Linköpings Universitet
Institutionen för datateknik (IDA)
UPP-gruppen 2012-12-10

Programmeringsstuga 3

Programspecifikation - Stack

En rekryterare på Ericsson har tagit fram en liten testuppgift som han ger alla sökande att lösa i samband med intervjun (medan han diskuterar med kollegor vilket intryck den sökande gett så långt). Han räknar med att en duktig C++-programmerare med examen löser uppgiften på mindre än en kvart. Uppgiften är att implementera bästa möjliga stack utan att använda standardbiblioteket (han vill ju se hur de sökande hanterar minne och pekare eftersom det är ett mycket viktigt kriterie i rekryteringen).

En stack är en datastruktur där man kan lägga till och ta bort dataelement. Konceptuellt fungerar stacken som en stapel med saker. Initialt är den tom. Varje sak som läggs till hamnar överst i stapeln, och saker som tas bort tas överst (annars rasar ju stapeln). Detta gör att stacken får egenskapen "först in - sist ut", vilket kan nyttjas i mågna algoritmer. Normalt återfinns följande funktioner på ett objekt som representerar en stack:

push Lägger till ett dataelement överst i stacken.

top Returnerar överta dataelementet utan att ta bort det. Det är lämpligt att kunna använda denna även om stacken är konstant. För effektivitets skull är det lämpligt att returnera en konstant referens till dataelementet.

pop Plockar bort och returnerar det översta elementet på stacken.

size Returnerar hur många dataelement det finns på stacken just nu. Det är lämpligt att denna kan användas även på en konstant stack.

Förberedelse

Sex sökande som som inte blev anställda undrar nu varför och du får den tråkiga uppgiften att förklara varför deras lösningar inte duger. Samtidigt får du uppgiften att ta fram en referenslösning som är korrekt för att visa på hur uppgiften skall lösas.

Du får själv hitta testfall (indata) för att testa att specifikationen är uppfylld. Försök hitta något testfall som inte kommer fungera.

Förberedelsen beräknas ta 2 - 4 timmar i anspråk. Du bör vara förberedd på att implementera en bättre lösning.

Uppgift på dojo

På programmeringsstugan skall du (informellt) förklara de allvarligaste brister du kommit fram till samt förklara varför det är fel (eller vad bästa praxis är). Fokusera speciellt på:

- Val av algoritm
- Val av datastruktur
- Korrekthet
- Om något saknas

Presentationen får ta den tid den tar, men cirka 5 minuter är kanske lämpligt för att diskutera och förklara de allvarligaste bristerna.

Samtliga lösningar är undermåliga på något sätt, en del mer än andra. Om du inte hitta så många brister får du gå in djupare i förklaringen av de brister du hittar. Behöver du "tips från coachen" får du själv se till att fråga någon. Alla varianter går att lösa bättre.

Publiken måste under "presentationen" vara observant på skillnader och likheter i deras tilldelade lösning, samt ställa frågor om varje lösning (assistent kommer närvara hela tiden om ni behöver input utanför gruppen).

När alla presenterat sin tilldelade lösning skall gruppen tillsammans (6 personer) först diskutera hur en så bra lösning som möjligt skall se ut, och sedan skriva en sådan tillsammans. Det går bra att utgå från en av de bristfälliga "lösningarna".