

Bedömningsmall för kontrollskrivning EDAA45 Programmering, grundkurs

2016-10-25, 14:00-19:00, Kårhusets Gasquesal

Hjälpmedel: Snabbreferenser för Scala och Java.

Instruktioner för kamraträttning

- Läs igenom hela denna bedömningsmall innan bedömningen påbörjas. Sätt dig in i mönsterlösningens angreppssätt och funktion, samt hur poängsättningen ska ske.
 - För varje del: läs noga lösningen du ska bedöma och sätt dig in i det specifika angreppssättet.
 - Ge maxpoäng om koden är korrekt, läsbar och lika bra som mönsterlösningen även om den är an-
norlunda. Det finns många olika sätt att lösa varje del, varav många är lika bra som mönsterlösningen.
En lösning kan vara lika bra som mönsterlösningen, även om den är längre eller kortare eller har ett
annat angreppssätt.
 - Ge noll poäng om lösningen saknas, är oläsbar, obegriplig eller helt fel.
 - Markering av poängavdrag (negativa) och uppgiftspoäng (inringad, positiv):
 - Markeringar ska göras med en penna med avvikande färg, gärna *röd*.
 - *Stryk under* de delar av en rad som är felaktiga eller bara delvis korrekta.
 - Om något saknas, *markera med en pil* var det som saknas borde finnas och skriv en kommentar
om vad som saknas.
 - Markera poängavdrag i *högerkanten* med *minuspoäng* enligt bedömningsriktlinjerna i detta
häfte.
 - När du markerat relevanta avdrag för en hel uppgift, beräkna och skriv total uppgiftspoäng *in-
ringad* i en cirkel, så att det går lätt att skilja uppgiftspoäng från poängavdrag. Uppgiftspoängen
ska inte vara negativ utan ligga i intervallet $[0, \text{Maxpoäng}]$.
 - Om det inte är uppenbart varför du ej givit maxpoäng, skriv då en kort förklaring i kanten.
 - Om en lösning är oläslig eller helt obegriplig, skriv "*oläsligt*" eller "*obegripligt*" i kanten.
 - Avsluta med att göra en *kvalitativ helhetsbedömning* av hela skrivningen och justera poäng för vissa
delar om du tycker att poängsumman för hela uppgiften inte överensstämmer med din helhetsbe-
dömning. Om du ändrar i din poängsättning, stryk över de gamla poängen med ett kryss och skriv
nya poäng bredvid.
 - Summera poängen och fyll i summan på omslagets framsida.
 - Ange era identitetsnummer i fälten för rättare på omslagets framsida.
 - Kamraträttningar med invändningar kommer att bedömas igen i efterhand av lärare. Även de kam-
ratbedömningar som godkänns utan invändningar kommer att stickprovsbedömas i efterhand av
lärare.
-

Del A. Totalt max 10 p

Efter init. av:	Vid kompile- ringsfel sätt kryss.	Vid exekve- ringsfel sätt kryss.	Ange statisk typ som kompilatorn härleder om ej kompilerings- eller körtidsfel.	Ange dynamiskt värde som tilldelas vid exekvering om ej kompilerings- eller kör- tidsfel.
u1			Point	Point(10.0,10.0)
u2		X		
u3			Movable	Point(100.0,100.0)
u4	X			
u5			Point	Point(130.0,120.0)

Rätta en rad i taget enligt nedan och sätt minuspoäng vid felaktigt svar i marginalen, t.ex. -1 eller -2 . Räkna ut totala poängen och resultatet inringat längst ner på sidan direkt i skrivningshäftet. Varje rad kan ge max 2p.

- Typerna och värdena ska skrivas som det står i tabellen ovan för full poäng. Dock får svaret skilja sig vad gäller i betydelselös användning av blanktecken och radbrytningar.
- Om rätt svar är ett kryss i kolumn 3 eller 4, men man kryssat för kompileringsfel när det ska vara exekveringsfel, eller tvärt om, eller om man kryssat i båda, ges -1 i avdrag.
- Om man svarat med fel typ men rätt värde ges -1 i avdrag.
- Om man svarat med rätt typ men fel värde ges -1 i avdrag.
- Om både värde och typ är fel ges -2 i avdrag.
- Om värdet är *nästan rätt*, så när som på enstaka detaljer i toString-representationen, till exempel 10 i stället för 10.0 på rad u1 ovan, ges -0.5 i avdrag.
- Om man svarat med typ/värde *och* kryss i kolumn 3 och/eller 4 på samma rad ges -2 i avdrag.

Del B. Totalt max 40 p

Allmänna bedömningsregler som gäller alla svar i del B:

- Olika formateringsvarianter som skiljer sig från mönsterlösning som ej påverkar funktionen, t.ex. extra (klammer)parenthespar eller radbrytningar, ger inga avdrag.
 - Felaktigt matchade (klammer)parenthespar, eller glömda (klammer)parenteser ger -1 i avdrag.
 - Dokumentationskommentarer och importsatser i specifikationer behöver ej upprepas i lösningen. Om ytterligare importsatser behövs men helt saknas ska -1 i avdrag ges, men smärre felaktigheter i själva sökvägen ger inga avdrag. Om en fullständig sökväg anges direkt på plats (i stället för import), ges inget avdrag om sökvägen har smärre felaktigheter.
 - Felaktigt avskriven metodsignatur från specifikationen ger -1 i avdrag.
 - Om nödvändigt semikolon saknas (t.ex. mellan flera satser på samma rad) ges -1 i avdrag.
 - Obetydliga fel i likhet med föregående två punkter där det framgår tydligt vad som egentligen avses ger -1 i avdrag.
 - Deklaration av **val** när det måste vara **var** för att koden ska fungera ger -2 i avdrag.
 - Deklaration av **var** när det skulle kunna vara en **val** och fungera lika bra ger -1 i avdrag.
 - Tillägg av element i samling på fel sätt, t.ex. $+$ i stället för $:+$ eller liknande, ger -1 i avdrag.
 - Enkla misstag som är lätta att åtgärda ger -1 eller -2 i avdrag, beroende på hur allvarligt felet är.
 - Allvarliga misstag eller stora ofullständigheter ger från -3 upp till $-maxpoäng$ i avdrag, beroende på hur mycket av koden som måste skrivas om för att det ska fungera.
 - Om samma typ av fel förekommer i samma uppgift mer än en gång görs bara avdrag en gång vid första förekomsten. Skriv "*samma fel igen*" i kanten för att indikera att avdraget redan är gjort.
 - Totala poängen för en uppgift kan inte bli negativ; om alla avdrag för en uppgift blir mer än maxpoängen ges 0 poäng.
-

Uppgift B1. Slumpmässiga identitetsnummer. 10p

```
1 object IdPrinter {  
2   import java.util.Random  
3  
4   def print(rnd: Random, n: Int, min: Int, max: Int): Unit = {  
5     var taken = Set.empty[Int]  
6     while (taken.size < n) {  
7       val r = rnd.nextInt(max - min + 1) + min  
8       if (!taken.contains(r)) {  
9         println(r)  
10        taken += r  
11      }  
12    }  
13  }  
14  
15 }
```

- Det räcker att svaret innehåller metoden print; det blir alltså inget avdrag om raderna 1, 2 och 15 i mönsterlösningen ovan saknas.
- Felaktig deklaration och/eller initialisering av mängd eller annan samling ger -1 i avdrag.
- Felaktigt villkor i while-satsen ger -1 i avdrag.
- Felaktigt villkor i if-satsen ger -1 i avdrag.
- Felaktig dragning av slumpstal ger -1 i avdrag om gränsen är fel endast med ± 1 , medan större fel ger -2 i avdrag.
- Glömd eller felaktig utskrift ger -2 i avdrag.
- Glömt eller felaktigt tillägg av slumpstal i samling ger -2 i avdrag.
- Slumptalen skrivs ej ut efterhand, utan på slutet i den ordning det råkar bli om man itererar över mängden (vilket gör att utskriften kommer att ske i hashkodsordning) ger -1 i avdrag.
- Det ger inga avdrag att använda en sekvenssamling i stället för en mängd, t.ex. en Vector, om man t.ex. anropar distinct eller toSet på lämpligt sätt så att funktionen blir som efterfrågats.

Uppgift B2. Roterar array på plats. 10p

```
1 object Rotate {
2   def rotateArrayOfStrings(xs: Array[String], fromPos: Int): Unit = {
3     if (fromPos >= 0 && fromPos < xs.length - 1) {
4       val temp = xs(fromPos)
5       for (i <- fromPos until xs.length - 1) xs(i) = xs(i + 1)
6       xs(xs.length - 1) = temp
7     }
8   }
9
10  def main(args: Array[String]): Unit = {
11    val argsErrorMsg = "Error: Not enough arguments."
12    if (args.length > 1) {
13      val numberOfSteps = args(0).toInt
14      for (i <- 1 to numberOfSteps) rotateArrayOfStrings(args, 1)
15      for (i <- 1 until args.length) println(args(i) + " ")
16    } else println(argsErrorMsg)
17  }
18 }
```

- Felaktig hantering av för få argument så att felutskrift ej blir rätt ger -2 i avdrag.
- Felaktig hantering av specialfall med tom array, eller array med ett element ger -2 i avdrag.
- Felaktig hantering av specialfall med fromPos negativ eller för stort ger -2 i avdrag.
- Enkla felaktigheter i översättning ger -1 i avdrag per fel.
- Lösningen behöver inte införa en strängvariabel för felmeddelandet som i mönsterlösningen.

Uppgift B3. Registrering av bonuspoäng. 20p

```

1 class BonusRegister {
2   import scala.collection.mutable
3   private val groupOfStudent: mutable.Map[String, String] = mutable.Map.empty
4   private val pointsOfGroup: mutable.Map[String, Vector[Int]] = mutable.Map.empty
5
6   def add(studentId: String, group: String, points: Int): Unit = {
7     groupOfStudent(studentId) = group
8     if (pointsOfGroup.isDefinedAt(group))
9       pointsOfGroup(group) = pointsOfGroup(group) :+ points
10    else
11      pointsOfGroup(group) = Vector(points)
12  }
13
14  def calculateBonus(): Map[String, Int] = {
15    val students = groupOfStudent.keySet
16    val nameAndBonusPairs = for (s <- students) yield {
17      val gr = groupOfStudent(s)
18      val cb = BonusRegister.collaborationBonus(pointsOfGroup(gr))
19      (s, cb)
20    }
21    nameAndBonusPairs.toMap
22  }
23 }
24
25 object BonusRegister {
26   def collaborationBonus(points: Seq[Int]): Int =
27     (points.sum / points.size.toDouble).round.toInt
28 }

```

- Om kompanjonsobjektet från specifikationen som finns på rad 25–28 i mönsterlösningen ovan inte är med i svaret ges inga avdrag.
- Om attributen på rad 3–4 är felaktigt avskrivna eller saknas ges totalt max –1 i avdrag.
- Metoden add omfattar 9 poäng av uppgiften och kan således max ge –9 i avdrag.
- Metoden calculateBonus omfattar 10 poäng av uppgiften och kan således max ge –10 i avdrag.
- Om metodhuvuden från specifikationen är fel avskrivna ges –1 i avdrag per felaktigt metodhuvud.
- Motsvarande funktionalitet som finns på raderna 7, 8, 9, 11 i metoden add motsvarar 2 poäng vardera och kan således ge –1 eller –2 i avdrag för respektive funktionalitet beroende felets allvarlighet. Om motsvarande funktionaliteten helt saknas ges –2 i avdrag per avsaknad.
- I metoden calculateBonus, beroende på felets allvarlighet:
 - Fel i eller avsaknad av funktionalitet som motsvarar raderna 15–16 ger upp till –3 i avdrag.
 - Fel i eller avsaknad av funktionalitet som motsvarar raderna 17–19 ger upp till –5 i avdrag.
 - Fel i eller avsaknad av funktionalitet som motsvarar raden 21 ger –1 i avdrag.