

TDP007 - 2017-02-22 - Opponering.  
Av Alexander Jonasson (alejo720)  
och Jonas Malmöf (jonma993)

### Uppgift1

"policy.rb":

- Vart är reglerna?
- Ska användaren kunna skriva med komma istället för punkt?
- Gillar att ni tillåter användaren att skriva med bindestreck istället för två punkter som är syntaxen för Ruby.
- Annars lättanvänt och tydlig

"uppgift1.rb":

- Ni har hårdkodat reglerna (rader 14-20). Det ska kunna ändras i "policy.rb".
- Bra att ni avrundar.
- Ni kan använda === operatoren för att jämföra en siffra med en Range.
- Annars snyggt med instance\_variable\_get och användning av method\_missing.
- Rad 28: Ni har väldigt lång rad för condition. Ni skulle kunna dela upp så det blir en condition per rad.

Testfall:

- Onödigt många testfall (funktioner). Varför inte "test\_points" där alla asserts som har med påängen att göra på samma ställe? Ha tydligare namn på testfallen.

### Uppgift 2

"rdparse.rb":

- Varför inte ha instansvariabel eller lokal istället för global. Speciellt om den är så generiskt namngiven.

```
239: "$container = Hash.new"
```

- Vi gillar eran flexibilitet med ":var". Denna rad kommer endast matchas om man skriver till exempel "(and true (hej))".

```
270: "rule :var do ; match('(' , :expr, ')') {[_ , a, _] a }"
```

- Gillar denna funktion. Den ger möjlighet för testfall. Man skulle kunna ha en check så den kan köras interaktivt också. Nu måste en sträng skickas in.

```
293: "@exprParser.parse str"
```

Testfall:

- Bra tester! Återigen tänk på namngivning och samla ihop liknande tester

### Utvecklingsblogg

Bra tankar, bra skrivet. Ni skulle kunna dela styckena i vilken uppgift det rör sig om.