



PROBEKLAUSUR

**Grundlagen der Informatik (PR1)
im Studiengang Bachelor Medizininformatik**

Ich wünsche Ihnen viel Erfolg!

Name, Vorname (in Druckbuchstaben): _____

Matrikelnummer: _____ Unterschrift: _____

WICHTIGE HINWEISE !

- Legen Sie einen gültigen Studentenausweis auf den leeren Platz neben sich.
- Schreiben Sie mit Füller (keine rote Tinte) oder mit Kugelschreiber.
- Schreiben Sie die Lösung zu einer Aufgabe auf das vorgesehene Blatt. Sollte der Platz nicht reichen, benutzen Sie eins der zusätzlichen leeren Blätter am Ende der Klausur. Geben Sie unbedingt einen Verweis auf dieses Blatt an und kennzeichnen Sie dieses Blatt als zu der entsprechenden Aufgabe gehörig.
- Die Klausur ist nicht in einzelne Blätter zu zerlegen. Lose Blätter werden nicht gewertet.
- Jeder Täuschungsversuch wird mit 0 Punkten = Note 5.0 gewertet.

Insgesamt können **90** Punkte erreicht werden. Die Bearbeitungszeit beträgt **90** Minuten.

Gesamtpunkte: _____

Note: _____

A01

Ein Klasse...

- ist_eine_Instance_von_einem_Objekt
- definiert_die_Methoden_eines_Objekts
- ist_immer_statisch
- ist_ein_Objekt
- kann_niemals_instanziert_werden

A02

Was ist ein Attribut?

- Eine_Attribut_ist_ein_Interface
- Eine_Attribut_ist_eine_Art_von_Klasse
- Eine_Attribut_ist_ein_Methode
- Eine_Attribut_ist_eine_Funktion
- Eine_Attribut_ist_eine_Variable

A03

Eine statische Variable

- existiert schon, bevor ein Objekt existiert
- kann aufgeladen werden
- existiert erst, wenn ein Objekt existiert
- ist immer von einem elementaren Datentyp

A04

Welches der folgenden Datentypen ist ein elementarer Datentyp?

- Matrix
- Integer
- long
- String
- fixedpoint

A05

Schreiben Sie eine For-Schleife, die die folgenden Zahlen mittels System.out.println() ausgibt:: 1,3,5,7,9,11,13,15,17

```
for (int i=1;i<18;i+=2){
    System.out.println(i+",");
}
```

A06 (7 pts)

Schreiben Sie eine do-while-Schleife, die die `check(int n)` Funktion mit folgenden Zahlen aufruft: `-9,-10,-11,...,-88`

```
i=-9;
do{
    check(i--);
}while(i>-89);
```

A07

Schreiben Sie eine while-Schleife, die die `check(int n)` Funktion mit folgenden Zahlen aufruft: `11,-9,7,-5,3,-1`

```
int i=11;
int vorz = 1;
while(i>=1){
    check(i*vorz);
    vorz=-vorz;
    i-=2;
}
```

A08

Gegeben: `i=5;j=7; x=i++ + ++j`; Welchen Wert nimmt `x` an?

- 5
- 6
- 9
- 10
- 11
- 12
- 13
- 14
- 15

A09

Eine Klasse `Konto` soll für die Membervariable `betrag` des Typs `float` eine Setter und Getter Methode bekommen. Ausserdem soll der Betrag beim Instanzieren der Klasse auf 0 gesetzt werden.

```
class Konto
{
    private float betrag;
    public Konto(){
        betrag = 0;
    }
    public void setBetrag(float x){
        betrag = x;
    }
    public float getBetrag(){
        return betrag;
    }
}
```

A10

Schreiben Sie eine Methode `isTriangle(int alpha, int beta, int gamma)`, die prüft, ob die Summe der übergebenen Winkel 180 Grad ist. Ist das der Fall, so soll die Funktion `true` zurückgeben, andernfalls `false`.

```
boolean isTriangle(int alpha, int beta, int gamma)
    if (alpha+beta+gamma==180)
        return true;
    return false;
}
```

A11

Programmieren Sie eine Methode `Recursive` mit einer Methode `recursive(int x)`, die sich selbst `x` mal aufruft. Das sieht dann so aus, dassz.B. `recursive(3) → recursive(2) → recursive(1)` aufruft und dann aus der Methode jeweils zurückkehrt.

```
public void recursive(int x){
    if (x>1)
        recursive(x-1);
    else
        return;
}
```

A12)

Vervollständigen Sie die folgende Funktion. Sie soll den kleinsten Wert der ersten `n` Elemente des übergebenen Arrays zurückgeben. Dabei soll KEIN Sortieralgorithmus verwendet werden!

```
int getLowestValue(int myArray[], int n){
    int lowest = myArray[0];
    for (int t=0;t<n && t<myArray.length;t++){
        if (myArray[t]<lowest){
            lowest = myArray[t];
        }
    }
    return lowest;
}
```

A13)

Programmieren Sie eine Methode `boolean isSymetric(int x)`, welche überprüft, ob die Zahl `x` von hinten gelesen genauso lautet wie von vorne gelesen. Ist das so, dann sollte `true` zurück gegeben werden, andernfalls `false`.

```
boolean isSymmetric(int x){
    int arr[] = new int[100];
    int index = 0;
    do{
        arr[index++] = x%10;
        x/=10;
    }while(x>0);
    for (int i=0;i<index;i++){
        if (arr[i]!=arr[index-i-1]){
            return false;
        }
    }
    return true;
}
```

A14

Leiten Sie die Klasse `Volkswagen` von der Klasse `Auto` ab. Die Abstrakte Methode `double getAnzahlReifen()` soll in der abgeleiteten Klasse implementiert werden und immer 4 zurückgeben.

```
abstract class Auto{
    abstract double getAnzahlReifen();
}
class Volkswagen extends Auto{
    double getAnzahlReifen(){
        return 4;
    }
}
```