Main.java
```java
public class Main {
    public static void main(String[] args) {
        new PersonsWindow();
    }
}
```
MyDate.java
```java
public class MyDate {
    private int year;
    private int month;
    private int day;

    public MyDate() {}

    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;
    }
}
```
Person.java
```java
public class Person {
    private String firstName;
    private String secondName;
    private int[] happyDays;
    private MyDate birthday;

    public Person() {}
    public Person(String firstName, String secondName, int[] happyDays, MyDate
birthday) {
        this.firstName = firstName;
        this.secondName = secondName;
        this.happyDays = happyDays;
        this.birthday = birthday;
    }

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getSecondName() {
        return secondName;
    }

    public void setSecondName(String secondName) {
        this.secondName = secondName;
    }

    public int[] getHappyDays() {
        return happyDays;
    }

    public void setHappyDays(int[] happyDays) {
        this.happyDays = happyDays;
    }

    public MyDate getBirthday() {
        return birthday;
    }

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

    @Override
    public String toString() {
        return firstName + " "
                + secondName +
                " " + Arrays.toString(happyDays) +
                " " + birthday + '\n';
    }

    public boolean happyIsEven() {
        int happySum = 0;
        for (int day : happyDays)
            happySum += day;
        if (happySum % 2 == 0)
            return true;
        return false;
        // return happySum % 2 == 0 ? true : false;
    }
}
```

```
PersonDAO.java
public class PersonDAO {
    private Person[] persons = new Person[3];

    public PersonDAO() {

    }

    public void addPersons() {
        Person person1 = new Person("Barry", "Smith", new int[]{5, 4, 20}, new
MyDate(2000, 12, 12));
        Person person2 = new Person("Jack", "Johnson", new int[]{10, 20, 22},
new MyDate(2001, 12, 21));
        Person person3 = new Person("Aaron", "James", new int[]{25, 26, 30}, new
MyDate(2003, 9, 9));

        persons[0] = person1;
        persons[1] = person2;
        persons[2] = person3;
    }

    public Person[] getPersons() {
        return persons;
    }

    public void sortBySecondName() {
        Person tmp = null;
        for (int i = 0; i < persons.length; i++) {
            for (int j = 0; j < persons.length - 1; j++) {
                if (persons[j].getSecondName().compareTo(persons[j +
1].getSecondName()) > 0) {
                    tmp = persons[j];
                    persons[j] = persons[j + 1];
                    persons[j + 1] = tmp;
                }
            }
        }
    }

    public Person[] evenHappyDays() {
        int happyCount = 0;
        for (Person person : persons)
            if (person.happyIsEven())
                ++happyCount;

        Person[] evenHappyPersons = new Person[happyCount];
        int index = 0;
        for (Person person : persons)
            if (person.happyIsEven())
                evenHappyPersons[index++] = person;

        return evenHappyPersons;
    }
}
PersonsWindows.java
public class PersonsWindow extends JFrame implements ActionListener {
    private JButton show = new JButton("Show persons");
    private JButton sort = new JButton("Sort by first name");
    private JButton evens = new JButton("Show happy evens");
```

```java
    private JTextArea personsInfo = new JTextArea();

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

        show.setSize(100,100);
        sort.setSize(100,100);
        evens.setSize(100,100);
        personsInfo.setPreferredSize(new Dimension(400, 400));
        add(show);
        add(sort);
        add(evens);
        add(personsInfo);

        show.addActionListener(this);
        sort.addActionListener(this);
        evens.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        PersonDAO personDAO = new PersonDAO();
        personDAO.addPersons();
        if (e.getSource() == show) {
            personsInfo.setText("");
            for (Person person : personDAO.getPersons()) {
                personsInfo.append(person.toString());
            }
        }
        if (e.getSource() == sort) {
            personDAO.sortBySecondName();
            personsInfo.setText("");
            for (Person person : personDAO.getPersons())
                personsInfo.append(person.toString());
        }
        if (e.getSource() == evens) {
            Person[] happyEvens = personDAO.evenHappyDays();
            personsInfo.setText("");
            for (Person person : happyEvens)
                personsInfo.append(person.toString());
        }
    }
}
```

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

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

public class Main {
    public static void main(String[] args) {
        SwingUtilities.invokeLater(PersonsWindow::new);
    }
}

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

    public MyDate() {}

    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;
    }
}

class Student {
    private String firstName;
    private String surname;
    private int[] grades;
    private MyDate birthday;

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

    public String getFirstName() {
        return firstName;
    }

    public String getSurname() {
        return surname;
    }

    public int[] getGrades() {
        return grades;
    }

    public MyDate getBirthday() {
        return birthday;
    }

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

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

    public boolean hasDuplicateLetters() {
        boolean[] charSet = new boolean[128];
        for (char c : firstName.toLowerCase().toCharArray()) {
            if (charSet[c]) return true;
            charSet[c] = true;
        }
        return false;
    }

    public boolean hasTwo20Grades() {
        int count20 = 0;
        for (int grade : grades) {
            if (grade == 20) count20++;
        }
        return count20 >= 2;
    }
}

class StudentDAO {
    private Student[] students = new Student[7];

    public StudentDAO() {
        students[0] = new Student("Barry", "Smith", new int[]{5, 4, 20, 15, 18},
new MyDate(2000, 12, 12));
        students[1] = new Student("Jack", "Johnson", new int[]{10, 20, 22, 18,
16}, new MyDate(2001, 12, 21));
        students[2] = new Student("Aaron", "James", new int[]{25, 26, 30, 28,
29}, new MyDate(2003, 9, 9));
        students[3] = new Student("Emma", "Watson", new int[]{20, 15, 22, 20,
18}, new MyDate(2000, 5, 15));
        students[4] = new Student("Oliver", "Taylor", new int[]{20, 20, 20, 20,
20}, new MyDate(2002, 8, 23));
        students[5] = new Student("Sophia", "Brown", new int[]{15, 25, 18, 20,
22}, new MyDate(2001, 3, 8));
        students[6] = new Student("Liam", "Miller", new int[]{20, 20, 20, 20,
20}, new MyDate(2002, 10, 30));
    }

    public Student[] getStudents() {
        return students;
    }

    public Student[] sortByGradeAverageDescending() {
        Arrays.sort(students, (s1, s2) -> Double.compare(s2.getGradeAverage(),
s1.getGradeAverage()));
        return students;
    }

    public Student[] filterByUniqueFirstNameAndTwo20Grades() {
        int count = 0;
        for (Student student : students) {
            if (!student.hasDuplicateLetters() && student.hasTwo20Grades()) {
                count++;
            }
        }
        Student[] filteredStudents = new Student[count];
        int index = 0;
        for (Student student : students) {
            if (!student.hasDuplicateLetters() && student.hasTwo20Grades()) {
                filteredStudents[index++] = student;
            }
        }
        return filteredStudents;
    }
}

class PersonsWindow extends JFrame implements ActionListener {
    private JButton show = new JButton("Show students");
    private JButton sort = new JButton("Sort by grade average");
    private JButton filter = new JButton("Filter by conditions");
    private JTextArea studentsInfo = new JTextArea();

    public PersonsWindow() {
        setTitle("Student Information");
        setLayout(new FlowLayout());
        setSize(500, 500);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        studentsInfo.setPreferredSize(new Dimension(400, 400));
        add(show);
        add(sort);
        add(filter);
        add(new JScrollPane(studentsInfo));
        show.addActionListener(this);
        sort.addActionListener(this);
        filter.addActionListener(this);

        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        StudentDAO studentDAO = new StudentDAO();
        if (e.getSource() == show) {
            studentsInfo.setText("");
            for (Student student : studentDAO.getStudents()) {
                studentsInfo.append(student.toString() + "\n");
            }
        }
        if (e.getSource() == sort) {
            Student[] sortedStudents =
studentDAO.sortByGradeAverageDescending();
            studentsInfo.setText("");
            for (Student student : sortedStudents) {
                studentsInfo.append(student.toString() + "\n");
            }
        }
        if (e.getSource() == filter) {
            Student[] filteredStudents =
studentDAO.filterByUniqueFirstNameAndTwo20Grades();
            studentsInfo.setText("");
            for (Student student : filteredStudents) {
                studentsInfo.append(student.toString() + "\n");
            }
        }
    }
}
```

1. create Student class with fields name, surname, grades[], birthday

2. sort them in descending order by average grade

3. print the students whose grades contain at least one prime number, and the students whose names are palindrome

print with JFrame

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

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

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

    public MyDate() {}
```

```java
    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;
    }
}

class Student {
    private String firstName;
    private String surname;
    private int[] grades;
    private MyDate birthday;

    public Student() {}

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

    public String getFirstName() {
        return firstName;
    }

    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }

    public String getSurname() {
        return surname;
    }

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

    public int[] getGrades() {
        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 firstName + " " + surname + " " + Arrays.toString(grades) + " " +
birthday + '\n';
    }

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

    public boolean containsPrime() {
        for (int grade : grades) {
            if (isPrime(grade)) {
                return true;
            }
        }
        return false;
    }

    private boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(n); i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }

    public boolean isPalindrome() {
        String name = firstName.toLowerCase();
        int i = 0, j = name.length() - 1;
        while (i < j) {
            if (name.charAt(i) != name.charAt(j)) {
                return false;
            }
            i++;
            j--;
        }
        return true;
    }
}

class StudentDAO {
    private Student[] students = new Student[7];

    public StudentDAO() {}

    public void addStudents() {
        students[0] = new Student("John", "Doe", new int[]{90, 85, 88, 92, 95},
new MyDate(2000, 5, 15));
        students[1] = new Student("Alice", "Smith", new int[]{75, 82, 79, 88,
90}, new MyDate(2001, 7, 20));
        students[2] = new Student("Bob", "Johnson", new int[]{85, 90, 87, 92,
88}, new MyDate(2000, 9, 10));
        students[3] = new Student("Anna", "Taylor", new int[]{92, 88, 90, 94,
91}, new MyDate(2002, 3, 25));
        students[4] = new Student("Eva", "Brown", new int[]{80, 85, 82, 78, 86},
new MyDate(2001, 11, 5));
        students[5] = new Student("David", "Wilson", new int[]{88, 84, 90, 92,
87}, new MyDate(2003, 8, 15));
        students[6] = new Student("Hannah", "Clark", new int[]{94, 91, 95, 92,
90}, new MyDate(2000, 2, 8));
    }

    public Student[] getStudents() {
        return students;
    }

    public void sortByGradeAverage() {
        Arrays.sort(students, (s1, s2) -> Double.compare(s2.getGradeAverage(),
s1.getGradeAverage()));
    }

    public Student[] studentsWithPrimeGrades() {
        int count = 0;
        for (Student student : students) {
            if (student.containsPrime()) {
                count++;
            }
        }
        Student[] studentsWithPrime = new Student[count];
        int index = 0;
        for (Student student : students) {
            if (student.containsPrime()) {
                studentsWithPrime[index++] = student;
            }
        }
        return studentsWithPrime;
    }

    public Student[] palindromeFirstNames() {
        int count = 0;
        for (Student student : students) {
            if (student.isPalindrome()) {
                count++;
            }
        }
        Student[] palindromeStudents = new Student[count];
        int index = 0;
        for (Student student : students) {
            if (student.isPalindrome()) {
                palindromeStudents[index++] = student;
            }
        }
        return palindromeStudents;
    }
}

class StudentsWindow extends JFrame implements ActionListener {
    private JButton show = new JButton("Show students");
    private JButton sort = new JButton("Sort by grade average");
    private JButton primes = new JButton("Show students with prime grades");
    private JButton palindromes = new JButton("Show students with palindrome
first names");
    private JTextArea studentsInfo = new JTextArea();

    public StudentsWindow() {
        setLayout(new FlowLayout());
        setSize(600, 400);
        setVisible(true);

        show.setPreferredSize(new Dimension(200, 50));
        sort.setPreferredSize(new Dimension(200, 50));
        primes.setPreferredSize(new Dimension(200, 50));
        palindromes.setPreferredSize(new Dimension(200, 50));
        studentsInfo.setPreferredSize(new Dimension(580, 250));
        add(show);
        add(sort);
        add(primes);
        add(palindromes);
        add(studentsInfo);

        show.addActionListener(this);
        sort.addActionListener(this);
        primes.addActionListener(this);
        palindromes.addActionListener(this);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        StudentDAO studentDAO = new StudentDAO();
        studentDAO.addStudents();
        if (e.getSource() == show) {
            studentsInfo.setText("");
            for (Student student : studentDAO.getStudents()) {
                studentsInfo.append(student.toString());
            }
        } else if (e.getSource() == sort) {
            studentDAO.sortByGradeAverage();
            studentsInfo.setText("");
            for (Student student : studentDAO.getStudents()) {
                studentsInfo.append(student.toString());
            }
        } else if (e.getSource() == primes) {
            Student[] primeStudents = studentDAO.studentsWithPrimeGrades();
            studentsInfo.setText("");
            for (Student student : primeStudents) {
                studentsInfo.append(student.toString());
            }
        } else if (e.getSource() == palindromes) {
            Student[] palindromeStudents = studentDAO.palindromeFirstNames();
            studentsInfo.setText("");
            for (Student student : palindromeStudents) {
                studentsInfo.append(student.toString());
            }
        }
    }
}
```