Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

Programmiertechnik II
Übung 5
23.06.2016

Maximilian Jenders

Aus aktuellem Anlass

- Aufgaben für Zweier-Teams ausgelegt.
- Zur Hilfe ein Ampelsystem:

Jede Aufgabe wird zusammen gelöst

- Gewünscht (Optimalfall)

Aufgabenteilung: Person A macht Aufgabe 1, Person B Aufgabe 2

- Nicht toll – schlechterer Lerneffekt

Einfach die Lösung eines anderen Teams nehmen

- Fällt auf (ja, wirklich. Auch bei 100 Studenten)! Mehr Stress für uns und euch.

Hausaufgabe 3
Nachtrag

Aufgabe 4

- Kleine Zeitintervalle mit Systemuhr gemessen

nanoTime

```
public static long nanoTime()
```

Returns the current value of the most precise available system timer, in nanoseconds.

This method can only be used to measure elapsed time and is not related to any other notion of system or wall-clock time. The value returned represents nanoseconds since some fixed but arbitrary time (perhaps in the future, so values may be negative). This method provides nanosecond precision, but not necessarily nanosecond accuracy. No guarantees are made about how frequently values change. Differences in successive calls that span greater than approximately 292 years (2^{63} nanoseconds) will not accurately compute elapsed time due to numerical overflow.

- Wie schon letztes mal gesagt: vielfach darüber iterieren!

Zusatz

- Meist gut gelöst, Kommentare oft mangelhaft

Rückblick Hausaufgabe 4

HA 4
Aufgabe 1 (Trip to Reno)

- b) häufig nicht bearbeitet oder falsch / ungenau
  - Algorithmen die, die als O(n) identifiziert wurden waren O(n^2) da sie einen versteckten oder Helfer Loop enthielten: Loop war in anderer Funktion, Array mit Länge in O(n) wurde kopiert.

- c) zu spät in float konvertiert
  - Integer-Division, falsches Ergebnis

HA 4
Aufgabe 2 (Dateien sortieren)

- Meist gut bearbeitet, öfters gute Kommentare / Exception Handling

- Fehler:
  - Lokale Dateipfade („C:\Users\$Name\Desktop\PT\fruits.txt“)
  - Teils keine Kommentare oder auto-generierte Kommentare mit TODO
  - Teils Methoden mit 100+ Zeilen
  - UTF-8 Encoding ignoriert
  - MaxLineCount ignoriert

HA 4
Aufgabe 3 (Sudoku)

- Unübersichtliche Bezeichner
  - *startM*, *startN*, *x*, *y*, *x2*, *y2*, *m*, *k*, *b*
- Inkonsistente Einrückungen:
- (…) {…} vs (…)\n{…}, Spaces vs Tabs
- Sehr verschachtelte Strukturen
- Rekursiver Ansatz zu ineffizient → Zeitüberschreitung

- Generell aber Backtracking gut verstanden, gute Lösungen

Generelles

Wünsche für letzes Tutorium?

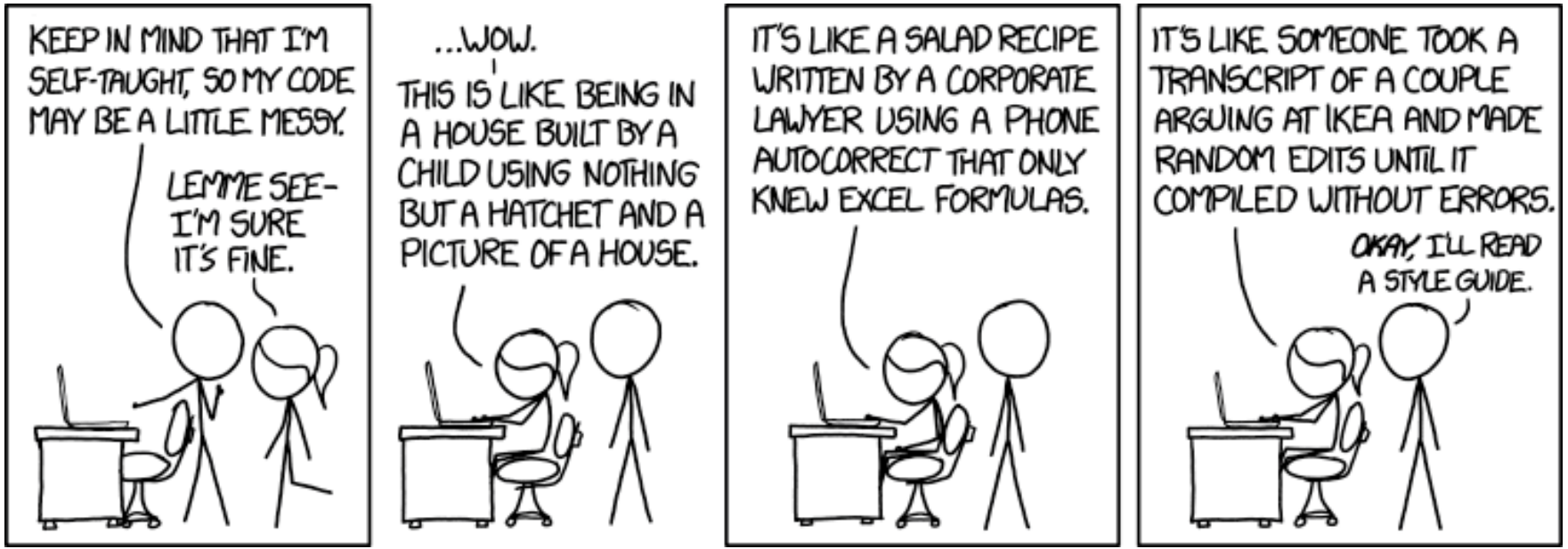
- Themen über die ihr noch etwas hören möchtet?
  - Muss nicht über Vorlesung sein
  - Vorschläge gerne jetzt oder per Email
- Noch thematische Fragen?
  - Kann ich spontan probieren zu beantworten, mit Vorbereitung natürlich einfacher / ausführlicher

- Schon vorab:
  - Klausur 02.08.2016 (Dienstag)
  - Ich bin im Urlaub vom 15.07.2016 – 24.07.2016, lese keine Emails
  - Danach / Davor beantworte ich gerne noch Fragen – aber bitte **frühzeitig** lernen / überlegen, nicht erst am Tag / Wochenende vor der Klausur

Hasso Plattner Institut

Codequalität

Generelles – Naming Conventions



- https://xkcd.com/1513/

Generelles – Naming Conventions

Hasso Plattner Institut

Welchen hab ich mir ausgedacht?

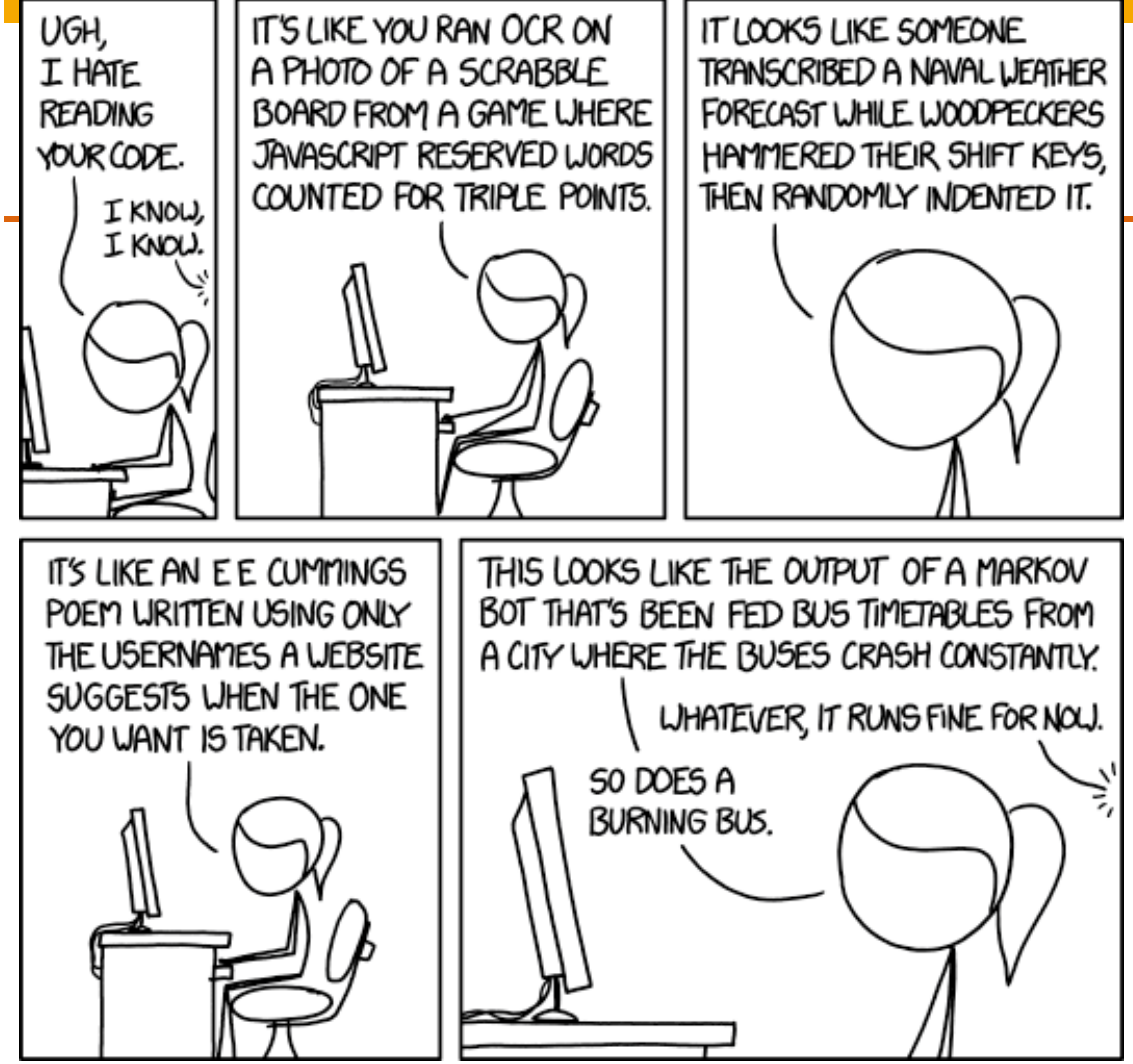
- RequestProcessorFactoryFactory
- SimpleBeanFactoryAwareAspectInstanceFactory
- InternalFrameTitlePaneMaximizeButtonWindowNotFocusedState
- HasThisTypePatternTriedToSneakInSomeGenericOrParameterizedTypePatternMatchingStuffAnywhereVisitor

Generelles – Naming Conventions

Hasso Plattner Institut

Keinen!

- RequestProcessorFactoryFactory Apache XML-RPC
- SimpleBeanFactoryAwareAspectInstanceFactory Spring
- InternalFrameTitlePaneMaximizeButtonWindowNotFocusedState Java Standardlib (AWT)
- HasThisTypePatternTriedToSneakInSomeGenericOrParameterizedTypePatternMatchingStuffAnywhereVisitor AspectJ Compiler



- https://xkcd.com/1695/

Garbage Collection

Generelles
(Nachtrag) Garbage Collection

- Schon letztes Mal angesprochen:

- Reference Counting GC
  - Keine Referenzen mehr auf Objekt? Tschüss!
  - Problem: Zyklische Referenzen

- Mark-and-Sweep GC
  - Start bei erreichbaren Objekten (Root set), alle Referenzen verfolgen (rekursiv)
  - Nicht erreichte (markierte) Objekte können gelöscht werden

Generelles
(Nachtrag) Garbage Collection

- Generational GC
  - Wie bei Game of Thrones: Nur wenige werden alt, viele sterben (zu?) früh
  - Separate Heapabschnitte für short/medium/long-living

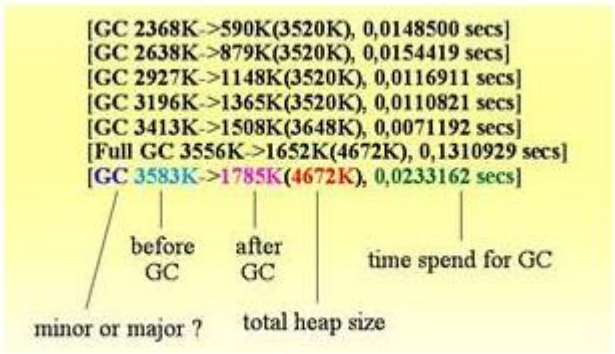


Abbildung 2: Garbage Collection Information per JVM-Option -verbose:GC

http://www.angelikalanger.com/Articles/EffectiveJava/26.GarbageCollection/26.GarbageCollection.html

Generelles
(Nachtrag) Garbage Collection

- „Junger“ Bereich (eden) ändert sich oft; stark fragmentiert

- Copy-GC:
  - 2 „Survivor“-Bereiche, einer immer passiv (leer)
  - Überlebende junge Objekte werden in Survivor-Bereich koopiert
  - Simpel, aber viel reservierter Speicher
- Mark-and-Compact
  - Objekte in-place defragmentieren
  - Effizient, wenn es wenige Überlebende gibt

Generelles
(Nachtrag) Garbage Collection

- Über Startup-Parameter konfigurierbar, z.B.:
- -Xms*n*
  - Specifies the initial size, in bytes, of the memory allocation pool
- -Xmx*n*
  - Specifies the maximum size, in bytes, of the memory allocation pool
- -Xmnsize or -XX:NewSize
  - Sets the size of the young generation (nursery).
- -XX:+UseConcMarkSweepGC or -XX:+UseG1GC
  - Enables either the Concurrent Mark Sweep (CMS) or the G1 garbage collectors.
- -verbose:gc
  - Reports on each garbage collection event.

Reguläre Ausdrücke

Reguläre Ausdrücke

- Wer hat schon Regex benutzt?

- „A sequence of characters that define a search pattern“
  - Pattern matching

- Viele Daten, relativ feste Struktur.

- Boolean: „|“
- Grouping: gr(a|e)y
- Quantification: H(ae?|ä)ndel == H(a|ae|ä)ndel

Anchors

^	Start of string, or start of line in multi-line pattern
\A	Start of string
$	End of string, or end of line in multi-line pattern
\Z	End of string
\b	Word boundary
\B	Not word boundary
\<	Start of word
\>	End of word

Character Classes

\c	Control character
\s	White space
\S	Not white space
\d	Digit
\D	Not digit
\w	Word
\W	Not word
\x	Hexadecimal digit
\O	Octal digit

POSIX

Quantifiers

*	0 or more	{3}	Exactly 3
+	1 or more	{3,}	3 or more
?	0 or 1	{3,5}	3, 4 or 5

Add a ? to a quantifier to make it ungreedy.

Escape Sequences

\	Escape following character
\Q	Begin literal sequence
\E	End literal sequence

"Escaping" is a way of treating characters which have a special meaning in regular expressions literally, rather than as special characters.

Common Metacharacters

^	[	.	$
{	*	(	\
+	)	|	?
<	>		

The escape character is usually \

Special Characters

\n New line

Groups and Ranges

.	Any character except new line (\n)
(a|b)	a or b
(...)	Group
(?:...)	Passive (non-capturing) group
[abc]	Range (a or b or c)
[^abc]	Not (a or b or c)
[a-q]	Lower case letter from a to q
[A-Q]	Upper case letter from A to Q
[0-7]	Digit from 0 to 7
\x	Group/subpattern number "x"

Ranges are inclusive.

Pattern Modifiers

g	Global match
i *	Case-insensitive
m *	Multiple lines
s *	Treat string as single line
x *	Allow comments and whitespace in pattern
e *	Evaluate replacement
U *	Ungreedy pattern

* PCRE modifier

https://www.cheatography.com/davechild/cheat-sheets/regular-expressions/

HPI Hasso Plattner Institut


WHENEVER I LEARN A NEW SKILL I CONCOCT ELABORATE FANTASY SCENARIOS WHERE IT LETS ME SAVE THE DAY.
OH NO! THE KILLER MUST HAVE FOLLOWED HER ON VACATION!
BUT TO FIND THEM WE'D HAVE TO SEARCH THROUGH 200 MB OF EMAILS LOOKING FOR SOMETHING FORMATTED LIKE AN ADDRESS!
IT'S HOPELESS!
EVERYBODY STAND BACK.
I KNOW REGULAR EXPRESSIONS.
PERL!
TAP TAP

Reguläre Ausdrücke

- Beispiele:
- IP-Adressen :
  - \b\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\b
- Mailadressen
  - ^[A-Z0-9._%+-]+@[A-Z0-9.-]+\.[A-Z]{2,}$
  - Findet aber nicht alle Email-Adressen

The official standard is known as RFC 5322. It describes the syntax that valid email addresses must adhere to. You can (but you shouldn't—read on) implement it with the following regular expression. RFC 5322 leaves the domain name part open to implementation-specific choices that won't work on the Internet today. The regex implements the "preferred" syntax from RFC 1035 which is one of the recommendations in RFC 5322:

```
\A(?:[a-z0-9!#$%&'*+/=?^_`{|}~-]+(?:\.[a-z0-9!#$%&'*+/=?^_`{|}~-]+)*
  |  "(?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21\x23-\x5b\x5d-\x7f]
        |  \\[\x01-\x09\x0b\x0c\x0e-\x7f])*")
@ (?:(?:[a-z0-9](?:[a-z0-9-]*[a-z0-9])?\.)+[a-z0-9](?:[a-z0-9-]*[a-z0-9])?
  |  \[(?:(?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}
        (?:25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?|[a-z0-9-]*[a-z0-9]:
           (?:[\x01-\x08\x0b\x0c\x0e-\x1f\x21-\x5a\x53-\x7f]
           |  \\[\x01-\x09\x0b\x0c\x0e-\x7f])+)
     \])\z
```

http://www.regular-expressions.info/email.html

Reguläre Ausdrücke

- Regex nützlich, aber…
  - Können schnell „explodieren“
  - Laufzeit kann sehr schlecht sein, wird auf alles gematcht
  - Wartbarkeit, Lesbarkeit – Kommentare!
  - Ausnahmen? Z.B. URL Matching: https, http, mit oder ohne www?
- Nicht für alles anwendbar, siehe StackOverflow:
  - „(HTML) match open tags except self-contained tags“
  - https://stackoverflow.com/questions/1732348/regex-match-open-tags-except-xhtml-self-contained-tags/1732454#1732454

You can't parse [X]HTML with regex. Because HTML can't be parsed by regex. Regex is not a tool that can be used to correctly parse HTML. As I have answered in HTML-and-regex questions here so many times before, the use of regex will not allow you to consume HTML. Regular expressions are a tool that is insufficiently sophisticated to understand the constructs employed by HTML. HTML is not a regular language and hence cannot be parsed by regular expressions. Regex queries are not equipped to break down HTML into its meaningful parts. so many times but it is not getting to me. Even enhanced irregular regular expressions as used by Perl are not up to the task of parsing HTML. You will never make me crack. HTML is a language of sufficient complexity that it cannot be parsed by regular expressions. Even Jon Skeet cannot parse HTML using regular expressions. Every time you attempt to parse HTML with regular expressions, the unholy child weeps the blood of virgins, and Russian hackers pwn your webapp. Parsing HTML with regex summons tainted souls into the realm of the living. HTML and regex go together like love, marriage, and ritual infanticide. The cannot hold it is too late. The force of regex and HTML together in the same conceptual space will destroy your mind like so much watery putty. If you parse HTML with regex you are giving in to Them and their blasphemous ways which doom us all to inhuman toil for the One whose Name cannot be expressed in the Basic Multilingual Plane, he comes. HTML-plus-regexp will liquify the nerves of the sentient whilst you observe, your psyche withering in the onslaught of horror. Regex-based HTML parsers are the cancer that is killing StackOverflow *it is too late it is too late we cannot be saved* the trangession of a child ensures regex will consume all living tissue (except for HTML which it cannot, as previously prophesied) *dear lord help us how can anyone survive this scourge* using regex to parse HTML has doomed humanity to an eternity of dread torture and security holes *using regex as a tool to process HTML establishes a breach between this world* and the dread realm of corrupt entities (like SGML entities, but *more corrupt) a mere glimpse* of the world of reg**ex parsers for HTML will ins**tantly transport a *programmer's consciousness i*nto a wo*r*ld of ceaseless screaming, he comes, ~~the pestilent sli~~thy regex-infection will **devour your HT**ML parser, application and existence for all time like Visual Basic only worse *he comes he comes do not fi*ght h**e comes, hi**s unholy radiańcé de*stroying all enlightenment, HTML tags **leaking from your eyes like liq**uid p*ain, the song of regular expr*~~ession parsing~~ will ext*inguish the voices of mor**tal man from the sp**here I can see it can you see it it is beautiful* the f`inal snuf` fing o*f the lie**s of Man ALL IS LOST ALL IS LO**ST the *pony he come*s he ~~comes~~ ~~he comes the ich~~or permeate*s a*ll MY FAC*E MY FACE oh god no NO NOOOO N*Θ stop *the an*gles `are` **not real ZALGO IS TONY THE PONY HE COMES**

Have you tried using an XML parser instead?