Abgabe-Übungsblatt A1 – Lösungen

Die Lösungen für dieses Übungsblatt sollen abgegeben werden. Die erzielten Punkte werden für die Studienleistung angerechnet. Durch dieses Übungsblatt können bis 8 Punkte erworben werden. Insgesamt können bis zu 40 Punkte erworben werden. Die Studienleistung ist bei mehr als 20 Punkten erfolgreich absolviert.
Bitte drucken Sie diese drei Blätter einfach aus und bearbeiten Sie die Aufgaben auf dem Ausdruck.

Die Abgabe kann bis zum 7.5.2018, 12.00 Uhr in dem mit SWT gekennzeichneten Briefkasten in der 1. Etage der OH12 am Übergang zum Erdgeschoß OH14 ein. Den Briefkasten finden Sie, wenn Sie in Richtung OH12 schauen.

Name/Matrikelnummer (Gruppenabgaben von bis zu 3 Studierenden):

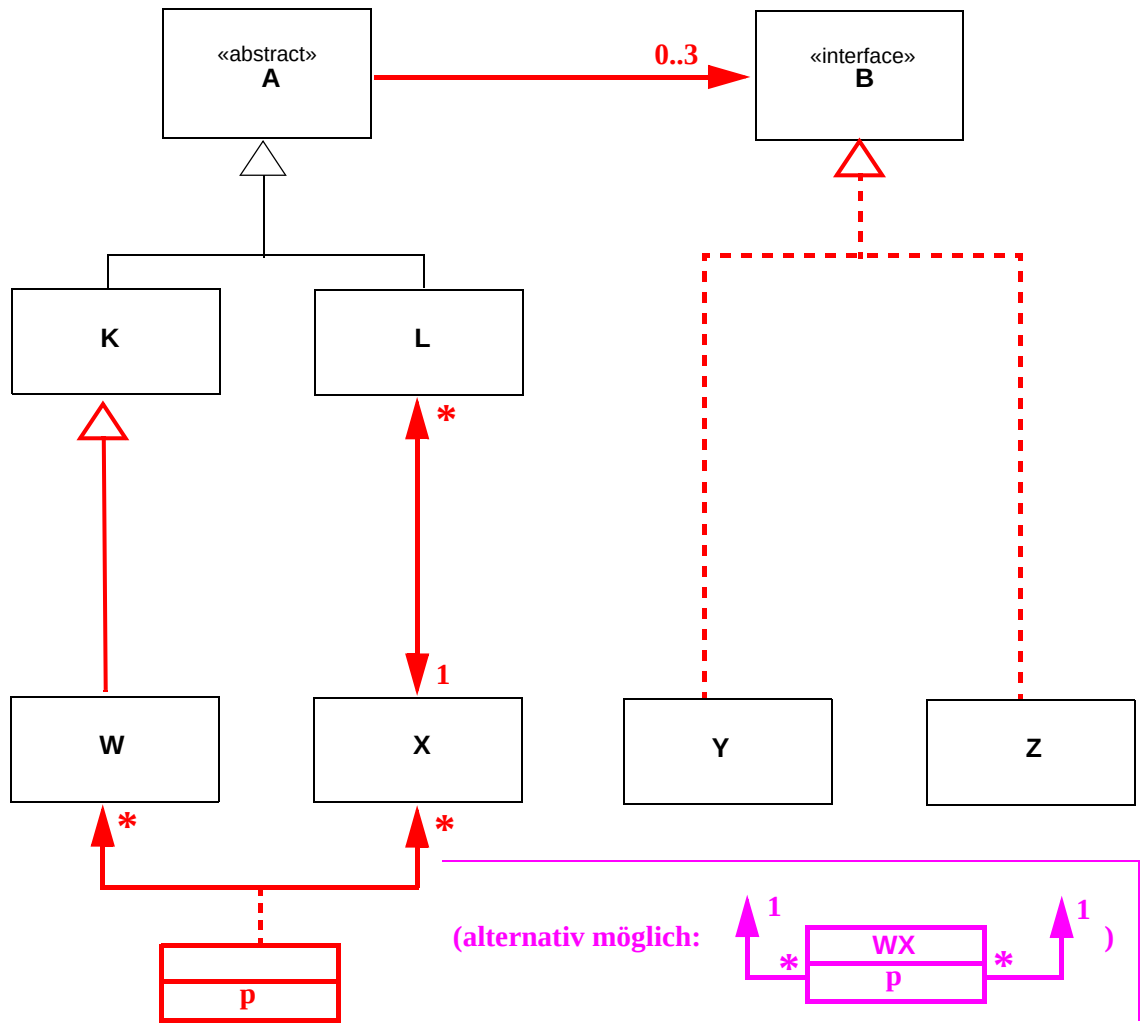
- _____ / _____
- _____ / _____
- _____ / _____

Übungsgruppe: _____

Aufgabe 1 – Klassendiagramm

(2 Punkte) Erweitern Sie das folgende Klassendiagramm derart, dass es die nachfolgende Beschreibung modelliert. Verwenden Sie die bereits vorgegebenen Elemente, wo dieses möglich ist, oder ergänzen Sie weitere Elemente.

- Die Klasse `W` spezialisiert die Klasse `K`.
- Die Klassen `Y` und `Z` realisieren das Interface B.
- Jedes `L`-Objekt kennt genau ein `X`-Objekt, ein `X`-Objekt kann beliebig viele `L`-Objekte kennen.
- Jedes `W`-Objekt kann beliebig viele `X`-Objekte kennen, jedes `X`-Objekt kann beliebig viele `W`-Objekte kennen. Nur genau dann, wenn ein `X`-Objekt ein `W`-Objekt kennt, existiert ein Wert `p`, der der Verbindung zwischen genau diesen beiden Objekten zugeordnet ist.
- Jedes zu `A` kompatible Objekt kann bis zu drei zu B kompatible Objekte kennen.



Abgabe-Übungsblatt A1 – Lösungen

Aufgabe 2 – Entwurfsmuster Adapter

(2 Punkte) Eine Anwendung erwartet ein Objekt, das zu der abstrakten Java-Klasse `Collection` kompatibel ist. Die vorgegebenen Methoden sollen folgende Wirkung haben:

- Die Methode `add` fügt das als Parameter übergebene Objekt in die Datenstruktur ein.
- Die Methode `remove` gibt das Objekt aus der Datenstruktur zurück, das am längsten enthalten ist, und entfernt es. Ist die Datenstruktur leer, wird `null` zurückgegeben.

Die Klasse `Data` liegt bereits als Java-Implementierung vor. Ein Objekt der Klasse `Data` verwaltet Objekte mit folgenden Methoden:

- Der Konstruktor legt ein neues Objekt der Klasse `Data` an.
- Die Methode `isEmpty` gibt `true` zurück, wenn kein Objekt verwaltet wird; sonst `false`.
- Die Methode `set` fügt das als Parameter übergebene Objekt hinzu.
- Die Methode `getIndex()` liefert die Position des Objekts, das am längsten in der Datenstruktur aufbewahrt wird. Existiert kein Objekt, wird `-1` zurückgegeben.
- Die Methode `get(int i)` gibt das Objekt an Position `i` zurück. Existiert an der Positon `i` kein Objekt, wird `null` zurückgegeben.
- Die Methode `remove(int i)` entfernt das Objekt an Position `i`. Existiert an der Position `i` kein Objekt, geschieht nichts.

```
public abstract class Collection {
     public abstract void add( Object o );
     public abstract Object remove();
}
```

```
public class Data {
     public Data() {...}
     public boolean isEmpty() {...}
     public void set( Object o ) {...}
     public int getIndex() {...}
     public Object get( int i ) {...}
     public void remove( int i ) {...}
}
```

Geben Sie die Java-Implementierung einer Klasse `Connect` an, die einen geeigneten ***Adapter*** bereitstellt, der `Data` nutzt, um die Anforderungen von `Collection` zu erfüllen.

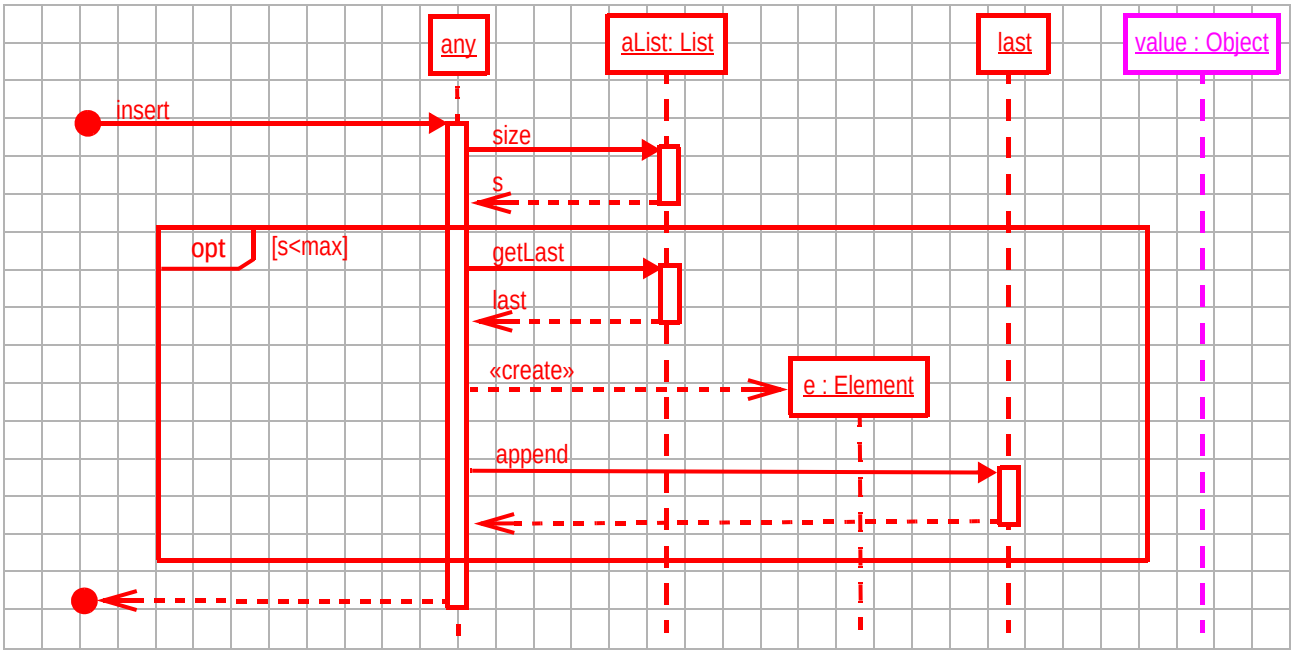
```
public class Connect extends Collection {
     private Data d = new Data();
     public void add( Object o ) {
      d.set( o );
     }
     public Object remove() {
      if ( d.isEmpty() ) {
        return null;
      } else {
        int i = d.getIndex();
        Object o = d.get( i );
        d.remove( i );
        return o;
      }
     }

}
```

Abgabe-Übungsblatt A1 – Lösungen

Aufgabe 3 – Sequenzdiagramm

(2 Punkte) Nehmen Sie an, dass die in der folgenden Methode `insert` benutzten Klassen und Methoden ihren Definitionen entsprechend verwendet und aufgerufen werden. Erstellen Sie ein Sequenzdiagramm, das den Ablauf eines Aufrufs der Methode `insert` zeigt.

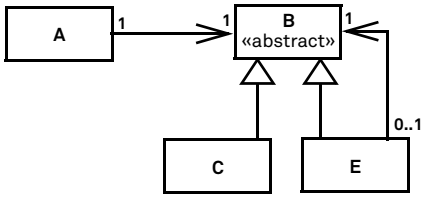
```
public void insert( List aList, Object value, int max ) {
    if ( aList.size() < max ) {
        aList.getLast().append( new Element( value ) );
    }
}
```



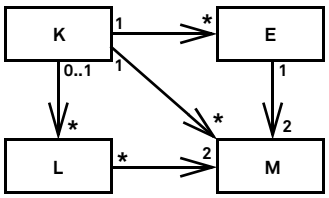
Aufgabe 4 – Objektdiagramm

(2 Punkte) Gegeben sind die folgenden beiden Klassendiagramme. Geben Sie für jedes dieser Klassendiagramme ein Objektdiagramm an, das ***genau vier*** Objekte enthält, von denen ***genau eines*** zur Klasse E gehört.

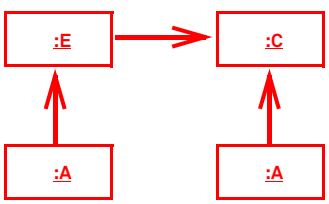
a)



b)



a)



b)