

Laborationsrapport
Algoritmer och datastrukturer I , 7,5 HP
HT2015

Laboration nr: 1
De abstrakta datatyperna Stack och Kö

av

Viktor Hanstorp (1994-04-13)

Institutionen för matematik, natur- och datavetenskap
Högskolan i Gävle

S-801 76 Gävle, Sweden

Email:
ndi14vhp@student.hig.se

Innehåll

1	Inledning.....	2
2	Förutsättningar och krav	2
3	Implementering och test.....	2
3.1	Systembeskrivning	2
3.2	Tester.....	3
4	Diskussion.....	4
4.1	Diskussion av resultatet.....	4
4.2	Diskussion av genomförandet	4
5	Referenser.....	4
6	Bilagor.....	4

1 Inledning

Laborationen går ut på att man skall bekanta sig med abstrakta datatyper i allmänhet och stack, kö och lista i synnerhet, samt för att fortsätta med JUnit test.

Den är skapad för att få en mjuk övergång från kursen "Objektorienterad design och programmering".

2 Förutsättningar och krav

Programmet skall kunna avgöra om en inmatad sträng är parentes balanserad eller inte.

Det skall även skapas test för programmet, samt StackList och QueueList.

Ett eget interface för Queue skall skapas.

3 Implementering och test

För att mata in parentes uttryck så skapades ett GUI där man kan skriva in det uttryck som man vill testa. Efter att man har skrivit in uttrycket så trycker man på knappen "check" för att se om det är balanserat. I en textpanel så markeras vilka parenteser som skapar obalanseringen med röd färg och en popup visas där det står om det är balanserat.

För att matcha parenteser så stegas strängen igenom så som instruktionerna instruerade, dock så samlas information om parentestyp, position i strängen och vilka parenteser som utgör par.

Denna data returneras senare i ett resultatobjekt för användning vid markering av felaktiga parenteser.

Att nämna är att programmet inte avslutas vid inmatad tom-rad, då det är ett balanserat parentes-uttryck (utan parenteser, ändock lika många höger och vänster parenteser; inga), det avslutas istället med den intuitiva stängknappen.

3.1 Systembeskrivning

Information om parenteser samlas i klassen "Parenthesis":

Parenthesis

final boolean open

final int position

"open" markerar om det är en höger eller vänster parentes och "position" lagrar parentesens position i strängen.

De är markerade "public final" istället för att använda getters.

I metoden "checkBalance" i klassen "ParenthesisBalanceChecker" returnerar ett "Result" objekt med information om det är balans i uttrycket.

Result

final boolean balanced

final HashMap<Parenthesis, Parenthesis> mappings

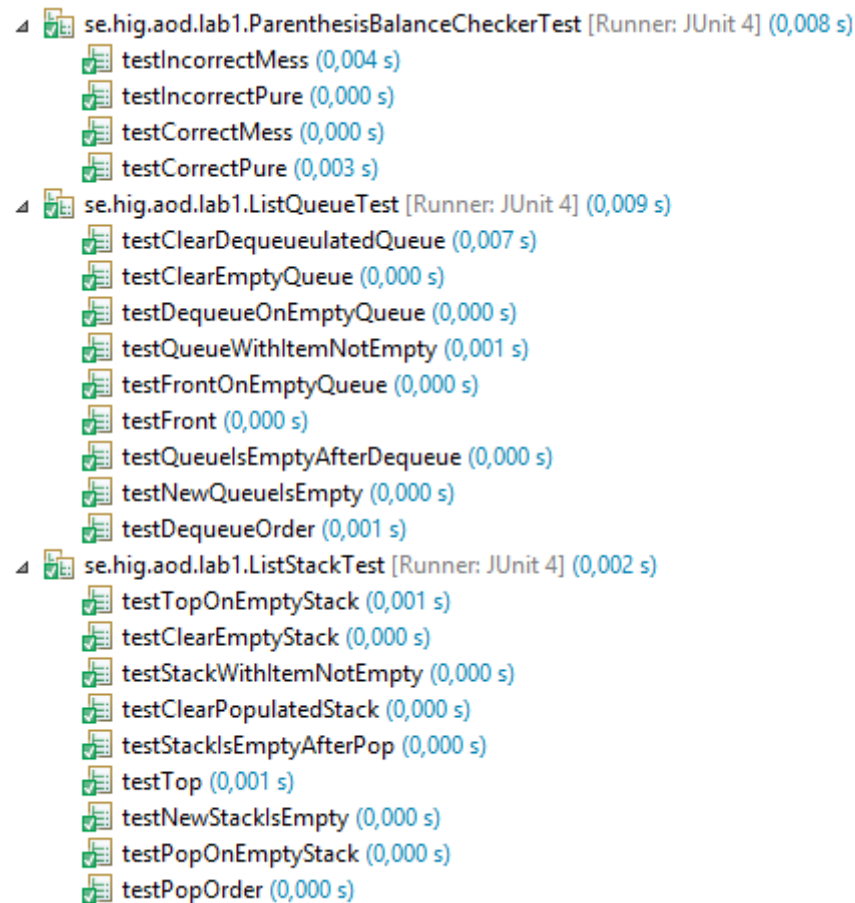
final int[] errorAt

”balanced” markerar om uttrycket är balansera, ”mappings” innehåller paren av parenteser (om uttrycket matchar) och ”errorAt” har positionerna för parenteserna som obalanserar uttrycket (om de finns).

3.2 Tester

Testerna är gjorda i JUnit och testar möjliga in-data och funktionsanrop för att försäkra sig om att det blir rätt resultat.

Testerna fungerar (se Figur 1):



Figur 1 - Testerna fungerar

”ParenthesisBalanceCheckerTest” testar ”ParenthesisBalanceChecker” genom att mata in olika parantes-uttryck och bekräftar att de ger rätt resultat.

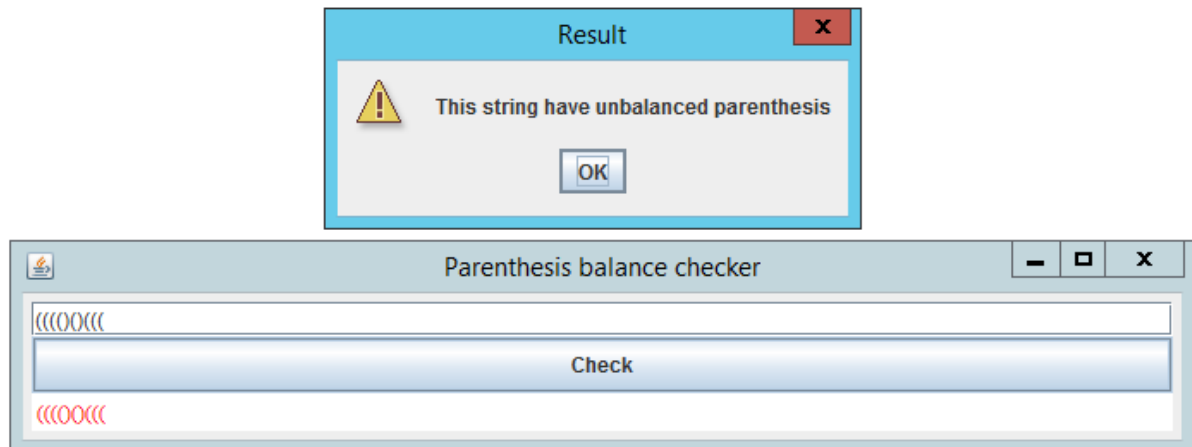
”ListQueueTest” testar ”ListQueue” genom att utföra de olika operationerna som kön erbjuder.

”ListStackTest” testar ”ListStack” även här genom att utföra de olika operationer so erbjuds.

4 Diskussion

4.1 Diskussion av resultatet

Produkten är ett grafik-baserat program som tillåter användaren att mata in parentes-uttryck, vilket sedan kan kontrolleras för att se om det är balanserat. I det fall det inte är balanserat så märks de parenteser som get upphov till obalansen ut med röd färg.(se Figur 2)



Figur 2 - Resultat

Kravspecifikationen kräver att man skall kunna kontrollera om ett parentes-uttryck är balanserat; vilket programmet gör.

Eventuella förbättringar skulle kunna vara att göra det lättare att hålla koll på vad som är i parentes-uttrycken så att det kan bli enklare att agera på datan. Eventuellt också att ge förslag på hur man skall balansera uttrycket.

4.2 Diskussion av genomförandet

Det var nära att det bar iväg och ytterligare kod som inte bidrog till att uppfylla kravspecifikationen lades till, dock så lyckades det undvikas.

Rapporten skulle eventuellt kunnat göras färdigt lite tidigare.

Ingen interaktion krävdes under handledningen, så att vara på plats kan eventuellt ses som slöseri på tid. Vid efterföljande labbar så kan det övervägas att jobba hemifrån.

5 Referenser

Inspiration för att lösa vissa uppgifter har hämtats från internet, referens till källa kan hittas vid gällande avsnitt i koden.

6 Bilagor

Källkod finns i bifogad folder.