|  |  |
| --- | --- |
| Rad: | Funktion: |
| 8-13 | Testar equals |
| 14-19 | Testar clone |
| 21-25 | Testar toDouble genom att skapa ett antal nya RatNums på formen (j/i), subtraktera (double)(j/i) från dessa och kolla om absolutbeloppet av det erhållna värdet är ungefär lika med noll (felmarginal på 1.0e-13) |
| 28-29 | Kör testet RatNumTest2 |
| 30 | Skapar en ny BufferedReader, *in* refererar till denna |
| 31 | Skriver ut text |
| 32 | Påbörjar while-loopen |
| 33 | Skriver ut ”> ” i början av varje rad och skriver ut allt som är skrivet i System.out (tömmer bufferten genom *flush*) |
| 34 | Skapar en string *s*, som består av det som användaren har skrivit |
| 35-36 | Om användaren inte skrev något så försöker programmet igen. |
| 37 | Skapar en ny scanner sc med stringen *s* som parameter. Denna scanner är indelad i tokens, som är avdelade med mellanslag i stringen *s*. |
| 38 | Deklarerar en array *a* som består av strings och har 3 element |
| 39 | Deklarerar en int *i* |
| 40-41 | Lägger till tokens till *a* ifall de existerar. För varje element som läggs till blir i=i+1, där ’startvärdet’ för i är 0 och maxvärdet är 3. |
| 42 | Startar en if-loop om i>0 (alltså om åtminstone ett element har lagts till till arrayen *a.* |
| 43 | Skriver ut stringen *s*, en tabulator och en pil |
| 44-45 | Kollar att i=3 (alltså 3 tokens har lagts till till arrayen *a*) och att *sc* inte har en till token (alltså stringen *s* inte består av mer än 3 element). Om något av de två vilkoren inte är sanna skrivs ”Felaktigt uttryck!” ut. |
| 46-47 | Annars försök att: |
| 48 | Gör en ny RatNum *r1* av den första tokenen i arrayen *a*. |
| 49 | Gör en ny String *op* den andra tokenen i *a*. |
| 50 | Gör en ny char av det första tecknet i *op*. |
| 51 | Gör en ny RatNum *r2* av den tredje tokenen i arrayen *a*. |
| 52-53 | Om *op* inte har längden 1 (alltså inte består av bara ett tecken) eller *c* inte tillhör ”+-\*/=<” så skriver programmet ut ”Felaktig operator!”. |
| 54 | Annars: |
| 55 | Skapar en ny RatNum *res* = null. |
| 56-63 | Använder metoden add, sub, mul eller div på *r1* och *r2* och sparar resultatet i *res* (programmet använder den metoden som behövs beroende på *c*)*.* |
| 64-67 | Använder metoden equals eller lessthan och skriver ut resultatet (även här använder programmet den metoden som behövs beronde på *c*). |
| 68-72 | Om add sub mul eller div användes (alltså *c* är +, -, \* eller /) så skrivs ”Fel i add, sub, mul eller div” ut om *res*==null. Annars skrivs *res* ut. |
| 73-74 | Avslutar else från rad 54 och try från rad 46-47. |
| 75-77 | Om en NumberFormatException *e* gavs i try så skriver programmet ut ”NumberFormatException: ” + e.getMessage(). |
| 78 | Avslutar if loopen från rad 44. |
| 79-81 | Avslutar programmet. |