Programmieren und Software-Engineering II
Übung 10

Name: _____ Klasse: _____ Datum: _____

Aufgabe 1: Schule
Für die **Notenverwaltung** an einer **Schule** soll ein Programm erstellt werden.
Die folgenden Klassen werden dazu benötigt:

Examination
course: String mark: int
toString(): String …

Student
name: String age: int examList: Examination[]
toString(): String addExam (Examination e): void getAverageMark(): double getAverageMark(String s): double printSchoolReport():void deleteExam(String s):void getBestMark():Examination[] sortExamList():void …

Classroom
name: String MAX_Student: static final int studentList: Student[]
toString(): String addStudent (Student: s): void getClassroomAverage ():double getClassroomAverage (String s):double sortListByName(): void sortListByAverageMark (): void sortListByAverageMark (String s): void printStudentList(): void getBestSudent():Student getBestSudent(String s):Student deleteStudent(Student: s): void …

Der Ablauf sollte folgendermaßen funktionieren:

- Zuerst werden die gewünschten Schüler angelegt (d.h. im HP werden **neue Students erzeugt**).
- Dann legen die Schüler mehrere Prüfungen ab (**addExam ()).**
- Es wird ein Zeugnis für die Schüler ausgedruckt (**printSchoolReport()**). Dabei werden alle Noten in einem Fach addiert und eine Durchschnittnote berechnet (ganzzahlig). Jedes Fach wird nur einmal angedruckt.
- Anschließend werden die Schüler der Klasse zugeordnet (**addStudent()**).
- Zum Schluss wird eine Klassenliste ausgedruckt (**printStudentList()**).

Das **Hauptprogramm** besitzt folgenden **Aufbau:**

```
public static void main(String[] args){
    Classroom c1=new Classroom("2BHIF");

     Examination e1=new Examination("Math",2);
     Examination e2=new Examination("Java",1);
     Examination e3=new Examination("Biology",3);
     Examination e4=new Examination("Linux",3);
```

```
        Student s1=new Student("Bart",16);
        s1.addExam(e1);
        s1.addExam(e2);
        s1.addExam(e3);
        s1.addExam(e4);
        System.out.println(s1);          // Bart : 16 : 2.25
        s1.printSchoolReport();          // Zeugnis drucken mit allen Fächern
        s1.sortExamList();               // alphabetisch Sortieren nach den Gegenständen
        s1.printSchoolReport();

        e1=new Examination("Math",3);
        e2=new Examination("Java",1);
        e3=new Examination("Latin",4);
        Student s2=new Student("Waylon",36);
        s2.addExam(e1);
        s2.addExam(e2);
        s2.addExam(e3);
        s2.deleteExam("Latin");
        s2.printSchoolReport();          // Zeugnis drucken mit allen Fächern

        e1=new Examination("Java",2);
        e2=new Examination("Linux",2);
        e3=new Examination("Biology",1);
        Student s3=new Student("Lisa",12);
        s3.addExam(e1);
        s3.addExam(e2);
        s3.addExam(e3);
        s3.printSchoolReport();          // Zeugnis drucken mit allen Fächern

        e1=new Examination("Java",5);
        e2=new Examination("Linux",4);
        e3=new Examination("Biology",4);
        Student s4=new Student("montgomery",93);
        s4.addExam(e1);
        s4.addExam(e2);
        s4.addExam(e3);
        s4.printSchoolReport(); // Zeugnis drucken mit allen Fächern

        c1.addStudent(s1);
        c1.addStudent(s2);
        c1.addStudent(s3);
        c1.addStudent(s4);

        System.out.println("\nListe");
        c1.printStudentList();
        System.out.println("\nSortiert nach Namen");
        c1.sortListByName();
        c1.printStudentList();
        System.out.println("\nSortiert nach   Notendurchschnitt");
        c1.sortListByAverageMark();
        c1.printStudentList();

        Student best=c1.getBestStudent();
        System.out.println("\nBester Student: "+best);   // lisa : 12 : 1.6666666666666667

        //…
}
```

Programmieren und Software-Engineering II
Übung 10

Aufgabe 2: Polynom
Ein Polynom lässt sich vollständig durch seinen Grad und durch ein Feld mit den Koeffizienten beschreiben. Zum Beispiel ist das Polynom $x^4 + 3x^2 + 2$ vom Grad 4, und der Vektor (1,0,3,0,2) beschreibt, welche Koeffizienten verwendet werden. Dies soll im Folgenden ausgenutzt werden, um eine Klasse Polynom zu entwerfen und zu implementieren. Dabei sollen nur Polynome mit ganzzahligen Koeffizienten betrachtet werden.

Besondere Polynome:

- Grad 1: $P(x) = 3x + 5$ // lineare Funktion
- Grad 2: $P(x) = -3x^2 - 2x + 1$ // quadratische Funktion
- Grad 3: $P(x) = 5x^3 - 3x^2 + 6x - 2$ // kubische Funktion
- Grad 4: $P(x) = 2x^4 - x^3 + 3x^2 + 2x - 1$ // quartische Funktion

Erstelle ein Java-Programm, welches wichtige Funktionalitäten eines Polynoms zur Verfügung stellt. In weiterer Folge könnte auch eine Methode zur graphischen Darstellung eines Polynoms implementiert werden.

Polynom
coeff: int []
add(p: Polynom): void calc(x: double): double getDegree():int toString(): String multiply(int c): void multiply(p: Polynom): void draw(a: int, b: int):void …

Das **Hauptprogramm** besitzt folgenden **Aufbau:**

```
public static void main(String[] args){
      int[] k1={1, 0, 3, 0, 2};
      int[] k2={5, 2, -3};
      Polynom p1=new Polynom (k1);
     Polynom s2=new Polynom (k2);

     System.out.println(p1);                          // x^4 + 3x^2 + 2
     System.out.println("p1(3) = "+p1.calc(3));       // p1(3) = 110
     System.out.println("p1(2) = "+p1.calc(2));       // p1(2) = 30

     System.out.println(p2);                          // 5x^2 + 2x – 3

      p1.add(p2);
      System.out.println(p1);                         // x^4 + 8x^2 + 2x – 1

      p1.multiply(2);
      System.out.println(p1);                         // 2x^4 + 16x^2 + 4x – 2

      p1.multiply(p2);
      System.out.println(p1);
```

Aufgabe 3: Punkt / Dreieck
Implementiere eine geeignete Klasse, um Punkte in der kartesischen Koordinatenebene zu repräsentieren.

Es soll folgende Methoden geben:

- Point (double x, double y) // Konstruktor
- double distance (Point p) // Liefert Abstand zum Punkt p
- boolean isSame (Point p) // gibt Auskunft, ob der Punkt mit p zusammenfällt
- Point moved (double dx, double dy) // Liefert einen neuen Punkt, der vertikal und horizontal verschoben wurde
- …

Implementiere zusätzlich eine geeignete Klasse, um Dreiecke in der kartesischen Koordinatenebene zu repräsentieren.

Programmieren und Software-Engineering II
Übung 10

Es soll folgende Methoden geben:

- Triangle (Point a, Point b, Point c) // Konstruktor
- double perimeter() // liefert Umfang des Dreiecks
- double area() // liefert Fläche des Dreiecks
- Point lowerLeft() // liefert linke untere Ecke des umschreibenden Rechtecks
- double upperRight() // liefert rechte obere Ecke des umschreibenden Rechtecks
- Triangle moved(double dx, double dy) // erzeugt ein um dx bzw. dy verschobenes Dreieck
- boolean isSame(Triangle t) // gibt Auskunft, ob dieses Dreieck mit dem Dreieck t übereinstimmt
- …