Hochschule -

Fakultät IV – Technische Informatik
Modul: Informatik
Professor: -

Portfolio

von

Sebastian Schramm Matrikel-Nr. -

9. Januar 2021

Inhaltsverzeichnis

1 Deckblatt

Deckblatt mit Ihrem vollständigen Namen, Matrikelnummer, Semester, Studiengang und Ihre unterschriebene Erklärung, dass Sie das Portfolio selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt haben. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.

Wel03 hello there

2 Was ist Informatik

Schreiben Sie einen Aufsatz zum Thema Was ist Informatik. Dieser Aufsatz sollte maximal 1500 Zeichen lang sein (ca. eine DIN A4 Seite).

3 Persönlichkeit der Informatik

Schreiben Sie einen Aufsatz über eine Persönlichkeit der Informatik. Dieser Aufsatz sollte maximal 1500 Zeichen lang sein (ca. eine DIN A4 Seite)

4 Wandeln Sie die folgenden Dezimalzahlen in das Hexadezimalsystem um: 2989, 57005, 48879.

- 2989 : 16 = 186 |Rest 2989 -16 * 486 = 13
- 186 : 16 = 11 |Rest 186 -16 * 11 = 10
- 11 : 16 = 0 |Rest 11

2989 = BAD

- 57005 : 16 = 3562|Rest 57005 -16 * 3562= 13
- 3562 : 16 = 222 |Rest 3562 -16 * 222 = 10
- 222 : 16 = 13 |Rest 222 -16 * 13 = 14
- 13 : 16 = 0 |Rest 13

57005 = DEAD

- 48879 : 16 = 3054|Rest 48879 -16 * 3054= 15
- 3054 : 16 = 190 |Rest 3054 -16 * 190 = 14
- 190 : 16 = 11 |Rest 190 -16 * 11 = 14
- 11 : 16 = 0 |Rest 11

48879 = BEEF

5 Wie zeigen Sie, die Äquivalenz der folgenden Ausdrücke mit XOR

$(x \vee y) \wedge \neg(x \wedge y)$

x	y	$(x \vee y)$	$\neg(x \wedge y)$	$a \wedge b$
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

$\neg((x \wedge y) \vee (\neg x \wedge \neg y))$

x	y	(x∧y)	(¬x∧¬y)	¬(a∨b)
0	0	0	1	0
0	1	0	0	1
1	0	0	0	1
1	1	1	0	0

¬(x∧y)∧(x∨y)

x	y	¬(x∧y)	(x∨y)	a∧b
0	0	0	1	0
0	1	1	1	1
1	0	1	1	1
1	1	1	0	0

6 UML

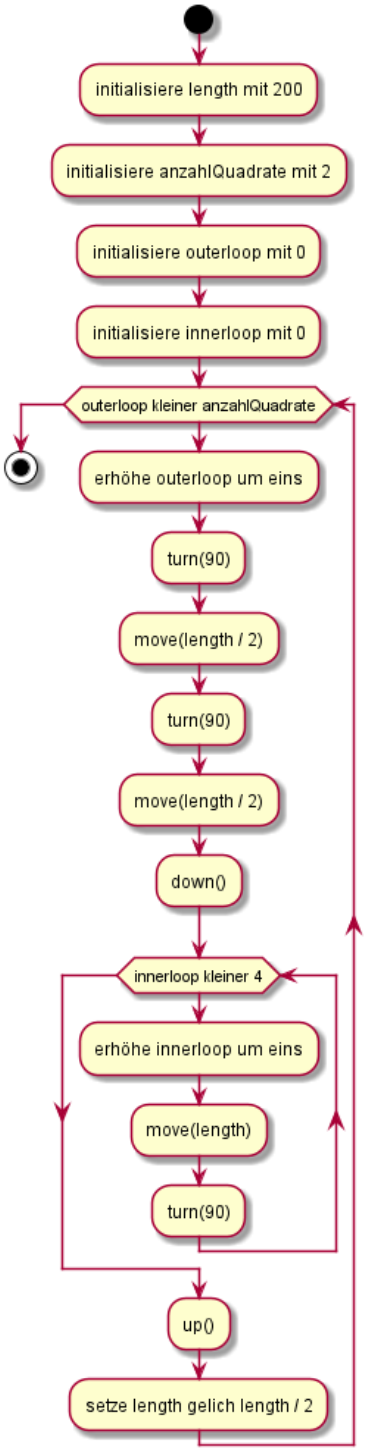
Schreiben Sie einen Aufsatz zum Thema UML. Dieser Aufsatz sollte maximal 3000 Zeichen lang sein (ca. zwei DIN A4 Seiten)

Währen der Blüte der Objekt orientierten Programmierung in den 1990er Jahren, kamen vermehrt Vorschläge für eine Modellierungssprache. Dazu haben sich die drei Grady Booch, Ivar Jacobson und James Rumbaugh. Welche schon ihre eigenen Modellierungssprachen entworfen hatten, sich zusammensetzten und daraus ein Konzept der UML entwickelten. Die UML oder auch Unified Modeling Language ist eine Objektorientierte Sprache und besitzt Notationen zur Beschreibung von Softwaresystemen. Der Grundgedanke hierbei, ist eine Einheitliche Notation für alle Softwaresysteme. Sie dient zur Spezifikation, Konstruktion, Dokumentation und Visualisierung von Software-Teilen und anderen Systemen (Wikipedia, 2020). In der UML gibt es Notationselemente, aus denen sich verschiedene Diagramme erzeugen lassen. Dabei verdeutlicht jedes Diagramm eine bestimmte perspektive auf das zu modellierende System und kann dem Entwickler, oder Nutzer einen guten Überblick auf die Software verschaffen. Es gibt verschiedene Modellierungsarten, wie das Aktivitätsdiagramm welches die Ablaufmöglichkeiten eines Systems beziehungsweise Anwendungsfalls mit Aktionen darstellt. Bei einem Anwendungsfalldiagramm oder auch Use Case Diagram, wird das Verhalten eines Systems aus der Benutzersicht dargestellt. Diese beschreiben aber auch die geplante Funktionalität eines Systems. Bei einem Klassendiagramm werden die Beziehungen zwischen den einzenlnen Klassen verdeutlich sowie, welche Attribute und Methoden es in der jeweiligen klasse vorhanden sind. Zusätzlich ist noch ersichtlich um welche datentypen es sich handelt und welche Attribute/Methoden public, private oder protected sind. Ein Zustandsdiagramm beschreibt den Lebenszyklus der Objekte einer Klasse

Wikipedia (19. Oktober 2020). Unified Modeling Language. Abgerufen am 5. Januar 2020 unt
"https://de.wikipedia.org/wiki/Unified_Modeling_Language"

7 zweiKonzentrischeQuadrate

Damit dieses Programm funktioniert gehen dir davon aus, dass wir oben Links starten. Also bei 0,0 und unser Startrichtung nach rechts(y) verläuft. Zusätzlich kann sich das Programm merken, in welche richtung sich der Kopf zuletzt bewegt hat.



8 IntStack

Schreiben Sie eine Java-Klasse IntStack, die einen ADT Stack implementiert. Die GröSSe (Kapazität) des Stacks soll fix sein und bei der Erzeugung eines IntStack-Objekts festgelegt werden. Intern sollen die Daten in einem int-Array abgelegt werden. Geben Sie den Source-Code ab.

8.1 Main.java

```
1 package aufgabe_08;
2
```

```java
import aufgabe_08.IntStack;

public class Main {

    public static void main(String[] args) {
        IntStack stack = new IntStack(32);
        stack.print();
        System.out.println(stack.empty());
        stack.push(12);
        stack.push(2);
        for (int i = 0; i < 12; ++i)
            stack.push((int) (Math.random()*100));
        stack.print();
        System.out.println(stack.empty());
        System.out.println(stack.full());
        stack.pop();
        stack.pop();
        stack.pop();
        stack.pop();
        stack.print();
        for (int i = 0; i < 12; ++i)
            stack.push((int) (Math.random()*100));
        System.out.println(stack.empty());
        System.out.println(stack.full());
        stack.print();
        for (int i = 0; i < 12; ++i)
            stack.push((int) (Math.random()*100));
        System.out.println(stack.empty());
        System.out.println(stack.full());
        stack.print();

        for (int i = 0; i < 40; ++i)
            stack.pop();

        stack.push(3);
        stack.print();
    }
}
```

8.2 IntStack.java

```java
package aufgabe_08;

public class IntStack {
    private int[] stack;
    private int stackPos;
    private int stackSize;

    public IntStack(int size) {
        init(size);
        stackPos = 0;
        stackSize = size;
    }

    public void init(int size) {
        stack = new int[stackSize];
    }

    public boolean empty() {
        return (stackPos == 0);
    }

```

```
22      public boolean full() {
23          return (stackPos >= stackSize);
24      }
25
26      public void push(int number) {
27          if (!full())
28              stack[stackPos++] = number;
29      }
30
31      public void pop() {
32          if (!empty())
33              stack[--stackPos] = 0;
34      }
35
36      public int top() {
37          return stack[stackPos];
38      }
39
40      public void print() {
41          for (int tmp : stack)
42              System.out.print(tmp + ", ");
43          System.out.println();
44      }
45  }
```

9 Aritmetischer Ausdruck

Preoder: *+a/bc-d*ef
Inorder: a+b/c*d-e*f
Postorder: abc/+def*-*