

Aufgabe 1 Java-Drill

15 Punkte

- (a) Geben Sie jeweils zwei Schlüsselwörter in Java, welche die Konzepte *Kapselung* und *Vererbung* umsetzen.
- (b) Nennen Sie drei Arten von *Polymorphie*, die in Java vorkommen, und geben Sie jeweils ein kurzes Beispiel.
- (c) Beschreiben Sie in zwei Sätzen, was *Annotationen* in Java sind. Was tut die `@Override` Annotation? Warum würde man sie einsetzen?
- (d) Was ist der Unterschied zwischen dem dynamischen und dem statischen Typ einer (Referenz-)Variable?
- (e) Gegeben seien die folgende Schnittstelle und Klassen.

```
interface I {
    void a();
}
class A implements I {
    public void a() { System.out.println("A"); }
}
class B implements I {
    public void a() { System.out.println("B"); }
    public void b() { System.out.println("C"); }
}
```

Kommentieren Sie für jede einzelne Zeile der folgenden `main`-Methode genau, ob diese erlaubt ist bzw. warum nicht. Streichen Sie fehlerhafte Zeilen, und geben Sie die Ausgabe des verbleibenden Programms an. Geben Sie auch für jede Variable den statischen und die (evtl. wechselnden) dynamischen Datentypen an. Versuchen Sie, die Aufgabe ohne Rechner zu lösen.

```
public static void main(String[] args) {
    A a = new A();
    a.a();
    a.b();
    B b = new B();
    b.a();
    b.b();
    I i;
    i = new I();
}
```

```

    i = a;
    i.a();
    i.b();
    i = b;
    i.a();
    i.b();
    b = i;
    b = (B)i;
    a = (A)i;
}

```

(f) Verfahren Sie wie bei (e) mit dem folgenden Codefragment.

```

public static void main(String[] args) {
    List<?> x = new ArrayList<String>();
    List<Object> y = new ArrayList<Integer>();
    List<? extends Number> y2 = new ArrayList<Integer>();
    Object[] z = new Integer[3];
    z[2] = "ALP3";
}

```

Aufgabe 2 Spezifizieren und Implementieren eines ADT

10 Punkte

Spezifizieren Sie sinnvoll den abstrakten Datentyp “double ended queue”, der das Einfügen und Löschen von Objekten an beiden Enden unterstützt, und zwar auf zwei verschiedene Arten: einmal verbal (versuchen Sie, möglichst genau zu sein) und einmal modellierend (mathematisch oder in Haskell).

Formulieren Sie ein entsprechendes Interface in Java, und implementieren Sie den Datentyp auf zwei verschiedene Arten. Vergleichen Sie die Vor- und Nachteile Ihrer beiden Implementierungen (bezüglich Laufzeit, Implementierungsaufwand, Flexibilität, usw.).

Aufgabe 3 Untere Schranken für Prioritätswarteschlangen

5 Punkte

- Zeigen Sie, wie man eine Prioritätswarteschlange verwenden kann, um eine Folge von n vergleichbaren Elementen zu sortieren.
- Aus ALP2 wissen Sie, dass jeder vergleichsbasierte Sortieralgorithmus mindestens $\Omega(n \log n)$ Operationen benötigt. Folgern Sie, dass in jeder vergleichsbasierte Implementierung der Prioritätswarteschlange mindestens eine Operation $\Omega(\log n)$ amortisierte Laufzeit haben muss.