

Vorlesung “Objektorientierte Programmierung” — 2. Programmieretest (Aufgabe B) —

Hinweise/Regeln:

Vermeiden Sie bitte jedes Verhalten, was als Täuschungsversuch (miss-)verstanden werden könnte. Wir müssten Sie sonst disqualifizieren, d.h. Sie hätten den Programmieretest nicht bestanden. Sie werden sowohl durch das Aufsichtspersonal im Rechnerpool überwacht, als auch auf den Rechnern per Programm/Aufzeichnung, sowie über das Netz.

- Bearbeitungsdauer: 70 Minuten (plus eventuell leichte Verlängerung).
- Es gibt nur “bestanden” oder “nicht bestanden”, keine Punkte für partiell korrekte Lösungen. Eventuell werden am Ende einige Minuten Bearbeitungszeit angehängt, wenn mehrere Teilnehmer knapp vor einer fertigen Lösung sind — aber nicht funktionierende Lösungen werden nicht akzeptiert, selbst wenn nur eine Kleinigkeit fehlt.
- Sie dürfen bis zu 3 Blätter “Spickzettel” / “Quick Reference” verwenden, sowie ein Buch (nicht zu groß, es muß noch auf den Tisch passen ohne den Nachbarn zu stören — ein Aktenordner wäre nicht akzeptabel). Sie dürfen außerdem ein Blatt leeres Papier und einen Stift verwenden, um sich Notizen zu machen.
- Mobiltelefone bitte ausschalten (oder mit der Aufsicht besprechen).
- Sie dürfen ein eigenes Notebook verwenden, wenn Sie es bisher in den Übungen benutzt haben. WLAN und sonstige Netzverbindungen müssen ausgeschaltet sein.
- Es ist nicht erlaubt, einen Web-Browser zu verwenden, ein EMail-Programm, oder sonstige Netzwerk-Zugriffe. Automatische Netzwerk-Zugriffe Ihrer Entwicklungsumgebung sind in Ordnung, aber ggf. geöffnete Webseiten schließen Sie bitte sofort.
- Die automatisch eingeblendeten Methoden-Signaturen und andere Hinweise gehören zur Entwicklungsumgebung, aber ansonsten verwenden Sie die Hilfe bitte nicht (insbesondere keine Suchfunktionen).
- Sie dürfen nicht versuchen, Dateien außerhalb Ihres Homeverzeichnisses abzulegen, oder auf Dateien außerhalb Ihres Homeverzeichnisses zuzugreifen (außer Dateien, die der Compiler bzw. die Entwicklungsumgebung verwendet).
- Auch auf vorhandene Dateien innerhalb Ihres Homeverzeichnisses dürfen Sie nicht zugreifen (außer Voreinstellungen etc. durch die Entwicklungsumgebung). Sie müssen daher das Programm vollständig neu eintippen, und nicht durch Modifikation von eventuell in Ihrem Homeverzeichnis bereits vorhandenen Dateien erstellen.
- Auch vorhandene Text-Dokumente (z.B. PDF) dürfen Sie nicht anzeigen.
- Die Homeverzeichnisse werden für Zugriffe von außen gesperrt. Falls Sie spezielle Zugriffsrechte gesetzt hatten, müssen Sie diese nach dem Test selbst wieder herstellen.

- Tauschen Sie keinesfalls irgendwelche Dinge mit den Nachbarn aus. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Sie müssen Mindestanforderungen an den Programmierstil erfüllen, z.B. entsprechend der Programmstruktur einrücken.
- Fragen Sie, wenn Ihnen die Aufgabe nicht klar ist! Wenn Sie an einer unverständlichen Fehlermeldung länger festhängen, können Sie probieren, zu fragen. Wir wollen aber nicht zu viele Tipps geben.
- Wenn Sie glauben, fertig zu sein, melden Sie sich bitte zur Kontrolle. Sollten wir noch einen Fehler finden, können Sie bis zur offiziellen Abgabefrist weiterarbeiten.

Aufgabe (Variante B)

Schreiben Sie eine Klasse `Glaettung`, die zur Glättung von Messwerten dient, die in regelmäßigen Abständen ankommen. Die Klasse soll sich die jeweils 5 letzten Messwerte merken, und daraus den Mittelwert berechnen. Die Messwerte sind nicht-negative Gleitkommazahlen (Typ `double`). In einzelnen soll Ihre Klasse folgende Schnittstelle haben:

- Einen Konstruktor (ohne Parameter). Er muss einen Zustand herstellen, in dem noch kein Messwert gespeichert ist.
- Eine Methode `store` zum Speichern eines Messwertes. Diese Methode hat einen `double`-Parameter (für den Messwert) und keinen Rückgabewert. Solange noch nicht 5 Werte gespeichert sind, wird der Wert zusätzlich gespeichert (in einem Array), ansonsten wird der älteste Messwert überschrieben.
- Eine Methode `avg`, die den Durchschnitt der letzten Messwerte liefert (d.h. die Summe der Messwerte geteilt durch die Anzahl). Es ist möglich, dass die Anzahl kleiner als 5 ist, wenn noch nicht genügend Werte gespeichert wurden. Falls bisher noch gar kein Wert gespeichert wurde, soll `-1` geliefert werden. Diese Methode hat einen Rückgabewert vom Typ `double`, aber keine Parameter.

Achten Sie darauf, dass nur die erwähnten Methoden von außen zugreifbar sind, und zwar auch von außerhalb des Paketes. Attribute dürfen nicht von außen zugreifbar sein.

Auf der nächsten Seite ist der Programmcode einer `main`-Methode angegeben, die einige einfache Tests mit der Klasse `Glaettung` durchführt.

Eine mögliche Anwendung Ihrer Klasse sieht so aus:

Listing 1: Hauptprogramm

```
1 public class Test
2 {
3     public static void main(String [] args)
4     {
5         Glaettung g = new Glaettung ();
6         System.out.println(g.avg()); // Muss -1.0 ausgeben
7         g.store(10.0);
8         System.out.println(g.avg()); // Muss 10.0 ausgeben
9         g.store(20.0);
10        System.out.println(g.avg()); // Muss 15.0 ausgeben
11        for(i = 1; i <= 5; i++)
12            g.store(1.0);
13        System.out.println(g.avg()); // Muss 1.0 ausgeben
14    }
15 }
```

Tipps:

- Sie verwenden ein `double`-Array der Größe 5 für die Speicherung der Messwerte. Wenn Sie einmal gespeicherte Werte nicht verschieben wollen, brauchen Sie außerdem ein Attribut für den Index der Position, an die der nächste Wert gespeichert werden soll.
- Eventuell brauchen Sie zusätzlich noch ein Attribut für die Anzahl der im Array gespeicherten Werte. Diese beginnt mit 0, wächst dann bei jedem Speichervorgang bis auf 5, und bleibt dann bei 5 stehen.
- Eine Alternative wäre, diese Anzahl bei jeder Durchschnittsbildung zu berechnen, dann müssen Sie das Array z.B. auf -1 initialisieren, und negative Werte bei der Durchschnittsbildung nicht mitzählen.
- Beim Index für den zu speichernden Wert können Sie so vorgehen, dass, wenn er den maximal möglichen Wert erreicht, er anschließend wieder auf 0 gesetzt wird.
- Alternativ können Sie den Wert auch immer weiter hochzählen und zum Array-Zugriff den Divisionsrest mit dem Operator `%` verwenden.