1. Create a student class that had fields name, surname, grades array with 5 grades, birthday. Create 7 objects of this class and print their data
2. Print student data by grade average
3. Print the data of those students whose name do not

contain duplicate letters and have at least 2 20 grades

the output should not be in the console but in a Frame textarea

Naye qnnutyany vor asum er nuyn kkrknvogh tar u meca 23 tarekanic

1. Main Class

```
public class Main {
  public static void main(String[] args) {
    new StudentWindow();
  }
}
```

2. MyDate Class

```
public class MyDate {
  private int year;
  private int month;
  private int day;

  public MyDate(int year, int month, int day) {
    this.year = year;
    this.month = month;
    this.day = day;
  }

  public int getYear() {
    return year;
  }

  public void setYear(int year) {
    this.year = year;
  }

  public int getMonth() {
    return month;
  }

  public void setMonth(int month) {
    this.month = month;
  }

  public int getDay() {
    return day;
  }

  public void setDay(int day) {
    this.day = day;
  }

  @Override
  public String toString() {
    return year + "/" + month + "/" + day;
  }
}
```

3. Student Class

```
import java.util.Arrays;

import java.util.Calendar;


public class Student {

  private String name;

  private String surname;

  private int[] grades;
  private MyDate birthday;

  public Student(String name, String surname, int[] grades, MyDate birthday) {
      this.name = name;
      this.surname = surname;
      this.grades = grades;
      this.birthday = birthday;
  }

  public String getName() {
      return name;
  }

  public void setName(String name) {
      this.name = name;
  }

  public String getSurname() {
      return surname;
  }

  public void setSurname(String surname) {
      this.surname = surname;
  }

  public int[] getGrades() {
```

```java
        return grades;
    }

    public void setGrades(int[] grades) {
        this.grades = grades;
    }

    public MyDate getBirthday() {
        return birthday;
    }

    public void setBirthday(MyDate birthday) {
        this.birthday = birthday;
    }

    @Override
    public String toString() {
        return name + " " + surname + " " + Arrays.toString(grades) + " " + birthday;
    }

    public double getAverageGrade() {
        double sum = 0;
        for (int grade : grades) {
            sum += grade;
        }
        return sum / grades.length;
    }

    public boolean hasRepeatingLetters() {
        boolean[] charMap = new boolean[26];
        String combinedName = (name + surname).toLowerCase();
        for (char c : combinedName.toCharArray()) {
            if (c >= 'a' && c <= 'z') {
                if (charMap[c - 'a']) {
                    return true;
                }
                charMap[c - 'a'] = true;
            }
        }
        return false;
    }

    public boolean isAtLeast23YearsOld() {
        Calendar today = Calendar.getInstance();
        int age = today.get(Calendar.YEAR) - birthday.getYear();
        if (today.get(Calendar.MONTH) + 1 < birthday.getMonth() ||
(today.get(Calendar.MONTH) + 1 == birthday.getMonth() &&
today.get(Calendar.DAY_OF_MONTH) < birthday.getDay())) {
            age--;
        }
        return age >= 23;
    }
```

```
}
```

4. StudentDAO Class

```
import java.util.Arrays;
import java.util.Comparator;
import java.util.List;
import java.util.stream.Collectors;

public class StudentDAO {
    private List<Student> students;

    public StudentDAO() {
        students = Arrays.asList(
            new Student("Alice", "Johnson", new int[]{18, 19, 20, 17, 16}, new MyDate(2000, 1, 1)),
            new Student("Bob", "Smith", new int[]{20, 15, 14, 20, 13}, new MyDate(1999, 2, 12)),
            new Student("Charlie", "Doe", new int[]{15, 14, 17, 20, 20}, new MyDate(2001, 3, 23)),
            new Student("Dave", "Brown", new int[]{18, 17, 15, 19, 16}, new MyDate(1998, 4, 15)),
            new Student("Eve", "Davis", new int[]{12, 13, 15, 17, 18}, new MyDate(2002, 5, 30)),
            new Student("Frank", "Miller", new int[]{20, 20, 18, 19, 17}, new MyDate(2001, 6, 10)),
            new Student("Grace", "Taylor", new int[]{10, 12, 15, 14, 13}, new MyDate(1997, 7, 21))
        );
    }

    public List<Student> getStudents() {
        return students;

    }


    public void sortByAverageGrade() {

        students.sort(Comparator.comparingDouble(Student::getAverageGrade));

    }


    public List<Student> getRepeatingLettersAndNotYoungerThan23() {

        return students.stream()

            .filter(Student::hasRepeatingLetters)

            .filter(Student::isAtLeast23YearsOld)

            .collect(Collectors.toList());

    }

}
```

5. StudentWindow Class

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.util.List;

public class StudentWindow extends JFrame implements ActionListener {
    private JButton showAll = new JButton("Show All Students");
    private JButton sortByAverage = new JButton("Sort by Average");
    private JButton repeatingLettersAndAge = new JButton("Repeating Letters and Age
>=23");
    private JTextArea studentInfo = new JTextArea();

    public StudentWindow() {
        setLayout(new FlowLayout());
    setSize(500, 500);
    setVisible(true);

    showAll.setSize(150, 50);
    sortByAverage.setSize(150, 50);
    repeatingLettersAndAge.setSize(150, 50);
    studentInfo.setPreferredSize(new Dimension(400, 400));
    add(showAll);
    add(sortByAverage);
    add(repeatingLettersAndAge);
    add(studentInfo);

    showAll.addActionListener(this);
    sortByAverage.addActionListener(this);
    repeatingLettersAndAge.addActionListener(this);
}

@Override
public void actionPerformed(ActionEvent e) {
    StudentDAO studentDAO = new StudentDAO();

    if (e.getSource() == showAll) {
        studentInfo.setText("");
        for (Student student : studentDAO.getStudents()) {
            studentInfo.append(student.toString() + "\n");
        }
    }

    if (e.getSource() == sortByAverage) {
        studentDAO.sortByAverageGrade();
        studentInfo.setText("");
        for (Student student : studentDAO.getStudents()) {
            studentInfo.append(student.toString() + "\n");
        }
    }

    if (e.getSource() == repeatingLettersAndAge) {
            List<Student> filteredStudents =
studentDAO.getRepeatingLettersAndNotYoungerThan23();
            studentInfo.setText("");
            for (Student student : filteredStudents) {
                studentInfo.append(student.toString() + "\n");
            }
        }
    }
}
```