Aufgabe 1 (10BE + 7BE + 5BE + (8BE + 5BE) = 35BE)

Die bekannte Bank .gibdeingeld bietet verschiedene Konten an. Ein Kunde muss mindestens ein Konto bei der Bank haben, welches im Konstruktor des Kunden angelegt wird und auch nicht gelöscht werden darf. Es werden (u. a.) verschiedene Arten an Konten angeboten:

- Girokonto (Kreditrahmen, KreditkarteNr(wird beim Erstellen eines Kontos angegeben))
- Anlagekonto (Zinsen)
  - Tagesgeldkonto
  - Festgeldkonto
    - Neues Attribut Laufzeit (wird beim Erstellen eines Kontos angegeben)
    - Laufzeit soll geschrieben und gelesen werden können
    - Festgeldkonto wird mit Laufzeit (und Kontonummer) erzeugt

Objekte sollen ausschließlich von den Klassen Girokonto, Tagesgeldkonto und Festgeldkonto erzeugt werden können. Für die generierung eines neuen Konto-Objektes wird immer die Kontonummer angegeben.

a) Beschreiben Sie mit Hilfe des gegebenen UML-Klassen-Diagramms (siehe Material) die Klassen `Konto` und `Kunde` und die Beziehungen zwischen diesen Klassen. Gehen Sie dabei insbesondere auf die Schutzklassen ein.

Beschreiben Sie, wie die Konten eines Kunden gespeichert werden und welche Methoden für die Verwaltung der Konten verwendet werden.

b) Implementieren Sie die Klasse `Konto` mit ihren Attributen und den Methoden `Konto`, `einzahlen` und `auszahlen` gemäß den Vereinbarungen aus dem UML-Diagramm. (Alle weiteren Mothoden müssen nicht angegeben werden.)

c) Implementieren Sie die Methode `getGuthaben` der Klasse Kunde, welche das gesamte Guthaben (aller Konten) eines Kunden ermittelt und zurückgibt.

d) Ergänzen Sie das gegebene UML-Klassendiagramm um die oben beschriebenen VIER Klassen. Implementieren Sie die Klasse `Festgeldkonto` unter Verwendung der Prinzipien der Vererbung.

Aufgabe 2 (7BE + 7BE + 3BE = 17BE)

Die Firma Toys4Me hat sich auf die Herstellung von Spielzeug spezialisiert, insbesondere auf Autos und Plüschtiere. Als Autos werden (normale) Autos, Bagger, Laster und Feuerwehrautos hergestellt. Alle Autos haben auswechselbare Räder. Die Bagger können je nach Wunsch mit einem Greifer oder einer Schaufel ausgerüstet werden. Plüschtiere werden in den Varianten Teddy und Hase hergestellt.

a) Entwerfen Sie für das beschriebene System ein UML-Beziehungsdiagramm (ohne Attribute und Methoden).

b) Erläutern Sie mit Hilfe des gegebenen Systems folgende Begriffe und Zusammenhänge:
Vererbung, Generalisierung und Spezialisierung, abstrakte Klasse und Aufruf von Methoden der Oberklasse.

c) Erläutern Sie, weshalb eine Komposition als Beziehung zwischen den Klassen Bagger und Schaufel bzw. Greifer nicht günstig ist. Begründen Sie Ihre Wahl für die Art dieser Beziehung.

Aufgabe 3 (2BE + 4BE + 6BE + 6BE + 3BE + 5BE + 5BE + 2BE = 33BE)

a) Abstrakte Datentypen werden u. a. durch die Eigenschaft *Verbergen von Informationen* (*information hiding*) beschrieben. Erläutern Sie die inhaltliche Bedeutung dieser Eigenschaft.

b) Im folgenden Quelltext ist eine Methode eines ADT Liste dargestellt. Analysieren Sie den Quelltext hinsichtlich der Art der Liste und der Aufgabe der Methode. (Begründung)

```
protected void eineMethode(String content) {

    Element aElement = new Element(content);

    if (current == root) {
        root = aElement;
    } else {
        Element scout = root;
        while (scout.getNext() != current) {
            scout = scout.getNext();
        }
        scout.setNext(aElement);
    }
    aElement.setNext(current);
    current = aElement;
    count++;
}
```

c) Erläutern Sie mit Hilfe einer geeigneten Grafik, welche Schritte zum Löschen eines Elements aus einer doppelt verketteten Liste notwendig sind. Gehen Sie bei Ihrer Erläuterung auch auf mögliche Sonderfälle ein.

d) Implementieren Sie die in c) diskutierte Methode zum Löschen eines Elements aus einer doppelt verketteten Liste.

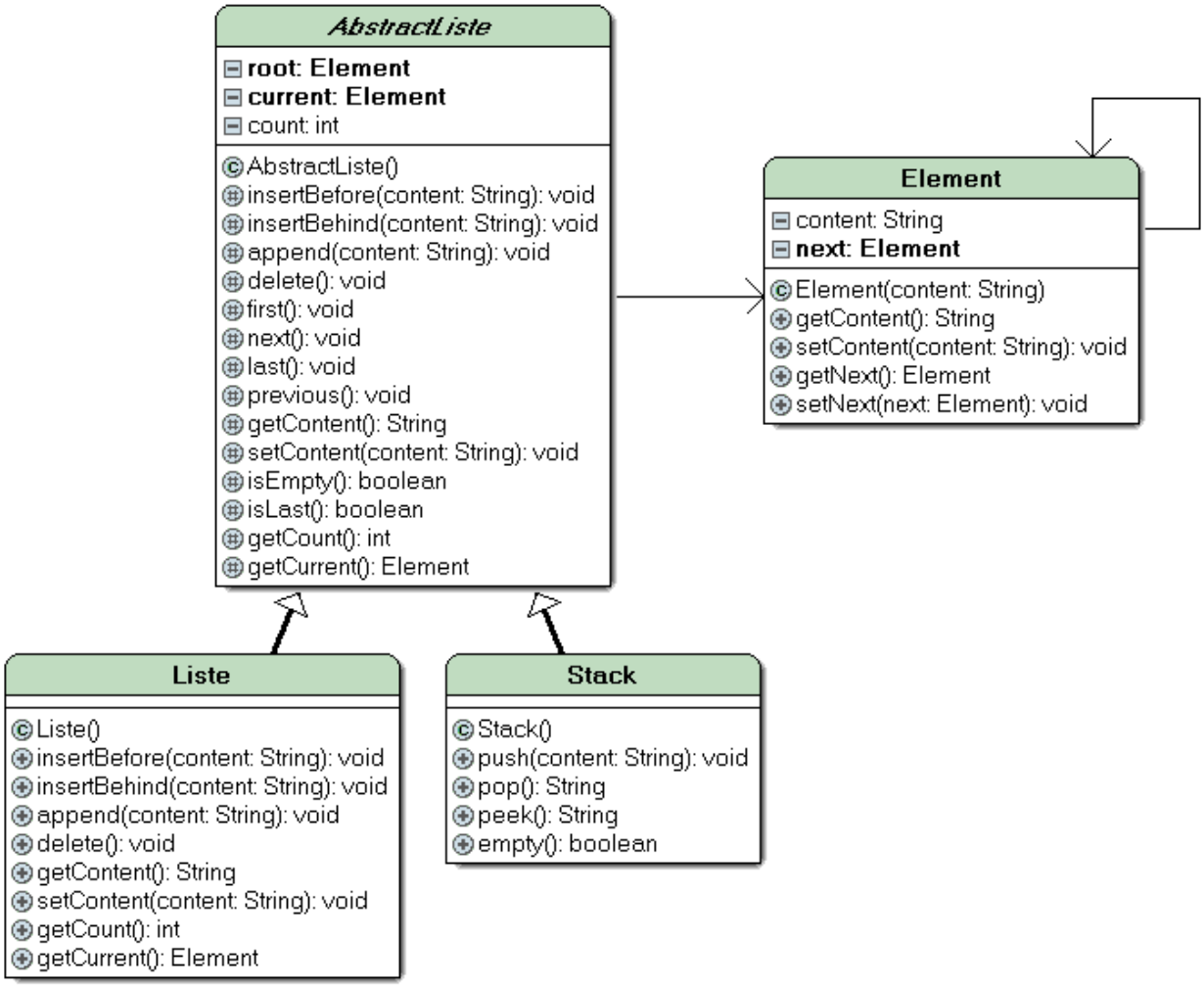
In Informatiksystemen werden manchmal Daten mit unterschiedlicher Geschwindigkeit erzeugt und verarbeitet, weshalb diese Daten in einem Puffer zwischengespeichert werden müssen. Die Druckerwarteschlange oder die Mailbox beim Internet-Provider sind dafür Beispiele. In solchen Situationen ist es notwendig Daten in einer so genannten *Warteschlange* (*Queue*) abzulegen. Dies ist ein ADT auf dessen Inhalt man nach dem FIFO-Prinzip (first in – first out) zugreifen kann. Praktisch kann man sich einen solchen Queue als Warteschlange etwa an einer Kasse vorstellen. Für die Arbeit mit dem ADT Queue werden folgende Zugriffsoperationen festgelegt:

- **put:** Die Daten werden hinten an die Warteschlange angehängt.
- **get:** Das erste Element der Warteschlange wird gelesen und entfernt.
- **getCount:** Es wird die Anzahl der Elemente in der Warteschlange zurückgegeben.

e) Erläutern Sie, wie die Daten in einer Warteschlange gespeichert werden, wenn für eine zu Beginn leere Warteschlange folgende Methodenaufrufe erfolgen und geben Sie die Belegung der Warteschlange nach der Abarbeitung an.

```
put("Aufgabe1.doc")
put("Birgit.doc")
put("Aufgabe.pas")
put("Lösung1.doc")
get()
get()
put("Lösung2.xls")
put("Gertrud.jpg")
get()
```

f) Ähnlich wie die im Unterricht erarbeiteten ADT's *Liste* und *Stack* soll der ADT *Queue* aus der abstrakten Klasse *AbstractListe* abgeleitet werden (siehe UML-Beziehungsdiagramm). Entwerfen Sie eine geeignete Klasse *Queue* und ergänzen Sie das gegebene UML-Diagramm. Beschreiben Sie anhand des vorliegenden Beispiels welchen Sinn eine abstrakte Klasse *AbstractListe* macht.



g) Implementieren Sie die Methoden *put() und get()* mit Hilfe der Klasse *AbstractListe*.

Häufig müssen die in einer Warteschlange gespeicherten Daten mit unterschiedlicher Priorität abgearbeitet werden, wichtige zuerst, weniger wichtige später. Bei diesem so genannten Prioritätspuffer werden die Daten immer in einer nach dem Schlüssel geordneten, aufsteigenden Reihenfolge eingefügt, d.h. dieser entspricht eigentlich mehreren aneinander gereihten Warteschlangen mit jeweils anderen Prioritäten.

h) Stellen Sie den Prioritätspuffer dar, wenn für diesen zu Beginn leeren Prioritätspuffer folgende Methodenaufrufe erfolgen:

```
put(1,"Kunde.doc")
put(2,"Birgit.doc")
put(1,"Aufgabe.pas")
put(4,"Lösung.doc")
put(2,"Blatt.xls")
get
get
put(3,"Rechnung.xls")
```

Material (Aufgabe 1)