Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 1(a) / Moodle-Aufgabe 1: 8 Punkte

Referenzlösung:

```
public interface X <A extends Canvas & IntPredicate> {

    default <B extends Component> boolean def                //  public wird akzeptiert
                ( LinkedList<? super Integer> x, B y  ) {
        if ( x == null )
           return false;
        if ( y is instanceof Canvas )
           x.add ( new Integer ( 1 ) );    // Bei beidem new Integer oder Boxing ok
        else
           x.add ( -1 );
        return true;
    )

    Double fct ( BiFunction<A,Integer,Double> op, A a );              //  public wird akzeptiert       }
}
```

Bewertungsschema:

A Die Verknüpfung der beiden Restriktionen im Kopf von `X` ist erkennbar korrekt.

Anmerkung: Sollte die Anforderung für A hier nicht, aber in (b) oder (c) erfüllt sein, wird dieser Punkt hier auch vergeben.

B *A nicht nochmals werten*: Der Kopf von `X` ist insgesamt erkennbar korrekt.

C Der generische Typparameter von `def` ist erkennbar korrekt und auch an der korrekten Stelle.

D Der Parameter `x` von `def` ist erkennbar korrekt.

E Die Rückgabe von `def` ist in jedem Fall erkennbar korrekt.

F Ob ein Wert an `x` angehängt wird und wenn ja welcher ist in jedem Fall korrekt.

G Methode `fct` ist erkennbar korrekt.

Anmerkung: Sollte `func` hier nicht korrekt, aber der Kopf von `func` in (b) korrekt sein, wird dieser Punkt hier auch vergeben.

H Nur zu vergeben, wenn auch alle anderen Punkte vergeben werden und auch darüber hinaus nichts falsch ist: Die gesamte Lösung ist erkennbar syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 1(b) / Moodle-Aufgabe 2: 5 Punkte

Referenzlösung:

```
abstract public class Y <A extends Canvas & IntPredicate>
                                              implements X<A>, Consumer<A> {

    public Double fct ( BiFunction<A,Integer,Double> op, A a ) {
       return op.apply ( a, 2 );    // Alternativ new Integer statt Boxing
    }
}
```

Bewertungsschema:

A Die `implements`-Klausel mit den beiden implementierten Interfaces ist erkennbar korrekt gebildet.

B *A nicht nochmals werten*: Der Kopf der Klasse ist erkennbar korrekt (insbesondere `abstract`).

Anmerkung: Ob der generische Typparameter korrekt gebildet ist, wird schon in (a) bewertet, soll hier nicht nochmals bewertet werden.

C Der Kopf der Methode `fct` ist erkennbar korrekt (insbesondere `public`).

D Der Rumpf der Methode `fct` ist erkennbar korrekt.

E Nur zu vergeben, wenn auch alle anderen Punkte vergeben werden und auch darüber hinaus nichts falsch ist: Die gesamte Lösung ist erkennbar syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 1(c) / Moodle-Aufgabe 3: 4 Punkte

Referenzlösung:

```
public class Z <A extends Canvas & IntPredicate> extends Y<A> {

    private A a;

    public void accept ( A t ) {    // Parametername a mit "this.a = a" ist auch ok
       a = t;
    }
}
```

Bewertungsschema:

A Die Methode `accept` ist erkennbar korrekt (insbesondere `public`).

B Die `extends`-Klausel ist erkennbar korrekt.

C Das Attribut ist erkennbar korrekt definiert.

D Nur zu vergeben, wenn auch alle anderen Punkte vergeben werden und auch darüber hinaus nichts falsch ist: Die gesamte Lösung ist erkennbar syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 2(a) / Moodle-Aufgabe 4: 5 Punkte

Referenzlösung:

```
public class Exc3 extends Exc1 {

   public Exc3 ( Supplier<String> supp ) {
      super ( supp == null || supp.get() == null ? 0 : 1 );
      // Alternativ supp == null ? 0 : supp.get() == null ? 0 : 1
    }
}
```

Bewertungsschema:

A Die Fallunterscheidung mittels Bedingungsoperator ist erkennbar syntaktisch und semantisch korrekt.

B Die Fallunterscheidung mittels Bedingungsoperator ist erkennbar logisch korrekt.

C *A+B nicht nochmals werten*: Der Aufruf von `super` ist erkennbar korrekt.

D Die `extends`-Klausel ist erkennbar korrekt.

E Nur zu vergeben, wenn auch alle anderen Punkte vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 2(b) / Moodle-Aufgabe 5: 8 Punkte

Referenzlösung:

```
public static char m ( Optional<Character> ch, Supplier<String> supp ) throws Exc1 {
    if ( ch == null || supp == null )   // Alternativ: zwei separate if's
       throw new Exc1 ( 3 );
    if ( ch.isEmpty() )
       throw new Exc2 ('a');
    char c = ch.get().charValue();   // Auto-Unboxing auch ok
    if ( c == 'b' )                  // ch.get() == 'b' auch ok
       throw new Exc3 ( supp );
    return c;
}
```

Anmerkungen:

1. In der `throws`-Klausel ist jede Aufzählung von einer oder mehreren Exception-Klassen, die `Exc1` mit beinhaltet, ok (natürlich sofern syntaktisch korrekt).
2. Die Referenzlösung folgt 1:1 der logischen Struktur der Aufgabenstellung. Logisch äquivalente Lösungen mit anderer Struktur sind selbstverständlich ebenfalls ok (sofern korrekt gebildet). Dies schließt natürlich auch die geeignete Ersetzung von `if`-Verzweigungen durch Ausdrücke mit Bedingungsoperator ein.
3. Anstelle der Zwischenspeicherung des in `ch` enthaltenen Zeichens darf auch 2x `get` aufgerufen werden.

Bewertungsschema:

A Die `throws`-Klausel ist erkennbar korrekt (muss nicht geschickt zusammengefasst sein).

Anmerkung: Falls die `throws`-Klausel hier inkorrekt, aber in Teilaufgabe (c) korrekt ist, soll dieser Punkt ebenfalls vergeben werden.

B *A nicht nochmals werten*: Der Kopf der Methode `m` ist erkennbar korrekt (insbesondere `static`).

C Der Fall `ch == null` wird erkennbar korrekt getestet und behandelt.

D Der Fall `supp == null` wird erkennbar korrekt getestet und behandelt.

E Der Fall, dass `ch` leer ist, wird erkennbar korrekt getestet und behandelt.

F Der Fall, dass `ch` das Zeichen `'b'` enthält, wird erkennbar korrekt getestet und behandelt.

G Der Fall, dass `ch` ein anderes Zeichen als `'b'` enthält, wird erkennbar korrekt getestet und behandelt.

H Nur zu vergeben, wenn auch alle anderen Punkte vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 2(c) / Moodle-Aufgabe 6: 9 Punkte

Referenzlösung:

```
public void m2 ( Supplier<String> supp ) throws Exc1 {
   try {
      Y.m ( Optional.ofNullable(new Character('c')), supp );
                                       // Boxing für Character ist auch ok.
   }
   catch ( Exc2 exc ) {
      throw new Exc3 ( supp );
   }
   catch ( Exc3 exc ) {
      throw new Exc2 ( 'd');
   }
}
```

Anmerkung: Alternativ darf der Rückgabewert von `m2` auch gespeichert werden.

Bewertungsschema:

A Der Kopf der Methode ist erkennbar korrekt (insbesondere kein `static`).

B Die Konstruktion des `Optional`-Objektes mit `ofNullable` ist erkennbar korrekt.

C *B nicht nochmals werten*: Der Aufruf von `m` mit `Y` ist erkennbar korrekt.

D *B und C nicht nochmals werten*: Der `try`-Block ist erkennbar korrekt gebildet.

E Der Wurf einer `Exc2` wird erkennbar korrekt gefangen und behandelt.

F Der Wurf einer `Exc3` wird erkennbar korrekt gefangen und behandelt.

G Eine von `m` geworfene `Exc1` wird erkennbar korrekt ohne Fangen weitergereicht.

H *B-G nicht nochmals werten*: Der gesamte `try-catch`-Block ist erkennbar korrekt.

I Nur zu vergeben, wenn auch alle anderen Punkte vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 3(a) / Moodle-Aufgabe 7: 14 Punkte

Referenzlösung:

```
public ListItem<T> bar ( ListItem<T> lst, ListItem<T> lstReversed ) {
   if ( lst == null )
      return lstReversed;
   ListItem<T> tmp = new ListItem<T>();
   tmp.key = lst.key;
   tmp.next = lstReversed;
   return bar ( lst.next, tmp );
}

public ListItem<T> foo ( ListItem<T> lst1, ListItem<T> lst2 ) {
   if ( lst1 == null )
      return bar ( lst2, null );
   ListItem<T> tmp = new ListItem<T>();
   tmp.key = lst1.key;
   tmp.next = foo ( lst1.next, lst2 );
   return tmp;
}
```

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 3(a) / Moodle-Aufgabe 7: 14 Punkte

Deckelung: bei 5 Punkten, falls auch nur eine Schleife vorkommt oder falls eine Liste in eine andere Art von Datenstruktur kopiert wird.

Bewertungsschema:

- Methode `bar`:

  A Der Kopf der Methode `bar` ist erkennbar korrekt.

  B Der Fall, dass die Liste in `bar` gleich `null` ist, ist entweder erkennbar gar nicht möglich (weil in `foo` abgefangen) oder wird erkennbar korrekt in `bar` erkannt und behandelt.

  C Das einzurichtende Listenelement wird erkennbar korrekt eingerichtet.

  D Das einzurichtende Listenelement wird erkennbar korrekt vorne an das bisher erstellte Suffix der Ergebnisliste angehängt.

  E Der Schlüsselwert ist für jedes einzurichtende Listenelement erkennbar korrekt gewählt und korrekt kopiert.

  F Der rekursive Aufruf von `bar` ist erkennbar korrekt.

  G Nur zu vergeben, wenn auch alle anderen Punkte A-F vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

- Methode `foo`:

  H Der Kopf der Methode `foo` ist erkennbar korrekt.
  Anmerkung: Falls der Kopf der Methode `foo` nicht hier, aber in (b) korrekt ist, wird dieser Punkt ebenfalls vergeben.

  I Der Aufruf von `bar` ist erkennbar korrekt und mit den korrekten aktualen Parametern.

  J *I nicht nochmals werten*: Der Fall, dass `lst1` gleich `null` ist, wird erkennbar korrekt erkannt und behandelt.

  K Das einzurichtende Listenelement wird erkennbar korrekt eingerichtet und mit korrektem Schlüsselwert versehen.

  L Der rekursive Aufruf von `foo` ist erkennbar korrekt und mit den korrekten aktualen Parametern.

  M Die Rückgabe von `foo` ist erkennbar korrekt.

  N Nur zu vergeben, wenn auch alle anderen Punkte H-M vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 3(b) / Moodle-Aufgabe 8: 14 Punkte

Referenzlösung:

```
public ListItem<T> bar ( ListItem<T> lst ) {
   ListItem<T> lstReversed = null;
   for ( ListItem<T> p = lst; p != null; p = p.next ) {
       ListItem<T> tmp = new ListItem<T>();
       tmp.key = p.key;
       tmp.next = lstReversed;
       lstReversed = tmp;
   }
   return lstReversed;
}

public ListItem<T> foo ( ListItem<T> lst1, ListItem<T> lst2 ) {
   if ( lst1 == null )
      return bar ( lst2 );
   ListItem<T>() head = new ListItem<T>();
   head.key = lst1.key;
   ListItem<T>() tail = head;
   for ( ListItem<T> p = lst.next; p != null; p = p.next )
      ListItem<T> tmp = new ListItem<T>();
      tmp.key = p.key;
      tail.next = tmp;
      tail = tmp;   // oder tail = tail.nect;
   }
   tail.next = bar ( lst2 );
   return head;
}
```

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 3(b) / Moodle-Aufgabe 8: 14 Punkte

Deckelung: bei 5 Punkten, falls Rekursion vorkommt oder falls eine Liste in eine andere Art von Datenstruktur kopiert wird.

Bewertungsschema:

A In `bar` oder `foo`: Der Fall, dass `lst2` gleich `null` ist, wird erkennbar korrekt erkannt und behandelt (nämlich durch Rückgabe von `bar(lst2)`).

- Methode `bar`:

  B Der Verweis auf den Kopf der Ergebnisliste (`lstReversed` in der Referenzlösung) wird erkennbar korrekt eingerichtet und initialisiert.

  C Die Elemente von `lst` werden erkennbar korrekt durchlaufen.

  D Das einzurichtende Listenelement wird erkennbar korrekt und mit dem korrekten Schlüsselwert eingerichtet.

  E Das einzurichtende Listenelement wird erkennbar korrekt vorne an die Ergebnisliste angehängt.

  F Nur zu vergeben, wenn auch alle anderen Punkte A-E vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

- Methode `foo`:

  G Der Fall, dass `lst1` gleich `null` ist, wird erkennbar korrekt getestet und behandelt.

  H Die initiale Einrichtung eines ersten Elements ist erkennbar korrekt und mit dem erkennbar korrekten Schlüsselwert.

  I Die Einrichtung von `head` und `tail` ist erkennbar korrekt.

  J Der Durchlauf durch `lst1` ist erkennbar korrekt (insbesondere Auslassen des ersten Elements).

  K Das einzurichtende Listenelement wird in jedem Schleifendurchlauf erkennbar korrekt und mit erkennbar korrektem Schlüsselwert eingerichtet.

  L Das einzurichtende Listenelement wird in jedem Schleifendurchlauf erkennbar korrekt vorne an die Ergebnisliste angehängt.

  M Der Verweis auf den Kopf der Ergebnisliste (also `head`) wird erkennbar korrekt aktualisiert.

  N Nur zu vergeben, wenn auch alle anderen Punkte G-M vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 3(c) / Moodle-Aufgabe 9: 10 Punkte

Referenzlösung:

```
public void bar ( List<T> lst, List<T> result ) {
    final int lengthOfLst1 = result.size();
    Iterator<T> it = lst.iterator();
    while ( it.hasNext() )
       result.add ( lengthOfLst1, it.next() );
}

public List<T> foo ( List<T> lst1, List<T> lst2 ) {
   List<T> result = new LinkedList<T>();
   Iterator<T> it = lst1.iterator();
   while ( it.hasNext() )
      result.add ( it.next() );
   bar ( lst2, result );
   return result;
}
```

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 3(c) / Moodle-Aufgabe 9: 10 Punkte

Deckelung: bei 5 Punkten, falls nicht allein über Iteratoren auf die Elemente der beiden Eingabelisten zugegriffen wird oder falls eine Liste in eine andere Datenstruktur kopiert wird (z.B. mit Methode `toArray` oder Methode `stream`).

Bewertungsschema:

- Methode `bar`:

  A Der Kopf der Methode `bar` ist erkennbar korrekt.

  B Die Einrichtung und Initialisierung von `lengthOfLst1` ist erkennbar korrekt (insbesondere `final`)

  C Der Durchlauf mittels Iterator ist erkennbar korrekt.

  D Das Einfügen in die Ergebnisliste ist erkennbar korrekt, insbesondere wird der korrekte Wert an der korrekten Position eingefügt.

  E Nur zu vergeben, wenn auch alle anderen Punkte A-D vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

- Methode `foo`:

  F Der Kopf der Methode `foo` ist erkennbar korrekt.

  G Die Einrichtung der Ergebnisliste ist erkennbar korrekt.

  H Der Durchlauf durch `lst1` mittels Iterator ist erkennbar korrekt.

  I Das Einfügen in die Ergebnisliste ist erkennbar korrekt, insbesondere wird der korrekte Wert an der korrekten Position eingefügt.

  J Nur zu vergeben, wenn auch alle anderen Punkte F-I vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 4(a) / Moodle-Aufgabe 10: 7 Punkte

Referenzlösung:

```
public interface Foo {
    Integer apply ( List<Integer> lst1, List<Integer> lst2 ); // public wäre ok
}

public class MyFoo implements Foo {
    public Integer apply ( List<Integer> lst1, List<Integer> lst2 ) {
        return lst1.get(0).intValue() * lst2.get(0).intValue();   //  Auto-Unboxing auch ok
    }
}
```

Bewertungsschema:

- Interface `Foo`:
  - A Die funktionale Methode ist erkennbar korrekt.
  - B Der Kopf der Definition des Interface ist erkennbar korrekt.
- Klasse `MyFoo`:
  - C Die Extraktion des `int`-Wertes aus der Liste ist mindestens einmal korrekt.

    Anmerkung: Die Berechnung der beiden Faktoren dürfen auch jeweils in mehrere Anweisungen mit Zwischenspeicherung zerlegt sein.
  - D *C nicht nochmals werten*: Der zurückzuliefernde Ausdruck in `apply` ist korrekt.
  - E Der Kopf der Methode `apply` ist erkennbar korrekt (insbesondere `public`).
  - F Der Kopf der Klasse ist erkennbar korrekt.
- G Nur zu vergeben, wenn auch alle anderen Punkte A-F vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 4(b) / Moodle-Aufgabe 11: 7 Punkte

Referenzlösung:

```
public interface Bar <T> {
    boolean test ( T x, T y, T z ); // public wäre ok
}

public class MyBar implements Bar<Integer> {
    public boolean test ( Integer x, Integer y, Integer z ) {
        return ( x < y ) && ( x < z );   // Darf auch ohne Auto-Unboxing, also mit intValue o.ä. sein.
    }
}
```

Bewertungsschema:

- Interface `Bar`:
  - A Die funktionale Methode ist erkennbar korrekt.
  - B Der Kopf der Definition des Interface ist erkennbar korrekt.
- Klasse `MyBar`:
  - C Der logische Ausdruck ist erkennbar korrekt.
  - D *C nicht nochmals werten*: Der zurückzuliefernde Ausdruck ist korrekt.
  - E Der Kopf der Methode `test` ist erkennbar korrekt.
  - F Der Kopf der Klasse ist erkennbar korrekt.

G Nur zu vergeben, wenn auch alle anderen Punkte A-F vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 4(c) / Moodle-Aufgabe 12: 13 Punkte

Referenzlösung:

```
public static BiFunction<List<Integer>,List<Integer>,Boolean>
    foobar ( Bar<Integer> bar, Foo foo, Integer t1, Integer t2 ) {
        return ( List<Integer> x, List<Integer> y ) -> bar.test ( foo.apply ( x, y ), t1, t2 ); };

X.foobar (
    ( Integer c, Integer d, Integer e ) -> ( c < d && c < e ),
    ( List<Integer> a, List<Integer> b ) -> a.get(0) * b.get(0),
    3, 5 )
.apply ( lst1, lst2 );
```

Anmerkung: Im Rahmen der Java-Regeln dürfen die Lambda-Ausdrücke selbstverständlich verkürzt/unverkürzt formuliert sein.

Bewertungsschema:

- Definition von `foobar`:
  - A Die Parameterliste in der Definition von `foobar` ist erkennbar korrekt.
  - B Der Rückgabetyp von `foobar` ist erkennbar korrekt.
  - C *A+B nicht nochmals werten*: Der Kopf der Methode `foobar` ist erkennbar korrekt.
  - D Der Aufruf von `foo` ist erkennbar korrekt.
  - E *F nicht nochmals werten*: Der Aufruf von `bar` ist erkennbar korrekt.
  - F *D+E nicht nochmals werten*: Der Lambda-Ausdruck in der Definition von `foobar` ist erkennbar korrekt.
- Aufruf von `foobar`:
  - G Der Rumpf des Lambda-Ausdrucks für `foo` ist erkennbar korrekt.
  - H *G nicht nochmals werten*: Der Lambda-Ausdruck für `foo` ist erkennbar korrekt.
  - I Der Rumpf des Lambda-Ausdrucks für `bar` ist erkennbar korrekt.
  - J *I nicht nochmals werten*: Der Lambda-Ausdruck für `bar` ist erkennbar korrekt.
  - K *G-J nicht nochmals werten*: Der Aufruf von `foobar` ist erkennbar korrekt.
    Anmerkung: Falls „`X.`“ fehlt, ist das hier auch ok.
  - L Der Aufruf der Rückgabe von `foobar` ist erkennbar korrekt.
- M Nur zu vergeben, wenn auch alle Punkte A-L vergeben werden und auch darüber hinaus nichts falsch ist: Die Definition von `foobar` ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 5(a) / Moodle-Aufgabe 13: 9 Punkte

Referenzlösung:

```
public static char[ ] foobar ( String[ ] a, String [ ] b ) {
   int numberOfValues = 0;
   int maxIndex = Math.min ( a.length, b.length );  // Alternativ if, switch oder Bedingungsoperator
   // Alternativ kann auf maxIndex verzichtet werden und das Minimum direkt in die Schleife kommen.
   for ( int i = 0; i < maxIndex; i++ )
      if ( a[i].charAt(0) == b[i].charAt(0) )
         numberOfValues++;
   char[ ] result = new char [ numberOfValues ];
   int indexInResult = 0;
   for ( int i = 0; i < maxIndex; i++ )
      if ( a[i].charAt(0) == b[i].charAt(0) ) {
         result[indexInResult] = a[i].charAt(0);
         indexInResult++;
      }
   return result;
}
```

Deckelung: bei 5 Punkten falls Rekursion vorkommt.

Bewertungsschema:

A Der Kopf der Methode `foobar` ist erkennbar korrekt.

Anmerkung: Falls nicht hier korrekt, aber in (b), soll dieser Punkt ebenfalls vergeben werden.

B Die Prüfung auf Gleichheit des ersten Zeichens in beiden Strings ist an mindestens einer der beiden notwendigen Stellen erkennbar korrekt.

C Der Rumpf der Schleife für das Bestimmen der Länge des zurückzuliefernden Arrays ist erkennbar korrekt.

D *C nicht nochmals werten*: Die Länge des zurückzuliefernden Arrays wird korrekt berechnet und das Array korrekt mit dieser Länge eingerichtet.

E Die Verwaltung des jeweils aktuellen Index im zurückzuliefernden Array (`indexInResult` in der Referenzlösung) ist erkennbar korrekt.

F *E nicht nochmals werten*: Es werden auf erkennbar korrekte Weise die korrekten Werte in das zurückzuliefernde Array geschrieben.

G Die Reihenfolge im zurückzuliefernden Array ist erkennbar korrekt.

H Es werden keine weiteren Zeichen außer den geforderten in das zurückzuliefernde Array geschrieben.

I Nur zu vergeben, wenn auch alle anderen Punkte A-H vergeben werden und auch darüber hinaus nichts falsch ist: Die Lösung ist erkennbar insgesamt syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 5(b) / Moodle-Aufgabe 14: 15 Punkte

Referenzlösung:

```
private static int foo ( String[ ] a, String[ ] b, int index ) {
   if ( index >= a.length || index >= b.length )
      return 0;
   return foo ( a, b, index + 1 ) + ( a[index].charAt(0) == b[index].charAt(0) ? 1 : 0 );
   // Alternativ switch oder Ausdruck mit Bedingungsoperator
}

private static void bar ( String[ ] a, String[ ] b, char[ ] result,
                          int indexInAAndB, int indexInResult ) {
   if ( indexInAAndB >= a.length || indexInAAndB >= b.length )
      return;
   if ( a[indexInAAndB].charAt(0) == b[indexInAAndB].charAt(0) ) {
      result[indexInResult] = a[indexInAAndB].charAt(0);
      indexInResult++;
   }
   bar ( a, b indexInAAndB+1, indexInResult );
}

public static char[ ] foobar ( String[ ] a, String[ ] b ) {
   char[ ] result = new int [ foo ( a, b, 0 ) ];
   bar ( a, b, result, 0, 0 );
   return result;
}
```

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 5(b) / Moodle-Aufgabe 14: 15 Punkte

Deckelung: bei 5 Punkten, falls auch nur eine Schleife vorkommt.

Bewertungsschema:

- Methode `foo`:
  - A Der Kopf der Methode ist erkennbar korrekt.
  - B Der Rekursionsabbruch wird erkennbar korrekt abgefragt und behandelt.
  - C Der rekursive Aufruf von `foo` ist erkennbar korrekt.
  - D *C nicht nochmals werten*: Die Rückgabe im Rekursionsschritt ist erkennbar in jedem Fall korrekt.
  - E Nur zu vergeben, wenn auch alle Punkte A-D vergeben werden und auch darüber hinaus nichts falsch ist: Die gesamte Methode `foo` ist erkennbar syntaktisch, semantisch und logisch vollständig korrekt.
- Methode `bar`:
  - F Der Kopf der Methode ist erkennbar korrekt.
  - G Der Rekursionsabbruch wird erkennbar korrekt abgefragt und behandelt.
  - H Es wird erkennbar genau im richtigen Fall ein Zeichen nach `result` kopiert, und dieser Kopierakt ist auch erkennbar korrekt.
  - I Der rekursive Aufruf von `bar` (Indizes!) ist erkennbar korrekt in dem Fall, dass das aktuelle Zeichen zu kopieren ist.

    Anmerkung: In der Referenzlösung wird die Unterscheidung für I und J durch Erhöhung von `indexInResult` im Falle von I realisiert. Stattdessen könnte `bar` aber beispielsweise auch zweimal separat aufgerufen werden, einmal mit `indexInResult` und einmal mit `indexInResult+1`.
  - J Der rekursive Aufruf von `bar` (Indizes!) ist erkennbar korrekt in dem Fall, dass das aktuelle Zeichen *nicht* zu kopieren ist.
  - K Nur zu vergeben, wenn auch alle Punkte F-J vergeben werden und auch darüber hinaus nichts falsch ist: Die gesamte Methode `bar` ist erkennbar syntaktisch, semantisch und logisch vollständig korrekt.
- Methode `foobar`:
  - L Der Aufruf von `foo` in `foobar` ist erkennbar korrekt und mit geeigneten aktualen Parameterwerten.
  - M Das zurückzuliefernde Array wird erkennbar korrekt und mit der korrekten Länge eingerichtet.
  - N Der Aufruf von `bar` in `foobar` ist erkennbar korrekt und mit geeigneten aktualen Parameterwerten.
  - O Nur zu vergeben, wenn auch alle Punkte L-N vergeben werden und auch darüber hinaus nichts falsch ist: Die gesamte Methode `foobar` ist erkennbar syntaktisch, semantisch und logisch vollständig korrekt.

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 6(a) / Moodle-Aufgabe 15: 6 Punkte

Referenzlösung und Bewertungsschema:

(i) Der erste Parameter
- A extrahiert die Studiengebührenhöhe (oder das Attribut o.ä.) und
- B addiert sie auf die Zwischensumme (o.ä.).

C Der zweite Parameter ist 0.

(i) Der erste Parameter
- D verknüpft (o.ä.) das jeweilige Element der Liste
- E mit den Zwischenergebnis (o.ä.)

F Der zweite Parameter ist der Initialwert (o.ä.).

Aufgabe 6(b) / Moodle-Aufgabe 16: 2 Punkte

Referenzlösung und Bewertungsschema:

A $F1$ (alternativ: die gesamte `for`-Schleife o.ä.)

B Wenn $F2$ in $F1$ ist (alternativ: ineinandergeschachtelt o.ä.)

Bewertungsgrundlage

Funktionale und objektorientierte Programmierkonzepte

4. April 2023, Version 01

Aufgabe 6(c) / Moodle-Aufgabe 17: 6 Punkte

Referenzlösung und Bewertungsschema:

- `BufferedReader`:

  A Die Zeichen (alternativ Daten o.ä.) werden nicht einzeln herausgelesen,

  B sondern immer gleich mehrere auf einmal in einem Buffer (alternativ Puffer o.ä.)

  C und der Puffer wird durch Einlesen gefüllt.

  Anmerkung: Dass der Puffer nicht vollständig gefüllt wird, wenn nicht mehr ausreichend viele Zeichen in der Datenquelle sind, darf fehlen.

- `BufferedWriter`:

  D Die Zeichen (alternativ Daten o.ä.) werden nicht sofort hinausgeschrieben,

  E sondern erst in einem Buffer (alternativ Puffer o.ä.)

  F und der Puffer wird hinausgeschrieben.

  Anmerkung: Jede Art von Flush darf fehlen.

Aufgabe 6(d) / Moodle-Aufgabe 18: 2 Punkte

Referenzlösung und Bewertungsschema:

A Nicht alle Vögel können fliegen.

B Fliegen müsste auch für alle abgeleiteten Klassen definiert sein.