Matrikelnummer:	*Punkte*:

Testklausur 1 zur Vorlesung

Modellierung und Programmierung I

Dr. Monika Meiler
Zeit: 60 Minuten

Bemerkungen:

- **Jedes Blatt ist mit der Matrikelnummer zu versehen.**
- **Jede Aufgabe ist auf dem vorgesehenen Blatt zu lösen. Reicht der dortige Platz nicht aus, so verwenden Sie ein mit der Matrikelnummer versehenes zusätzliches Blatt.**
- **Es sind außer Papier und Schreibzeug *keine* weiteren Hilfsmittel erlaubt (keine Taschenrechner, keine Unterlagen, . . .).**
- **Es ist leserlich und *nicht* mit Bleistift zu schreiben.**
- **Beantworten Sie Fragen pro Pfeil „➢" mit *genau* einem Sachverhalt.**

Schalten Sie Ihr Handy aus!

Punkte	Note
0	5
25	4
35	3
45	2
55	1

Matrikelnummer:	*Punkte*:

Klausuraufgabe 1. (16 Punkte)

Zahlendarstellung

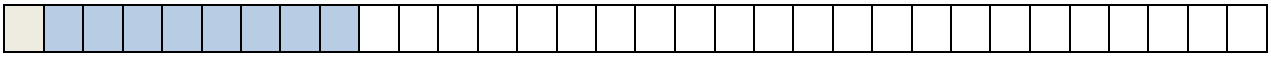
Für *Gleitpunktzahlen* wird in der Programmiersprache Java der IEEE-Code* zur Zahlendarstellung verwendet. Die kleinste positive Maschinenzahl $x_{\min}$ und die größte positive Maschinenzahl $x_{\max}$ vom Elementardatentyp **`float`** sind:

$x_{\min} = 2^r = 2^{-126}$

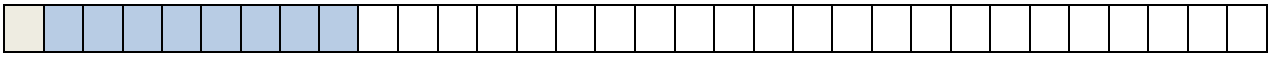
$x_{\max} = (1.1111'1111'1111'1111'1111'111)_2 * 2^R = (1.1111'1111'1111'1111'1111'111)_2 * 2^{127}$

(a) Stellen Sie $x_{\min}$ und die nächstgrößere Maschinenzahl $x_{\min+1}$ jeweils als Bitfolge dar und berechnen Sie aus den Bitdarstellungen den Abstand $d_{\min} = |x_{\min+1} - x_{\min}|$ zwischen beiden benachbarten Zahlen als Zweierpotenz. **6P**

➢ $x_{\min}$



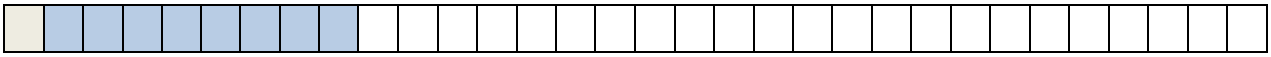
➢ $x_{\min+1}$



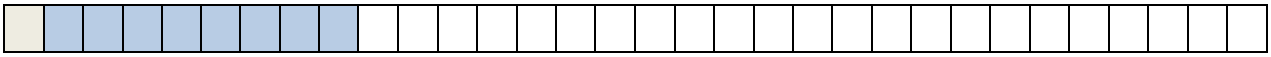
➢ $d_{\min} = |x_{\min+1} - x_{\min}| =$

(b) Stellen Sie $x_{\max}$ und die nächstkleinere Maschinenzahl $x_{\max-1}$ jeweils als Bitfolge dar und ermitteln Sie aus den Bitdarstellungen den Abstand $d_{\max} = |x_{\max} - x_{\max-1}|$ zwischen beiden benachbarten Zahlen als Zweierpotenz. **6P**

➢ $x_{\max}$



➢ $x_{\max-1}$



➢ $d_{\max} = |x_{\max} - x_{\max-1}| =$

* siehe Rückseite

(c) Berechnen Sie als Zweierpotenzen jeweils den Rundungsfehler im ungünstigsten Fall für die Darstellung von Zahlen $z \in [x_{min}; x_{min+1}]$ und für die Darstellung von Zahlen $z \in [x_{max-1}; x_{max}]$ als Maschinenzahlen vom Elementardatentyp **`float`**. **2P**

➢ $z \in [x_{min}; x_{min+1}]$**:**

➢ $z \in [x_{max-1}; x_{max}]$**:**

(d) Bewerten Sie das Ergebnis unter Berücksichtigung des negativen Zahlenbereichs. Welche Aussage kann man allgemein über die Dichte von Maschinenzahlen eines Gleitpunktdatentyps machen? **2P**

➢

IEEE – *Standard, Institute of Electrical and Electronics Engineers*

Bezeichnung	Byteanzahl	*s* [Bit]	*M* [Bit]	*E* [Bit]	*r*	*R*	*C*	Java
`single`	4	1	24	8	-126	127	127	`float`
`double`	8	1	53	11	-1022	1023	1023	`double`

s Vorzeichenbit: 0 ≙ +, 1 ≙ -
M Länge der Mantisse, einschließlich *hidden bit*
E Anzahl der Exponentenbit
r kleinster Exponent
R größter Exponent
C Verschiebungskonstante

(Maschinenzahl: ***s*** | ***E'*** | ***M'*** mit ***E'=E+C*** und ***M*** ≈ **1.*M'***)

Matrikelnummer:	*Punkte*:

Klausuraufgabe 2. (14 Punkte)

Methoden

Gegeben sei die folgende Klassenmethode zur Berechnung der Funktion $\sin(x)$ mittels Reihenentwicklung auf Rechnergenauigkeit:

```
public static double mysin( double x)
{
  double summeNeu, summeAlt, summand;
  int i = 1;

  summand = summeNeu = x;                          // erstes Reihenglied
  do                                 // Berechnung weiterer Reihenglieder
  {
    summeAlt = summeNeu;
    i++; summand = - summand * x * x / i;
    i++; summand = summand / i;
    summeNeu = summeAlt + summand;
  }
  while( summeNeu != summeAlt);

  return summeNeu;
}
```

(a) In der Variablen **`summand`** wird das erste Reihenglied der Sinusreihe mit dem Wert von ***x*** festgelegt. Jeder Schleifendurchlauf berechnet in der Variable **`summand`** ein weiteres Reihenglied. Geben Sie die nächsten drei Reihenglieder formelmäßig an. **3P**

- x, , , .

(b) Ergänzen Sie die im Programm zur Berechnung der Funktion $\sin(x)$ verwendete mathematische Summenformel für die Reihenentwicklung. **3P**

- $\sin(x) = \sum$

(c) Für sehr große Beträge von *x* werden die Ergebnisse auf Grund der geringen Zahlendichte in ihrer Darstellung unbrauchbar: **8P**

Sin.out

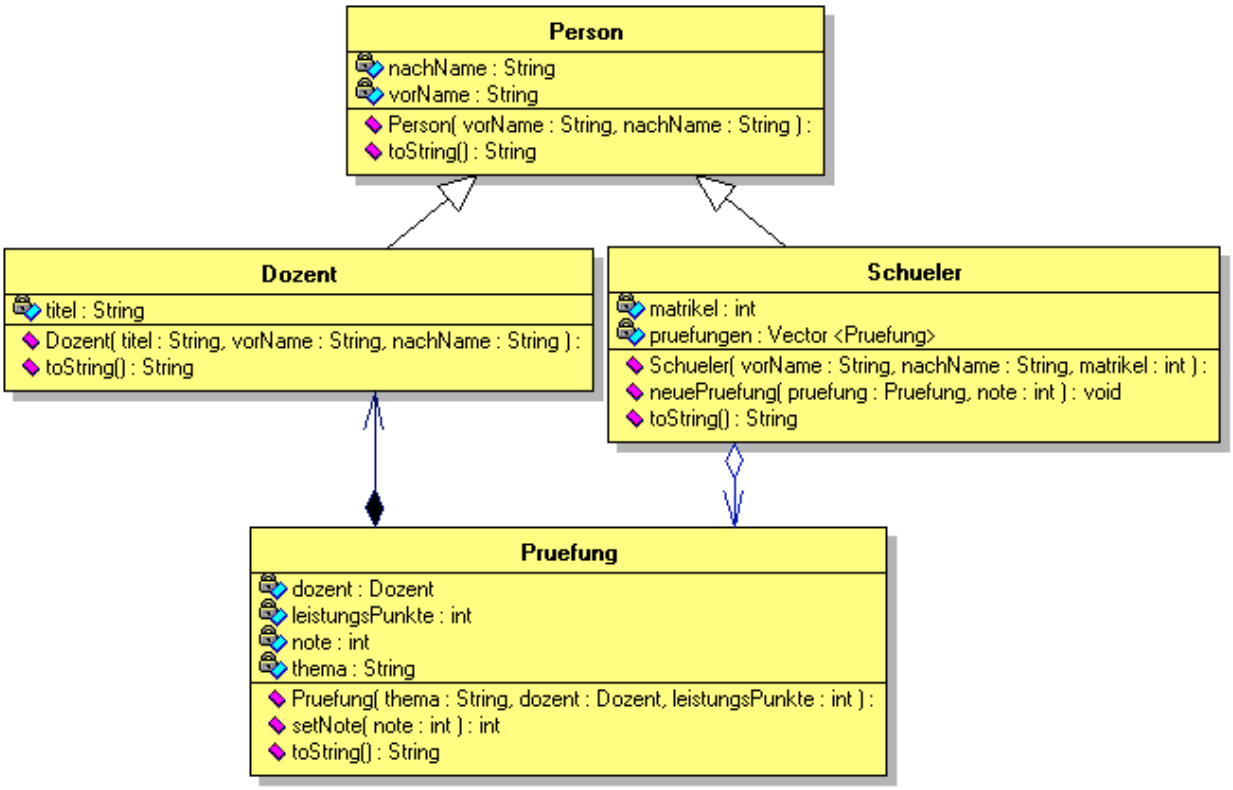
x	mysin	sin
0.523599	0.500000	0.500000
3.665191	-0.500000	-0.500000
41.364303	-1.904822	-0.500000
44.505896	147.240652	0.500000
47.647489	405.883710	-0.500000
50.789081	97904.919410	0.500000

Schreiben Sie eine weitere Klassenmethode ***mysinBesser***, welche zunächst den Parameter auf das dichtere Intervall $x \in [0, \pi]$ transformiert und anschließend die Methode ***mysin*** für den transformierten Wert aufruft (π: `Math.PI`).

```
public static double mysinBesser( double x)
{









}
```

Matrikelnummer:	*Punkte*:

Klausuraufgabe 3. (20 Punkte)



Implementieren Sie zu dem obigen Klassendiagramm:

(a) Geben Sie, ausgehend von dem folgenden Konstruktor der Klasse `Person`, den Konstruktor der Klasse `Schueler` an. **3P**

Person.java

```
public Person( String vorName, String nachName)
{
   this.vorName = vorName;
   this.nachName = nachName;
}
```

Schueler.java

```
public Schueler( String vorName, String nachName, int matrikel)
{









}
```

(b) Gegeben seien die folgenden `toString`-Methoden: **2P**

Person.java
```
public String toString()
{
   return vorName + " " + nachName;
}
```
Dozent.java
```
public String toString()
{
   return titel + " " + super.toString();
}
```
Pruefung.java
```
public String toString()
{
   return dozent.toString() + ": " + thema
    + "(" + leistungsPunkte + ") Note " + note;
}
```

Welcher String wird für eine Prüfung des ***Prof. Severus Snape*** zum Thema ***Zaubertränke*** mit ***5*** Leistungspunkten und einer Prüfungsnote ***3*** zurückgegeben (Verwenden Sie das Zeichen ␣ als Leerzeichen)?

➢

(c) Überschreiben Sie die `toString`-Methode der Klasse `Schueler` so, dass für jeden Schüler ein String mit dem vollständigen Namen, der Matrikelnummer und allen seinen Prüfungsergebnissen erzeugt wird*. **5P**

Schueler.java
```
public String toString()
{




}
```

* Einige Methoden der Klasse `Vector`: s. Rückseite

Matrikelnummer:	*Punkte*:

(d) Schreiben Sie ein Testprogramm `main` mit dem folgenden Inhalt: **10P**

Harry Potter (Matrikelnummer: ***4334***) hat die Prüfung ***Verteidigung gegen dunkle Künste*** (***10*** Leistungspunkte) beim Dozenten ***Prof. Quirin Quirrell*** mit der Note ***2*** erfolgreich abgeschlossen.
Erzeugen Sie auf der Konsole mittels Standardausgabe Harrys Prüfungsstand.

Schueler.java

```
public static void main( String[] args)
{









}
```

Einige Methoden:

Klasse `java.util.Vector<E>`

Name	Parameter-anzahl	Parameter-typ	Ergebnis-Typ	Beschreibung
`add`	1	`E`	`boolean`	fügt Objekt am Ende ein
`get`	1	`int`	`E`	liefert Objekt an Position
`size`	0		`int`	liefert Anzahl der Objekte

Klausuraufgabe 4. (10 Punkte)

Erweitern Sie die UML-Klassenhierarchie:

Modellieren Sie als UML-Klassenhierarchie (ohne Funktionalitäten) eine Internatsschule bestehend aus mehreren Häusern. **10P**

- ⇨ Eine *Schule* verfügt über einen Schulnamen, einen Schuldirektor, einen Stellvertreter, eine Lehrerschaft und mehreren Schulhäuser.
- ⇨ Jedes *Schulhaus* besitzt einen Hausnamen, einen Hauslehrer und Schüler.
- ⇨ Schuldirektor, Stellvertreter und Lehrerschaft sind in der Schule als Dozenten tätig.

Erweitern Sie das obige hier etwas vereinfachte UML-Diagramm um die neuen Klassen. Achten Sie dabei auf kleine, übersichtliche Klassen und Wiederverwendbarkeit dieser bei ähnlichen Problemen. Geben Sie die Instanzvariablen, Datentypen und Beziehungen der verwendeten Klassen an.