

Repetition

Programmering för språkteknologer 2

Sara Stymne

2013-10-04

Innehåll

- ▶ Gott och blandat
- ▶ Repetition

?:-operatorn

- ▶ Alternativ till if i vissa kontexter
- ▶ Kallat det villkorliga uttrycket (ej en sats)
- ▶ villkor ? if-true : if-false
- ▶ Exempel:
 - ▶ `String s = (x>1 ? "flera" : "en");`
 - ▶ `return (x > 0 ? x : -x);`
 - ▶ `return (x != null ? x.value : "UNK");`

Slumptal

- ▶ Ej riktiga slumptal – beräknas med en algoritm, baserat på ett seed-värde
- ▶ Slumptalsgeneratorn bör initialiseras EN gång, innan den används för att ta fram slumptal

KODEXEMPEL

i++ / ++i

- ▶ Prefix, ++i, returnerar det ökade värdet
- ▶ Postfix, i++, returnerar originalvärdet
- ▶ Exempel:
 - ▶ `int i = 5;`
 - ▶ `System.out.println(++i);`
 - ▶ `System.out.println(i++);`

Objektorientering

- ▶ Arv
- ▶ Polymorfism
- ▶ Abstrakta klasser
- ▶ Gränssnitt
- ▶ Synlighet

Objektorientering – kodbegrepp

- ▶ class, interface
- ▶ this
- ▶ super
- ▶ private, protected, public
- ▶ abstract
- ▶ static
- ▶ extends, implements

Regulära uttryck och automater

- ▶ Sätt att beskriva strängar
- ▶ Ni ska kunna tolka och skapa uttryck och automater
- ▶ Ni ska kunna använda reguljära uttryck i Java

KODEXEMPEL – ANVÄNDNING I JAVA

Sökning

- ▶ Linjär sökning
- ▶ Binär sökning

Sortering

- ▶ Selection sort
- ▶ Bubble sort
- ▶ Insertion sort
- ▶ Shell sort
- ▶ Quick sort
- ▶ Merge sort

Länkade listor

- ▶ Länkad struktur:
 - ▶ Listklassen känner till första noden
 - ▶ Varje nod känner till nästa nod
- ▶ Varianter:
 - ▶ Enkellänkad lista
 - ▶ Enkellänkad lista med last-pekare
 - ▶ Dubbellänkad lista
 - ▶ Länkade träd
- ▶ Iteratorer

Stackar/Köer

- ▶ Stack – sist in först ut
- ▶ Kö – först in först ut
- ▶ Kan implementeras med hjälp av:
 - ▶ Länkade strukturer
 - ▶ Fält (arrayer)

Undantag

- ▶ Ett sätt att hantera fel som uppstår i program
- ▶ Relevanta aspekter:
 - ▶ Kasta undantag när något blir fel
 - ▶ Fånga och hantera undantag
 - ▶ Skapa egna undantagsklasser

Mappningar

- ▶ Hashtabeller
 - ▶ Hashkod
 - ▶ Krockhantering
- ▶ Binära sökträd
 - ▶ Obalanserade: insättning/sökning
 - ▶ Balanserade: förstå principen (inga detaljer)

Komplexitet

- ▶ Kunna föra grundläggande resonemang kring komplexitet
- ▶ Kunna reodgöra för komplexitet för
 - ▶ Sökalgoritmer
 - ▶ Sorteringsalgoritmer
 - ▶ Vanliga operationer på datastrukturer:
 - ▶ Lista (länkad+fält)
 - ▶ Stack
 - ▶ Kö
 - ▶ Hashtabell
 - ▶ Binära sökträd

Vad jag hoppas ni fått ut av kursen

- ▶ Blivit bättre programmerare
 - ▶ Lärt er mer programmeringskoncept
 - ▶ Fått programmera ganska mycket själva
 - ▶ Koda från scratch, dela upp i klasser, metoder
- ▶ Fått bättre teoretisk grund för att bli bra programmerare
 - ▶ Kunna diskutera kring komplexitet
 - ▶ Förstå och kunna välja lämpliga datastrukturer
 - ▶ Ha kännedom om vanliga algoritmer
- ▶ Programmering för språkteknologer
 - ▶ Allmänt bättre på programmering!
 - ▶ Kunna använda reguljära uttryck i Java

Hur ni kan jobba vidare med programmering

- ▶ Programmera, programmera, programmera!
- ▶ Lär er om mer datastrukturer och algoritmer
- ▶ Designmönster
- ▶ Engagera er i open source projekt
- ▶ Skapa portfolio med det ni programmerat (tex spellabben)
- ▶ Lär er fler programmeringsspråk

Kvar att göra

- ▶ De sista labbarna
 - ▶ Slutdeadline: 8/11
 - ▶ Uppsamlingsdeadline: 6/12
- ▶ Tenta 8/11
 - ▶ Lycka till!
- ▶ Kursutvärdering
 - ▶ Viktigt att fylla i
 - ▶ Länk finns på hemsidan