Die folgenden Fragen beziehen sich auf die Klasse Konto aus dem Unterricht:

```
public class Konto
{
    // Attribute (Membervariablen)
    private int kontonummer;
    private String inhaber;
    private double kontostand;

    …….
    …….
}
```

Frage 1:

Codieren Sie die Anweisung zum Erzeugen eines Objekts vom Typ Konto, dessen Member-variablen folgenden Inhalt haben sollen: Konto k1 = new Konto(1122,“Mueller“,1289.65)

kontonummer ← 1122
inhaber ← Mueller
kontostand ← 1289.65

Frage 2:

Welche Methoden gehören üblicherweise standardmäßig zur Klasse Konto?
Akzessoren (Setter und Getter), Konstruktoren, to.String()

Frage 3:

Sie haben (z.B. in einer main()-Methode) ein Objekt k1 vom Typ Koto erzeugt. Wie können Sie

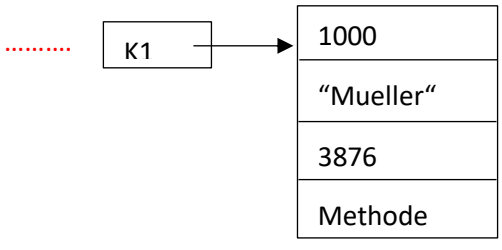
Über . Operator Direkter Zugriff von außen nicht möglich, nur immer über Akzessoren.

a) auf die Attribute des Objekts k1 zugreifen? String s = K1.getinhaber();
b) auf die Methoden des Objekts zugreifen? System.out.println(k1.toString);

Geben Sie jeweils ein Beispiel an!

Frage 4:

Codieren Sie ein Programmstück (z.B. Auszug aus einer main-Methode,), das 3 Objekte k1, k2 und k3 vom Typ Konto erzeugt. Überlegen Sie sich passende Werte für die Attribute und zeichnen Sie die Objekte im Hauptspeicher! Konto k1 = new Konto(1000,“Mueller“,,3876);



Frage 5:

Nennen Sie drei Prinzipien der objektorientierten Softwareentwicklung!
- Vererbung - Verkapselung - Abbildung der realen Welt - Enge verknüpf. von Daten u. Method.

Frage 6:

Die Klasse Konto wird um ein neues Attribut erweitert:

private static double zinssatz;

Warum bezeichnet man das Attribut **zinssatz** als Klassenattribut und was ist damit gemeint?
- Es ist der Klasse zugeordnet und Existiert nur ein einizges mal (zugriff Konto.Zinssatz)

Frage 7:

Nachdem die Klasse Konto um das Attribut zinssatz erweitert wurde, werden (z.B. in einer main()-Methode) zwei Objekte k1 und k2 vom Typ Konto angelegt. Zeichnen Sie jetzt die die Objekte K1 und k2 im Hauptspeicher! Ist immernoch wie in Frage 4, da das Attribut Zinssatz an einer anderen Adresse im Hauptspeicher steht.

Frage 8:

Codieren Sie einen Copy-Konstruktor für die Klasse Konto!
public Konto(Konto p) { this.kontonummer = p.kontonummer; this.kontostand = p.kontostand;

Frage 9: this.kontoinhaber = p.kontoinhaber;}

Erläuterm Sie die Begriffe **Überladen** von Methoden und **Überschreiben** einer Methode! Enthält die Klasse Konto (Methoden, die standardmäßig vorhanden sein sollten, sind codiert) Beispiele für diese Begriffe? - Überladen mehrere Methoden mit selben namen aber verschied. paramter.

Frage 10: - Überschreiben @override eine vererbte Methode wird ersetzt nicht gelöscht

Klasse **Kont**o ist die Elternklasse der Kindklasse **Konto_Tagesgeld**. Welche Attribute und Methoden enthält die Kindklasse? - Alle der Elternklasse (vererbt) + die eigenen

Frage 11:

Können die Methoden der Kindklasse **Konto_Tagesgeld** direkt auf die von der Elternklasse **Konto** geerbten Attribute zugreifen, also ohne Verwendung geeigneter Getter- und Setter?
- Ein direkter Zugriff ist nicht möglich wenn keine Getter und Setter vererbt werden

Frage 12:

Codieren Sie die Anweisung zum Anlegen eines Arrays vom Typ Konto! Was kann in den Elementen des Arrays gespeichert werden?! Konto [ ] arra = new Konto [5]
- Es werden die Adressen im Hauptspeicher vom Typ Konto gespeichert

Frage 13:

Die Codierung einer Klasse ABC beginnt folgendermaßen:

```
public abstract class ABC
{
        ……
        ……
}
```

- Welche Wirkung hat der Modifikator abstract?! - von dieser Klasse können keine Objekte erzeugt werden

- Vererbt sich die Eigenschaft, die abstract bewirkt, auf etwaige Kindklassen von ABC?

- die Eigenschaft abstract vererbt sich nicht

- Wird folgende Anweisung vom Compiler akzeptiert:

```
ABC var;
```

- Kann als Referenz Objekt verwendet werden

- Wird folgende Anweisung vom Compiler akzeptiert:

```
ABC [] arr = new ABC[10];
```

- Kann als Referenz Objekt verwendet werden

Frage 14:

Was versteht man unter dem IS-A-Prinzip?

- Objekte von Kindklassen ist eine Spezialisierung der Elternklasse ein Mitarbeiter ist auch eine Person, ist also nicht nur vom Typ der Kindklasse, sondern auch vom Typ der Elternklasse

Frage 15:

Was ist gemeint mit **Compile Time** und was mit **Run Time**?

Frage 16:

Die Codierung einer Methode beginnt so:

```
public double rechnen (int a, int b, float c, double d)
{
      …….
      …….
]
```

- Welchen Kopf hat die Methode? - double rechnen (int,int,float,double)
- Welche Signatur hat die Methode? - rechnen (int,int,float,double)
- Kann es in der Klasse, zu der die Methode rechnen() gehört, noch eine weitere, andere Methode rechnen() geben?! - Ja, jedoch muss sich der Kopf der Methode Irgendwo Unterscheiden

Frage 17:

Was ist der Zweck der Annotation

@Override

Frage 18:

Folgendes Java-Programm liegt vor:

```
 public class Demo1
{
    public static void main(String [] args)
   {
            int a = 12;
            int b = 7;
            int c;

             c = summe(a,b);
            System.out.println("c = "+c);
  }
  public static int summe(int x, int y)
  {             return x + y;
  }

 }
```

- Erklären Sie in Worten die folgenden Begriffe:

  - Aktueller Parameter - Die übergebenen Werte

  - Formaler Parameter - Platzhalter für die übergebenen Werte

- Kennzeichnen Sie im obigen Programm

  alle aktuellen Parameter
  alle formalen Parameter

Frage 19:

```
public class Demo2
{
     public static void main(String [] args)
    {              i
            nt [] arr = {34, 3, 15,-6, 12,71, 27, 31, 2, 89, 45, 62, 9, 38, 17};

            maxdist = maximale_distanz(arr);
```

```
        System.out.println("Maximale Distanz = "+maxdist);
    }
}
```

Die Methode maximale_distanz() soll die Differenz zwischen dem kleinsten und dem größten Wert im Array arr ermitteln und den ermittelten Wert an das aufrufende Programm (main()) zurückgeben.

Codieren Sie die Methode maximale_distanz() so, dass sie in die Klasse Demo2 eingefügt werden kann!

Frage 20:

Strings kann man auf verschiedene Art und Weiseerzeugen. Erläutern Sie in Worten den Unterschied zwischen

String s1 = "Textverarbeitung"; - Wird während der Compiletime erzeugt

und

String s2 = new String("Textverarbeitung)"; - Wird während der Runtime erzeugt

Erläutern Sie den Unterschied zwischen den folgenden Methoden der Klasse String:

- equals()
- compareTo()

In welcher typischen Situation wird equals() eingesetzt? - vergleich auf Gleichheit

In welcher typischen Situation wird compareTo() eingesetzt? - zum Sortieren (nach ASCI code)

Frage 21:

```
public class Demo3
{
    public static void main(String[] args)
    {
        String s = "Das  ist  aber  toll".toUpperCase();
        String t = s.replace(" ","");
        int i = s.replace(" ","").indexOf("toll");
        int j = s.replace(" ","").length();
        int k = s.substring(6,16).replace(" ","").length();
        System.out.println("s = "+s);     - s = DAS IST ABER TOLL
        System.out.println("t = "+t);     - t = DASISTABERTOLL
        System.out.println("i = "+i);     - i = -1
        System.out.println("j = "+j);     - j = 14
        System.out.println("k = "+k);     - k = 5
    }
```

}

Welche Ausgabe erzeugt das Programm?

Frage 22:

- Mit welchem Begriff aus dem realen Leben ist eine Klasse vergleichbar?
  Beispiel? - Bauplan
- Mit welchem Begriff aus dem realen Leben ist ein Objekt vergleichbar?
  Beispiel? - Mit realen gegenstand (z.B KFZ, Bankkonto, …)
- Aus welchen prinzipiellen Komponenten besteht eine Klasse? Attribute, Konstruktoren, Akzesso.
- Welche Aussagen über Konstruktoren sind richtig (+), welche falsch (-) ?

Der Compiler baut immer einen Standardkonstruktor in eine Klasse ein (-)

Eine Klasse kann mehrere Konstruktoren haben (+)

Der Zugriffsmodifikator eines Konstruktors kann nicht private sein (-)

Ein Konstruktor kann keinen Wert mit return liefern (+)

Einen Konstruktor kann man nur zusammen mit new aufrufen (+)

Frage 23:

Wie kann man aus einem Konstruktor einer Kindklasse heraus einen Konstruktor der Elternklasse aufrufen? - Mit Super

Frage 24:

Aus welchen prinzipiellen Abschnitten besteht eine Klasse? - Attribute, Konstruktoren, Akzessoren

Frage 25:

Was versteht man unter dem **statischen Typ** einer Variablen, was unter dem **dynamischen Typ?** Was versteht man unter **Late Binding?**

Warum die die Methode **toString()** ein Beispiel für eine **polymorphe Methode?**

Frage 26:

Kann eine Variable während des Programmablaufs ihren Typ ändern? - Ja mit Cast