100, 330, 250, 30,
                                       this);
21     TextField CorrectAnswer = addTextField("", 370, 330, 100, 30, this);
22     Label ValueL = addLabel("Choose value: (1, 2 or 3)", 100, 380, 180, 30, this);
23     TextField Value = addTextField("", 280, 380, 100, 30, this);
24     Button Add = addButton("Add Question", 50, 430, 100, 40, this);
25     Button Close = addButton("Close", 300, 430, 80, 40, this);
26     Button Exit = addButton("Exit", 400, 430, 80, 40, this);
27
28     public AddQuestion() {
29
30        setTitle("Adding Questions");
31        Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
32        Title.setFont(new Font("Arial", 0, 20));
33        setBounds(50, 50, 550, 490);
34        ValueL.setFont(new Font("Arial", 0, 15));
35        CorrectAnswerL.setFont(new Font("Arial", 0, 15));
36     }
37
38     public AddQuestion(String questionEdit) {
39
40        setTitle("Adding Questions");
41        Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
42        Title.setFont(new Font("Arial", 0, 20));
43        setBounds(50, 50, 550, 490);
44        ValueL.setFont(new Font("Arial", 0, 15));
45        CorrectAnswerL.setFont(new Font("Arial", 0, 15));
46
47        // get all fields initial values
48        // then Edit data and add in the file with Add() method
49        int index1 = 0;
50        int index2 = questionEdit.indexOf("~");
51        Question.setText(questionEdit.substring(0, index2));
52        index1 = index2 + 1;
53        index2 = questionEdit.indexOf("~", index1);
54        AnswerA.setText(questionEdit.substring(index1, index2));
```

```
55      index1 = index2 + 1;
56      index2 = questionEdit.indexOf("~", index1);
57      AnswerB.setText(questionEdit.substring(index1, index2));
58      index1 = index2 + 1;
59      index2 = questionEdit.indexOf("~", index1);
60      AnswerC.setText(questionEdit.substring(index1, index2));
61      index1 = index2 + 1;
62      index2 = questionEdit.indexOf("~", index1);
63      AnswerD.setText(questionEdit.substring(index1, index2));
64      index1 = index2 + 1;
65      index2 = questionEdit.indexOf("~", index1);
66      CorrectAnswer.setText(questionEdit.substring(index1, index2));
67      index1 = index2 + 1;
68      Value.setText(questionEdit.substring(index1));
69  }
70
71  public void actions(Object source, String command) {
72      if (source == Add) {
73          Add();
74      }
75      if (source == Close) {
76          dispose();
77      }
78      if (source == Exit) {
79          System.exit(0);
80      }
81  }
82
83  //methods section
84  void Add() {
85      String question = Question.getText();
86      String answerA = AnswerA.getText();
87      String answerB = AnswerB.getText();
88      String answerC = AnswerC.getText();
89      String answerD = AnswerD.getText();
90
91      String correctA = CorrectAnswer.getText();
92      String value = Value.getText();
93      if ((correctA.equals("A") || correctA.equals("B") || correctA.equals("C") ||
            correctA.equals("D")) && (value.equals("1") || value.equals("2") || value.equals("3"))) {
94          if (question.equals("") || answerA.equals("") || answerB.equals("") || answerC.equals("") ||
            answerD.equals("") || correctA.equals("") || value.equals("")) {
95              outputString("Worning!\nFill in all fileds!");
96          } else {
97
98              //write question in appropriate file directly
99              try {
100                 RandomAccessFile subjectFile = new RandomAccessFile(Teacher.subject + ".txt",
                                "rw");
101                 subjectFile.seek(subjectFile.length());
102                 subjectFile.writeBytes(question + "~" + answerA + "~" + answerB + "~" + answerC
                                + "~" + answerD + "~" + correctA + "~" + value + "\n");
103
104             } catch (IOException e) {
105                 e.getMessage();
```

```
106             }
107             Question.setText("");
108             AnswerA.setText("");
109             AnswerB.setText("");
110             AnswerC.setText("");
111             AnswerD.setText("");
112             CorrectAnswer.setText("");
113             Value.setText("");
114           }
115       } else {
116           outputString("Warning!\nInvelid Data");
117       }
118   }
119 }
```

```
1 package gameforchildren;
2
3 import java.awt.*;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Student extends EasyApp {
8
9    Label Title = addLabel("Create Teams", 50, 50, 200, 50, this);
10    Button Team = addButton("Create New Team", 70, 200, 130, 80, this);
11    Label List = addLabel("List", 250, 70, 100, 30, this);
12    Button ListB = addButton("List", 420, 70, 80, 40, this);
13    List Listl = addList("List of teams:", 250, 120, 250, 250, this);
14    Button Close = addButton("Close", 50, 390, 80, 40, this);
15    Button Exit = addButton("Exit", 150, 390, 80, 40, this);
16    Button Info = addButton("Teams Info", 300, 390, 100, 40, this);
17    Button Start = addButton("Start", 420, 390, 80, 40, this);
18
19    public Student() {
20       setTitle("Create Teams");
21       Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
22       Title.setFont(new Font("Arial", 0, 20));
23       setBounds(50, 50, 550, 480);
24    }
25
26    public void actions(Object source, String command) {
27       if (source == Team) {
28          new Team();
29       }
30       if (source == ListB) {
31          ListButton();
32       }
33       if (source == Start) {
34          Start();
35       }
36       if (source == Info) {
37          Info();
38       }
39       if (source == Close) {
40          dispose();
41       }
42       if (source == Exit) {
43          System.exit(0);
44       }
45    }
46
47    void ListButton() {
48       String Line1;
49       String Line2;
50       String date;
51       Listl.removeAll();
52       Listl.add("List of teams:\n");
53       Listl.add("\n");
54       //read questions/lines from the appropriate file
55       try {
```

```
56          RandomAccessFile teamFile = new RandomAccessFile("team.txt", "rw");
57
58          while (teamFile.getFilePointer() != teamFile.length()) {
59              date = teamFile.readLine();
60              Line1 = teamFile.readLine();
61              Line2 = teamFile.readLine();
62              int index1 = Line1.indexOf("~", 2);
63              String str1 = Line1.substring(2, index1);
64              int index2 = Line2.indexOf("~", 2);
65              String str2 = Line2.substring(2, index2);
66              Listl.add(date + " | " + str1 + " - " + str2);
67          }
68      } catch (IOException e) {
69          e.getMessage();
70      }
71  }
72  static String Teams;
73
74  void Start() {
75      Teams = Listl.getSelectedItem();
76      if (Teams == null || Teams.equals("\n") || Teams.equals("List of teams:\n")) {
77          outputString("Error! choose Teams");
78      } else {
79          new Start();
80      }
81  }
82
83  void Info() {
84      Teams = Listl.getSelectedItem();
85      if (Teams == null || Teams.equals("\n") || Teams.equals("List of teams:\n")) {
86          outputString("Error! choose Teams");
87      } else {
88          new Info();
89      }
90  }
91 }
```

```
 1 package gameforchildren;
 2
 3 import java.awt.*;
 4 import java.io.IOException;
 5 import java.io.RandomAccessFile;
 6
 7 public class Team extends EasyApp {
 8
 9    Label Title = addLabel("Create Teams", 40, 40, 170, 50, this);
10     Label Date = addLabel("Enter date: (dd/mm/yyyy)", 220, 50, 140, 40, this);
11     TextField Date1 = addTextField("", 370, 50, 140, 30, this);
12     Label Name = addLabel("Name:", 50, 140, 120, 30, this);
13     Label TeamI = addLabel("I Team", 170, 100, 150, 30, this);
14     TextField Team1 = addTextField("", 170, 140, 140, 30, this);
15     Label TeamII = addLabel("II Team", 370, 100, 150, 30, this);
16     TextField Team2 = addTextField("", 370, 140, 140, 30, this);
17     Label member1 = addLabel("member 1", 50, 180, 120, 30, this);
18     TextField member11 = addTextField("", 170, 180, 140, 30, this);
19     TextField member21 = addTextField("", 370, 180, 140, 30, this);
20     Label member2 = addLabel("member 2", 50, 220, 120, 30, this);
21     TextField member12 = addTextField("", 170, 220, 140, 30, this);
22     TextField member22 = addTextField("", 370, 220, 140, 30, this);
23     Label member3 = addLabel("member 3", 50, 260, 120, 30, this);
24     TextField member13 = addTextField("", 170, 260, 140, 30, this);
25     TextField member23 = addTextField("", 370, 260, 140, 30, this);
26     Label member4 = addLabel("member 4", 50, 300, 120, 30, this);
27     TextField member14 = addTextField("", 170, 300, 140, 30, this);
28     TextField member24 = addTextField("", 370, 300, 140, 30, this);
29     Label member5 = addLabel("member 5", 50, 340, 120, 30, this);
30     TextField member15 = addTextField("", 170, 340, 140, 30, this);
31     TextField member25 = addTextField("", 370, 340, 140, 30, this);
32     Button Close = addButton("Close", 50, 400, 80, 40, this);
33     Button Exit = addButton("Exit", 150, 400, 80, 40, this);
34     Button Create = addButton("Create", 430, 400, 80, 40, this);
35
36     public Team() {
37        setTitle("Create Teams");
38        Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
39        Title.setFont(new Font("Arial", 0, 20));
40        setBounds(50, 50, 550, 480);
41        TeamI.setFont(new Font("Arial", 0, 15));
42        TeamII.setFont(new Font("Arial", 0, 15));
43     }
44
45     public void actions(Object source, String command) {
46        if (source == Create) {
47           Create();
48        }
49        if (source == Close) {
50           dispose();
51        }
52        if (source == Exit) {
53           System.exit(0);
54        }
55     }
```

```
56
57  void Create() {
58     int count1 = 1;  //Team 1 students number
59     int count2 = 1;  //Team 2 students number
60     String line1;
61     String line2;
62     if (Team1.getText().equals("") || member11.getText().equals("") ||
                Team2.getText().equals("") || member21.getText().equals("") ||
                Date1.getText().equals("")) {
63        outputString("fill appropriate fields!");
64     } else {
65        line1 = Team1.getText() + "~" + member11.getText();
66        // 1st team name and members names
67        if (!member12.getText().equals("")) {
68           count1++;
69           line1 = line1 + "~" + member12.getText();
70        }
71        if (!member13.getText().equals("")) {
72           count1++;
73           line1 = line1 + "~" + member13.getText();
74        }
75        if (!member14.getText().equals("")) {
76           count1++;
77           line1 = line1 + "~" + member14.getText();
78        }
79        if (!member15.getText().equals("")) {
80           count1++;
81           line1 = line1 + "~" + member15.getText();
82        }
83
84        line2 = Team2.getText() + "~" + member21.getText();
85        // 2nd team name and members names
86        if (!member22.getText().equals("")) {
87           count2++;
88           line2 = line2 + "~" + member22.getText();
89        }
90        if (!member23.getText().equals("")) {
91           count2++;
92           line2 = line2 + "~" + member23.getText();
93        }
94        if (!member24.getText().equals("")) {
95           count2++;
96           line2 = line2 + "~" + member24.getText();
97        }
98        if (!member25.getText().equals("")) {
99           count2++;
100           line2 = line2 + "~" + member25.getText();
101        }
102
103        line1 = count1 + "~" + line1;
104        line2 = count2 + "~" + line2;
105
106        try {
107           RandomAccessFile teamFile = new RandomAccessFile("team.txt", "rw");
108           teamFile.seek(teamFile.length());
```

```
109             teamFile.writeBytes(Date1.getText() + "\n");
110             teamFile.writeBytes(line1 + "\n");
111             teamFile.writeBytes(line2 + "\n");
112         } catch (IOException e) {
113             e.getMessage();
114         }
115         clean();
116     }
117 }
118
119 void clean() {
120     Team1.setText("");
121     member11.setText("");
122     member12.setText("");
123     member13.setText("");
124     member14.setText("");
125     member15.setText("");
126
127     Team2.setText("");
128     member21.setText("");
129     member22.setText("");
130     member23.setText("");
131     member24.setText("");
132     member25.setText("");
133
134     Date1.setText("");
135 }
```

```
1 package gameforchildren;
 2
 3 import java.awt.*;
 4 import java.io.IOException;
 5 import java.io.RandomAccessFile;
 6
 7 public class Start extends EasyApp {
 8
 9    String competition = Student.Teams;
10     int index1 = competition.indexOf(" ");
11     String date = competition.substring(0, index1);
12     String team1 = competition.substring(index1 + 3, competition.indexOf(" ", index1 + 4));
13     String team2 = competition.substring(competition.lastIndexOf(" ") + 1);
14     Label Title = addLabel("Competition", 50, 50, 200, 30, this);
15     Label Teamname = addLabel("Question for: ", 330, 50, 100, 50, this);
16     Label teamName = addLabel(team1, 330, 100, 150, 50, this);
17     Label day = addLabel("Date: " + date, 70, 80, 100, 30, this);
18     Label Team1 = addLabel("Team 1 -  " + team1, 70, 110, 120, 30, this);
19     Label Team2 = addLabel("Team 2 -  " + team2, 70, 140, 120, 30, this);
20     Label chooseS = addLabel("Choose Subject:", 50, 190, 120, 30, this);
21     Label chooseV = addLabel("Choose Value:", 350, 190, 120, 30, this);
22     Choice Subject = addChoice("Art|Math|Music|CS|UOI", 50, 220, 150, 50, this);
23     Choice Value = addChoice("Value1|Value2|Value3", 350, 220, 100, 50, this);
24     Button question = addButton("Choose question", 260, 320, 110, 40, this);
25     Button result = addButton("Results", 380, 320, 110, 40, this);
26     Button Close = addButton("Close", 40, 320, 80, 40, this);
27     Button Exit = addButton("Exit", 130, 320, 80, 40, this);
28
29     public Start() {
30        setTitle("Competition");
31        Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
32        Title.setFont(new Font("Arial", 0, 20));
33        teamName.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
34        teamName.setFont(new Font("Arial", 0, 20));
35        setBounds(100, 100, 530, 400);
36        CreateArray();
37     }
38
39     public void actions(Object source, String command) {
40        if (source == question) {
41           Question();
42        }
43        if (source == result) {
44           new Result();
45        }
46        if (source == Close) {
47           dispose();
48        }
49        if (source == Exit) {
50           System.exit(0);
51        }
52     }
53
54     //create array with all questions
55     static String[][] questionArray = new String[90][9];
```

```
56  //subject, question, answer 1, answer 2, answer 3, answer 4, correct answer, value, yes/no
57  static int countQuestions;
58
59  void CreateArray() {
60     countQuestions = 0;
61     String line;
62     int i1, i2;
63
64     try {
65        RandomAccessFile questionsFileArt = new RandomAccessFile("Art.txt", "rw");
66        questionsFileArt.readLine(); //title line
67        while (questionsFileArt.length() != questionsFileArt.getFilePointer()) {
68           line = questionsFileArt.readLine();
69           i1 = 0;
70           i2 = line.indexOf("~");
71           questionArray[countQuestions][0] = "Art";
72           questionArray[countQuestions][1] = line.substring(i1, i2);
73           i1 = i2 + 1;
74           i2 = line.indexOf("~", i1);
75           questionArray[countQuestions][2] = line.substring(i1, i2);
76           i1 = i2 + 1;
77           i2 = line.indexOf("~", i1);
78           questionArray[countQuestions][3] = line.substring(i1, i2);
79           i1 = i2 + 1;
80           i2 = line.indexOf("~", i1);
81           questionArray[countQuestions][4] = line.substring(i1, i2);
82           i1 = i2 + 1;
83           i2 = line.indexOf("~", i1);
84           questionArray[countQuestions][5] = line.substring(i1, i2);
85           i1 = i2 + 1;
86           i2 = line.indexOf("~", i1);
87           questionArray[countQuestions][6] = line.substring(i1, i2);
88           i1 = i2 + 1;
89           questionArray[countQuestions][7] = line.substring(i1);
90           countQuestions++;
91        }
92        // next subject
93        RandomAccessFile questionsFileMath = new RandomAccessFile("Math.txt", "rw");
94        questionsFileMath.readLine(); //title line
95        while (questionsFileMath.length() != questionsFileMath.getFilePointer()) {
96           line = questionsFileMath.readLine();
97           i1 = 0;
98           i2 = line.indexOf("~");
99           questionArray[countQuestions][0] = "Math";
100          questionArray[countQuestions][1] = line.substring(i1, i2);
101          i1 = i2 + 1;
102          i2 = line.indexOf("~", i1);
103          questionArray[countQuestions][2] = line.substring(i1, i2);
104          i1 = i2 + 1;
105          i2 = line.indexOf("~", i1);
106          questionArray[countQuestions][3] = line.substring(i1, i2);
107          i1 = i2 + 1;
108          i2 = line.indexOf("~", i1);
109          questionArray[countQuestions][4] = line.substring(i1, i2);
110          i1 = i2 + 1;
```

```
111             i2 = line.indexOf("~", i1);
112             questionArray[countQuestions][5] = line.substring(i1, i2);
113             i1 = i2 + 1;
114             i2 = line.indexOf("~", i1);
115             questionArray[countQuestions][6] = line.substring(i1, i2);
116             i1 = i2 + 1;
117             questionArray[countQuestions][7] = line.substring(i1);
118             countQuestions++;
119         }
120         // next subject
121         RandomAccessFile questionsFileMusic = new RandomAccessFile("Music.txt", "rw");
122         questionsFileMusic.readLine(); //title line
123         while (questionsFileMusic.length() != questionsFileMusic.getFilePointer()) {
124             line = questionsFileMusic.readLine();
125             i1 = 0;
126             i2 = line.indexOf("~");
127             questionArray[countQuestions][0] = "Music";
128             questionArray[countQuestions][1] = line.substring(i1, i2);
129             i1 = i2 + 1;
130             i2 = line.indexOf("~", i1);
131             questionArray[countQuestions][2] = line.substring(i1, i2);
132             i1 = i2 + 1;
133             i2 = line.indexOf("~", i1);
134             questionArray[countQuestions][3] = line.substring(i1, i2);
135             i1 = i2 + 1;
136             i2 = line.indexOf("~", i1);
137             questionArray[countQuestions][4] = line.substring(i1, i2);
138             i1 = i2 + 1;
139             i2 = line.indexOf("~", i1);
140             questionArray[countQuestions][5] = line.substring(i1, i2);
141             i1 = i2 + 1;
142             i2 = line.indexOf("~", i1);
143             questionArray[countQuestions][6] = line.substring(i1, i2);
144             i1 = i2 + 1;
145             questionArray[countQuestions][7] = line.substring(i1);
146             countQuestions++;
147         }
148         // next subject
149         RandomAccessFile questionsFileCS = new RandomAccessFile("CS.txt", "rw");
150         questionsFileCS.readLine(); //title line
151         while (questionsFileCS.length() != questionsFileCS.getFilePointer()) {
152             line = questionsFileCS.readLine();
153             i1 = 0;
154             i2 = line.indexOf("~");
155             questionArray[countQuestions][0] = "CS";
156             questionArray[countQuestions][1] = line.substring(i1, i2);
157             i1 = i2 + 1;
158             i2 = line.indexOf("~", i1);
159             questionArray[countQuestions][2] = line.substring(i1, i2);
160             i1 = i2 + 1;
161             i2 = line.indexOf("~", i1);
162             questionArray[countQuestions][3] = line.substring(i1, i2);
163             i1 = i2 + 1;
164             i2 = line.indexOf("~", i1);
165             questionArray[countQuestions][4] = line.substring(i1, i2);
```

```
166             i1 = i2 + 1;
167             i2 = line.indexOf("~", i1);
168             questionArray[countQuestions][5] = line.substring(i1, i2);
169             i1 = i2 + 1;
170             i2 = line.indexOf("~", i1);
171             questionArray[countQuestions][6] = line.substring(i1, i2);
172             i1 = i2 + 1;
173             questionArray[countQuestions][7] = line.substring(i1);
174             countQuestions++;
175          }
176          // next subject
177          RandomAccessFile questionsFileUOI = new RandomAccessFile("UOI.txt", "rw");
178          questionsFileUOI.readLine(); //title line
179          while (questionsFileUOI.length() != questionsFileUOI.getFilePointer()) {
180             line = questionsFileUOI.readLine();
181             i1 = 0;
182             i2 = line.indexOf("~");
183             questionArray[countQuestions][0] = "UOI";
184             questionArray[countQuestions][1] = line.substring(i1, i2);
185             i1 = i2 + 1;
186             i2 = line.indexOf("~", i1);
187             questionArray[countQuestions][2] = line.substring(i1, i2);
188             i1 = i2 + 1;
189             i2 = line.indexOf("~", i1);
190             questionArray[countQuestions][3] = line.substring(i1, i2);
191             i1 = i2 + 1;
192             i2 = line.indexOf("~", i1);
193             questionArray[countQuestions][4] = line.substring(i1, i2);
194             i1 = i2 + 1;
195             i2 = line.indexOf("~", i1);
196             questionArray[countQuestions][5] = line.substring(i1, i2);
197             i1 = i2 + 1;
198             i2 = line.indexOf("~", i1);
199             questionArray[countQuestions][6] = line.substring(i1, i2);
200             i1 = i2 + 1;
201             questionArray[countQuestions][7] = line.substring(i1);
202             countQuestions++;
203          }
204       } catch (IOException e) {
205          e.getMessage();
206       }
207    }
208
209    static int count = 1; // counting question number MAX = 30 for both teams
210    static String subject;
211    static String value;
212    static String teamsName;
213    static String[][] team1array = new String[15][3];// to save information about subject, value and
          points
214    static String[][] team2array = new String[15][3];
215
216    void Question() {
217       subject = Subject.getSelectedItem();  // subject the question belongs to
218       value = Value.getSelectedItem();  // question value 1,2,3
219       count++;
```

```
220         if (count <= 30) {  //max questions = 30 - 15 for each teams
221             // 15 = 5 subject * 3 questions from each subject
222             if (count % 2 == 0) {  // even number question - for team II
223                 teamName.setText(team2);
224                 teamsName = team2;
225                 new Question();
226             } else {
227                 teamName.setText(team1);  // odd number question - for team I
228                 teamsName = team1;
229                 new Question();
230             }
231         } else {
232             outputString("The Competition is Finished!\nCheck the results!");
233         }
234     }
235 }
```

```
 1 package gameforchildren;
 2
 3 import java.awt.*;
 4 import java.io.IOException;
 5 import java.io.RandomAccessFile;
 6
 7 public class Info extends EasyApp {
 8
 9    String InfoDate1, InfoNames1, InfoPoints1;
10     Label Title = addLabel("Teams' Info", 50, 50, 200, 50, this);
11     Button Close = addButton("Close", 50, 370, 80, 40, this);
12     Button Exit = addButton("Exit", 150, 370, 80, 40, this);
13
14     public Info() {
15        setTitle("Teams' Info");
16        Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
17        Title.setFont(new Font("Arial", 0, 20));
18        setBounds(50, 50, 530, 450);
19        String line = Student.Teams;  //selected line
20        InfoDate1 = line.substring(0, line.indexOf("|") - 1); // competition/Game date
21        InfoNames1 = line.substring(line.indexOf("|") + 2);   //partisipant teams names
22        Label InfoNames = addLabel(InfoNames1, 50, 100, 200, 50, this);
23        Label InfoDate = addLabel("Date: " + InfoDate1, 50, 150, 200, 50, this);
24        InfoNames.setFont(new Font("Arial", 0, 20));
25        InfoNames.setForeground(Color.getHSBColor(0.7f, 0.8f, 0.7f));
26        InfoDate.setFont(new Font("Arial", 0, 15));
27        members();  // team members
28        points();   // results
29     }
30
31     public void actions(Object source, String command) {
32        if (source == Close) {
33           dispose();
34        }
35        if (source == Exit) {
36           System.exit(0);
37        }
38     }
39
40     void members() {
41        String line1, line2, line3;  // information about members are writen in three lines
42        String date; //Game date
43        String name1; //Team 1 name
44        String name2; //Team 2 name
45        String team1members = "";
46        String team2members = "";
47        try {
48           RandomAccessFile teamFile = new RandomAccessFile("team.txt", "rw");
49           // all registered teams information is written in this file
50           while (teamFile.getFilePointer() != teamFile.length()) {
51              line1 = teamFile.readLine();
52              line2 = teamFile.readLine();
53              line3 = teamFile.readLine();
54              date = line1.substring(0, line1.length());
55              int index1 = line2.indexOf("~");
```

```
56              name1 = line2.substring(index1 + 1, line2.indexOf("~", index1 + 1)); //1st team name
        and members
57              int index2 = line3.indexOf("~");
58              name2 = line3.substring(index2 + 1, line3.indexOf("~", index2 + 1)); //2nd team name
        and members
59
60              if (InfoNames1.equals(name1 + " - " + name2) && date.equals(InfoDate1)) {
61                  //search teams according date and names
62                  team1members = line2.substring(index1 + 1);
63                  team1members = team1members.replaceAll("~", " - ");
64                  team1members = team1members.replaceFirst(" - ", " | ");
65                  team2members = line3.substring(index2 + 1);
66                  team2members = team2members.replaceAll("~", " - ");
67                  team2members = team2members.replaceFirst(" - ", " | ");
68                  Label infoMembers1 = addLabel(team1members, 50, 250, 500, 50, this);
69                  infoMembers1.setFont(new Font("Arial", 0, 15));
70                  Label infoMembers2 = addLabel(team2members, 50, 300, 500, 50, this);
71                  infoMembers2.setFont(new Font("Arial", 0, 15));
72                  break;
73              }
74          }
75      } catch (IOException e) {
76          e.getMessage();
77      }
78  }
79
80  void points() {
81      String line1, line2, line3;  // in result file competition result is written in three lines
82      try {
83          RandomAccessFile resultFile = new RandomAccessFile("Results.txt", "rw");  //all results
              are saved in file
84          while (resultFile.getFilePointer() != resultFile.length()) {
85              line1 = resultFile.readLine();
86              line2 = resultFile.readLine();
87              line3 = resultFile.readLine();
88
89              if (line1.equals(InfoDate1) && line2.equals(InfoNames1)) {
90                  Label InfoPoints = addLabel(line3, 50, 190, 200, 50, this);
91                  InfoPoints.setFont(new Font("Arial", 0, 25));
92                  InfoPoints.setForeground(Color.getHSBColor(0.68f, 0.9f, 0.9f));
93              }
94          }
95      } catch (IOException e) {
96          e.getMessage();
97      }
98  }
99 }
```

```
1 package gameforchildren;
2
3 import java.awt.*;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Result extends EasyApp {
8
9    Label Title = addLabel("Teams Result", 50, 50, 200, 50, this);
10    Button Close = addButton("Close", 80, 250, 80, 40, this);
11    Button Finish = addButton("Finish", 180, 250, 80, 40, this);
12    String competition = Student.Teams;
13    int index1 = competition.indexOf(" ");
14    String date = competition.substring(0, index1);
15    String team1 = competition.substring(index1 + 3, competition.indexOf(" ", index1 + 4));
16    String team2 = competition.substring(competition.lastIndexOf(" ") + 1);
17
18    public Result() {
19       setTitle("R E S U L T");
20       Title.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
21       Title.setFont(new Font("Arial", 0, 20));
22
23       Label teamName1 = addLabel(team1, 50, 100, 100, 50, this);
24       Label teamName2 = addLabel(team2, 200, 100, 100, 50, this);
25       teamName1.setFont(new Font("Arial", 0, 20));
26       teamName2.setFont(new Font("Arial", 0, 20));
27       setBounds(100, 100, 330, 320);
28       calculation();  // calculates results
29    }
30
31    public void actions(Object source, String command) {
32
33       if (source == Finish) {
34          Finish();
35       }
36       if (source == Close) {
37          dispose();
38       }
39    }
40    int team1result = 0;
41    int team2result = 0;
42
43    void calculation() {
44       int i = 0;
45       while (Start.team1array[i][2] != null && Start.team2array[i][2] != null) {
46          team1result = team1result + Integer.parseInt(Start.team1array[i][2]);
47          team2result = team2result + Integer.parseInt(Start.team2array[i][2]);
48          i++;
49       }
50
51       Label Result1 = addLabel(Integer.toString(team1result), 60, 150, 100, 50, this);
52       Label Result2 = addLabel(Integer.toString(team2result), 210, 150, 100, 50, this);
53       Result1.setFont(new Font("Arial", 0, 30));
54       Result2.setFont(new Font("Arial", 0, 30));
55    }
```

```
56
57    void Finish() {
58        String competition = Student.Teams;
59        int index1 = competition.indexOf(" ");
60        String date = competition.substring(0, index1);
61        String team1 = competition.substring(index1 + 3, competition.indexOf(" ", index1 + 4));
62        String team2 = competition.substring(competition.lastIndexOf(" ") + 1);
63
64        try {
65            RandomAccessFile resultFile = new RandomAccessFile("Results.txt", "rw");
66            resultFile.seek(resultFile.length());
67            resultFile.writeBytes(date + "\n");
68            resultFile.writeBytes(team1 + " - " + team2 + "\n");
69            resultFile.writeBytes(team1result + " : " + team2result + "\n");
70
71        } catch (IOException e) {
72            e.getMessage();
73        }
74        System.exit(0);
75    }
76 }
```

```
1 package gameforchildren;
2 import java.awt.*;
3 import java.io.IOException;
4 import java.io.RandomAccessFile;
5
6 public class Question extends EasyApp {
7
8    Label Subject = addLabel("Subject - " + Start.subject, 40, 50, 200, 40, this);
9    Label Value = addLabel("Value " + Start.value.substring(5), 40, 90, 200, 40, this);
10    Label questionN = addLabel("Question N" + (Start.count / 2), 40, 130, 200, 30, this);
11    Label question = addLabel("", 40, 160, 500, 30, this);
12    Label answerA = addLabel("1", 40, 190, 450, 30, this);
13    Label answerB = addLabel("2", 100, 230, 440, 30, this);
14    Label answerC = addLabel("3", 100, 270, 440, 30, this);
15    Label answerD = addLabel("4", 100, 310, 440, 30, this);
16    Label correctA = addLabel("Correct Answer: ", 150, 360, 130, 30, this);
17    TextField Answer = addTextField("", 290, 360, 70, 30, this);
18    Button check = addButton("check", 450, 360, 80, 40, this);
19
20    public Question() {
21       setTitle("Question");
22       setBounds(350, 150, 600, 430);
23       Subject.setForeground(Color.getHSBColor(0.9f, 0.8f, 0.6f));
24       Subject.setFont(new Font("Arial", 0, 20));
25       Value.setFont(new Font("Arial", 0, 20));
26       correctA.setFont(new Font("Arial", 0, 15));
27       questionN.setFont(new Font("Arial", 0, 15));
28       questionChoice();
29    }
30
31    public void actions(Object source, String command) {
32       if (source == check) {
33          check();
34       }
35    }
36
37    int i;
38    String subject = Start.subject;
39    String val = Start.value.substring(5); // it will be 1,2 or 3
40    int count = Start.count - 1;
41
42    void questionChoice() {
43 System.out.println("vaime = " + Start.countQuestions);
44       for (i = 0; i < Start.countQuestions; i++) {   //search method
45
46          if (Start.questionArray[i][8] == null && Start.questionArray[i][0].equals(subject) &&
                Start.questionArray[i][7].equals(val)) {
47             // appropreate question is found!
48             question.setText(Start.questionArray[i][1]);
49             answerA.setText("answers:   A) " + Start.questionArray[i][2]);
50             answerB.setText("B) " + Start.questionArray[i][3]);
51             answerC.setText("C) " + Start.questionArray[i][4]);
52             answerD.setText("D) " + Start.questionArray[i][5]);
53             Start.questionArray[i][8] = "Yes";
54             break;
```

```
55          }
56      }
57  }
58
59  static String point;  // how many point gain the team
60  void check() {   //is the answer correct or incorrect; if yes, hom many point the team gained
61      String answer = Answer.getText();
62      if (answer == null || !answer.equals("A") && !answer.equals("B") && !answer.equals("C")
            && !answer.equals("D")) {
63          outputString("Enter Correct Answer!\nTry again...");
64      } else {
65          if (answer.equals(Start.questionArray[i][6])) {
66              point = Start.questionArray[i][7];
67          } else {
68              point = "0";
69          }
70          outputString("You gained " + point + " point");
71          if (count % 2 == 0) {
72              Start.team2array[count / 2 - 1][0] = subject;
73              Start.team2array[count / 2 - 1][1] = val;
74              Start.team2array[count / 2 - 1][2] = point;
75          } else {
76              Start.team1array[count / 2][0] = subject;
77              Start.team1array[count / 2][1] = val;
78              Start.team1array[count / 2][2] = point;
79          }
80          dispose();
81      }
82  }
83 }
```