

Name:

Vorname:

Matrikelnr.:

Ihr Hauptfach:

Tutor:

Barbara

Lena

David

Pascal

Klausurzulassung: 2003

2004

Probeklausur zur Vorlesung Informatik B

(Dr. Frank Hoffmann)

Sommersemester 2005

13. Juli 2005

Beginn: 8²⁰, Ende: 10⁰⁰ (100 min)

1.	2.	3.	4.	Σ
/6	/6	/6	/6	/24

Außer Schreibutensilien sind keine Hilfsmittel erlaubt!

Auf diesem Klausurbogen ist genügend Platz, um die Lösungen der Aufgaben aufzuschreiben. Auch die Rückseiten der Blätter können verwendet werden (bitte auf der Vorderseite anmerken). Bitte geheftet lassen! **Zusätzliche lose Blätter** müssen mit der Matrikelnummer, Namen und Aufgabennummer versehen werden. Auf einem Zusatzblatt jeweils nur eine Aufgabe bearbeiten. Nicht mit Bleistift und nicht mit Rot schreiben. *Der Klausurbogen ist auf jeden Fall abzugeben!*

Viel Erfolg!

Aufgabe 1 Java

/6

(a) Betrachten Sie das folgende Codefragment:

```
int[] a = {6,5,4,3,2,1};
int[] b = a;
b[4] = 7;
int[] c = (int[])a.clone();
c[2] = 2*2;
c[3] = 4;
boolean x_ab = (a==b);
boolean x_ac = (a==c);
boolean z_1 = (a[2]==c[2]);
boolean z_2 = (a[3]==c[3]);
boolean z_3 = (b[4]==c[4]);
```

Welche Werte haben die folgenden Variablen?

x_ab =
x_ac =
z_1 =
z_2 =
z_3 =

(b) In einigen der folgenden Sätze sind Fehler verborgen. Finden Sie die falschen Sätze heraus! Versuchen Sie, die falschen Aussagen durch Änderung eines Worts oder einer Wendung in einen **sinnvollen, aussagekräftigen und wahren** Satz zu verwandeln.

Achtung: Bloße Negation ist hier nicht gefragt! Die falsche Aussage 'Der 7. Juni 2005 ist ein Mittwoch' sollte nicht durch 'Der 7. Juni 2005 ist kein Mittwoch', sondern durch 'Der 8. Juni 2005 ist ein Dienstag' korrigiert werden.

- 1) Jede nichtabstrakte Klasse besitzt einen parameterlosen Konstruktor. *wahr/falsch*
- 2) Eine Klasse kann mehrere Konstruktoren besitzen. *wahr/falsch*
- 3) Eine **private** Methode kann nicht überschrieben werden. *wahr/falsch*
- 4) Eine Methode , die in jeder Unterklasse sichtbar sein soll, muss **public** deklariert werden. *wahr/falsch*
- 5) Eine Methode in einem Interface darf nicht **protected** deklariert werden. *wahr/falsch*
- 6) Abstrakte Klassen können nicht instanziiert werden. *wahr/falsch*
- 7) Ein Interface kann Methoden seiner Unterklassen überschreiben. *wahr/falsch*

Aufgabe 2 Heaps

/6

- (a) Was ist ein Heap?
- (b) Wie kann man einen Heap in linearer Zeit bauen?
Illustrieren Sie die Konstruktion anhand der Schlüsselsequenz

(2, 5, 16, 4, 10, 23, 39, 18, 26, 15, 7, 9, 30, 31, 40)

Begründen Sie die lineare Laufzeit.

- (c) Vervollständigen Sie die folgende Aussage:
Für Min-Heaps mit n gespeicherten Elementen gilt: Die Operation `DeleteMin()` läuft in $\Omega(\log n)$, weil...

Aufgabe 3 Graphen

/6

- (a) Beschreiben Sie einen möglichst effizienten Algorithmus, der in einem gerichteten azyklischen Graphen $G = (V, E)$ die Länge eines längsten gerichteten Weges bestimmt. Was ist die Laufzeit? Begründung!
Hinweis: Topologisches Sortieren!
- (b) Testen Sie möglichst effizient (Laufzeit ?), ob ein ungerichteter Graph bipartit ist! Zur Erinnerung: Bipartit heißt, dass sich die Knotenmenge V in zwei disjunkte Mengen A und B zerlegen lässt, so dass alle Kanten des Graphen nur zwischen A und B verlaufen.

Aufgabe 4 Pseudocode

/6

Betrachten Sie das folgende Stück Pseudocode. Gegeben ist ein Array A von n positiven ganzen Zahlen und es wird ein anfangs leere Warteschlange Q benutzt.

```
t ← 0
for i ← 0 to n-1 do
  if A[i] ist ungerade Zahl then
    Q.enqueue()
  else
    while Q ist nicht leer do
      t ← t+Q.dequeue()
while Q ist nicht leer do t← t+Q.dequeue()
return t
```

- (a) Was gibt der Algorithmus bei dem Array $A = \{1, 9, 11, 14, 5, 3, 7, 13, 3, 12, 5\}$ aus?
- (b) Beschreiben Sie in einem Satz, was der Algorithmus tut!
- (c) Was ist die Laufzeit des Algorithmus in O -Notation in Abhängigkeit von n ? Begründung! .

