

Prüfung Programmieren I für Björn Müßter

1. Semester AMP/IN/MIN/WIN - Bachelor

Im Wintersemester 2020/2021 am 17. Dezember 2020 ab 14.00 Uhr

Dauer: Donnerstag, 17.12., 14.00 Uhr bis Montag, 21.12., 08.00 Uhr.

Gesamtzahl aller Aufgabenblätter (inkl. Deckblatt): 4

Hinweise

Programmierung

- Sie dürfen für die Bearbeitung der Aufgaben nur die jeweils bei den Aufgaben angegebenen .NET Bibliotheksfunktionen verwenden! Falls keine angegeben sind, müssen Sie die Funktionalität entsprechend selbst programmieren! Ausnahmen sind die Funktionen der Klassen `Console`, `Random` und `Convert` (alternativ die Parse-Methode des Datentyps) sowie `Length` und `GetLength`.
- `Random` stellt einen Generator für Pseudozufallszahlen dar. `Random.Next` ist eine Methode, um zufällige ganze Zahlen zu erhalten.
- Die Verwendung von `goto` ist nicht zulässig und führt zu Punktabzug.

Organisation

- Programmierumgebung: **Visual Studio**, **.NET Core**
- Legen Sie **für jede zu programmierende Aufgabe eine eigene Solution** an.
- Benennen Sie die jeweilige Solution mit **Matrikelnummer_AufgabeX** (z.B. 123456_Aufgabe1).
- Laden Sie Ihre programmierten Lösungen im **zip-Format** hoch.
- Die Lösungen müssen innerhalb der angegebenen Prüfungszeit im Moodle hochgeladen werden. Das System nimmt nach dem Ende der verfügbaren Zeit keine Lösungen mehr an. Sie können auch keine Lösungen im Nachhinein per E-Mail einreichen. Planen Sie also bitte ausreichend Puffer für das Hochladen Ihrer Lösungen ein!
- Sie müssen bei jeder Aufgabe am Ende auf **Abgabe** klicken. Bei diesem Vorgang geben Sie Ihre **Selbstständigkeitserklärung** ab. Wenn Sie final auf Abgabe geklickt haben, dann können Sie an Ihrer Lösung nichts mehr ändern.
- Fragen, die im Laufe dieser Testklausur zu Modalitäten, Download und Upload aufgetaucht sind, werden ab Montagvormittag beantwortet. Bitte fragen Sie, wenn es Probleme oder Unklarheiten gab. Diese Testklausur soll helfen, dass Sie sich am Prüfungstag ganz auf den Inhalt der Prüfung konzentrieren können.

- Die Testklausur wird weder korrigiert noch bewertet. Nutzen Sie die Aufgaben zum Üben, und tauschen Sie sich selbständig über die Lösungsmöglichkeiten aus.

Plagiate

Plagiate führen zur Note 5 für alle Beteiligten!

Eine Prüfung, die wegen Unterschleifs mit 5 bewertet wird, gilt als angetreten. Dieser Versuch wird gezählt.

Punkteverteilung

Aufgabe	1	2	Summe
Punkte	28	30	58

Upload der Lösungen im Prüfungsmoodle

Die Abgabe erfolgt über ein Moodle (<https://elearning.ohmportal.de/course/view.php?id=8586>). Der Einschreibeschlüssel ist **Prog1TestWiSe20/21**.

Wir wünschen Ihnen viel Erfolg!

Aufgabe 1: Zahlen und Arrays

- (a) Füllen Sie ein zweidimensionales 5x5 Array zeilenweise mit Zufallszahlen zwischen 0 und 9. Die Klasse `Random` dürfen Sie dazu benutzen.

Es gelten folgende Regeln:

- Pro Zeile darf keine Ziffer doppelt vorkommen.
- Pro Spalte darf keine Ziffer doppelt vorkommen.

[15 Punkte]

- (b) Schreiben Sie eine Funktion zur Ausgabe eines beliebig großen zweidimensionalen Arrays in der Form:

```
1 1 1 1 1
1 1 1 1 1
...
```

Geben Sie damit Ihr Array aus.

[3 Punkte]

- (c) Schreiben Sie eine Funktion, die die Summe aller Elemente beider Diagonalen des Arrays berechnet, ohne eine Position doppelt zu zählen, und auf der Konsole ausgibt. Überprüfen Sie vorher, dass das Array quadratisch mit ungeradzahlgiger Kantenlänge ist. Wenn die Bedingungen nicht erfüllt sind, geben Sie 0 zurück.

[8 Punkte]

- (d) Rufen Sie alle Funktionen in Ihrer eigenen `main` auf. Verwenden Sie auch hier sprechende Namen.

[2 Punkte]

Aufgabe 2: Auslesen eines Log-Files

Sie haben Probleme mit Ihrer Webseite. Für die technischen Belange haben Sie einen Administrator, der von Ihnen Unterstützung bei der Fehlersuche braucht. Aus Datenschutzgründen dürfen nur Sie die gesamte Log-Datei anschauen.

Ein einzelner Eintrag einer Log-Datei sieht beispielsweise so aus:

```
[Fri Jul 17 06:02:13.295359 2020] [mpm_prefork:notice] [pid 968]
AH00163: Apache/2.4.18 (Ubuntu) OpenSSL/1.0.2g configured -- resuming
normal operations
```

In eckigen Klammern stehen die sogenannten Metadaten. Der Teil, der sich nicht in eckigen Klammern befindet, ist die eigentliche Log-Meldung. Die Log-Datei finden Sie im Ordner Aufgabe2.

Hinweise:

- Sie müssen nicht überprüfen, ob die Datei existiert.
 - Sie dürfen die Klasse `DateTime` nicht verwenden.
 - Vermeiden Sie redundanten Code.
 - Verwenden Sie sprechende Variablennamen.
- (a) Schreiben Sie ein Programm, das das Log-File liest, alle Zeilen mit der Prozess-ID 968 `[pid 968]` aus dem Logfile filtert und aus diesen Zeilen *Uhrzeit* (`std:min:sec`); *Meldung* in eine Ausgabedatei mit Namen `pid_968.txt` schreibt. Vergessen Sie nicht, die Dateien auch wieder zu schließen. Achten Sie auf die Dateipfade!
Beispiel: 06:02:13; AH00163: Apache/2.4.18 (Ubuntu) OpenSSL/1.0.2g configured – resuming normal operations

[15 Punkte]

- (b) Bestimmen Sie auch die Anzahl aller Zeilen, die einen Eintrag `error` enthalten und geben Sie die Anzahl über die Console aus.

[3 Punkte]

- (c) Berechnen Sie, wieviel Zeit (in Minuten) zwischen dem ersten und dem letzten Eintrag `[core:notice]` vergangen ist und geben Sie das Ergebnis aus. Sie dürfen davon ausgehen, dass die Einträge alle vom gleichen Tag sind.

[12 Punkte]