

# Ada.O4: Undantagshantering

## Mål

Du skall efter denna uppgift kunna:

- Fånga undantag på korrekt ställe i koden.
- Resa undantag från korrekt ställe.
- Skapa egna undantag.
- Upptäcka radslut med *End\_Of\_Line*.

## Uppgift

I denna uppgift ska du komplettera samt utöka ett givet program som finns på filen *test\_exceptions.adb*. Filen innehåller ett huvudprogram som anropar olika underprogram som du ska skapa.

Programmet skall först felhantera inmatningen av ett heltal (del 1), sedan felhantera inmatningen av tre strängar med olika längder (del 2) samt felhantera inmatningen av fem korrekta datum (del 3). Innan uppgiften lämnas in ska även huvudprogrammet delas upp i underprogram (del 4). I det givna huvudprogrammet finns redan anropen till de underprogram du skall skapa. Din uppgift är att skapa själva underprogrammen samt lägga till undantagshantering där detta behövs.

**KRAV (detta gäller samtliga delar):** Alla huvud- och underprogram får ha som mest ca 20 rader kod mellan "begin" och "end". Tänk över uppdelningen av underprogram, vissa underprogram kanske kan anropas från flera ställen i koden.

### Del 1:

Skriv ett underprogram *Get\_Value\_Safe* som har tre heltalsparametrar *Min*, *Max* och *Value*. Underprogrammet skall låta användaren mata in *Value* via tangentbordet med felhantering enligt nedan.

*Value* ska få matas in tills denna är som minst *Min* och som mest *Max*. Programmet skall även felhantera *Value*-inmatningen så att användaren kan mata in felaktiga datatyper utan att varken hela programmet krashar eller underprogrammet avslutas. Istället ska underprogrammet låta användaren mata in *Value* igen tills att *Value* har ett korrekt värde. Huvudprogrammet skall till slut skriva ut det inmatade värdet.

Du kan anta att användaren alltid matar in korrekta värden på *Min* och *Max*, ingen felhantering av dessa krävs. Användaren matar alltså endast in heltal och *Min* är garanterat mindre än eller lika stort som *Max*.

## **Del 2:**

Skriv ett underprogram *Get* som har en strängparameter *Item*. Underprogrammet skall låta användaren mata in *Item* via tangentbordet med felhantering enligt nedan.

Underprogrammet skall låta användaren mata in en text som kan inledas med godtyckligt många blanktecken och radbrytningar. Underprogrammet skall, likt *Get* för heltal och flyttal, ignorera alla inledande blanktecken och radbrytningar för att sedan läsa in själva texten till *Item*.

OBS! Om den inmatade texten är kortare än vad *Item* är lång (vi kommer till radslut innan vi fyllt strängen) ska undantaget *Length\_Error* kastas/resas av underprogrammet.

Underprogrammet ska inte läsa in eller rensa bort det som ligger "efter" den inmatade texten i tangentbordsbufferten (texten kan innehålla blanktecken).

Om *Length\_Error* kastas/reses ska huvudprogrammet fånga detta undantag, skriva ut ett felmeddelande och sedan direkt avsluta programmet.

**Notera!** Den *Get* som beskrivs ovan har alltså vissa likheter med den *Get\_Line* som du redan använt i kursen, men med skillnaden att denna ignorerar inledande blanktecken och radbrytningar samt att den **inte** skall ta bort radslutstecknet efter själva texten. Du kan alltså inte använda *Get\_Line* i din lösning.

### **Del 3:**

I denna del ska du skapa:

1. en postdatatyp *Date\_Type* som består av tre heltal. Denna datatyp representerar ett datum angivet med år, månad och dag.
2. ett underprogram *Put* som har en *Date\_Type*-parameter *Item*. Underprogrammet skall skriva ut *Item* på formatet YYYY-MM-DD (förklaring av datumformatet hittas längre ner i uppgiften).
3. ett underprogram *Get* som har en *Date\_Type*-parameter *Item*. Underprogrammet skall låta användaren mata in ett datum enligt formatet YYYY-MM-DD (förklaring av datumformatet hittas längre ner i uppgiften). *Get* skall låta användaren mata in **ett** datum och felkontrollera detta enligt:
  - Om datumformatet inte följs ska undantaget *Format\_Error* resas/kastas. Detta gäller även om användaren matar in ett för kort datum (ex. 2001-2-03).
  - Om året är mindre än 1532 eller större än 9000 skall undantaget *Year\_Error* kastas/resas.
  - Om månaden är felaktig skall undantaget *Month\_Error* kastas/resas.
  - Om dagen är felaktig (utanför månadens antal dagar med hänsyn till skottår) skall undantaget *Day\_Error* kastas/resas. Längre ner finns en beskrivning av skottår.

Om datumet är korrekt ska detta datum "skickas tillbaka" via *Item*-parametern. Om datumet är inkorrekt på något sätt (se ovan) ska huvudprogrammet hantera det undantag som resas/kastas, skriva ut ett felmeddelande och sedan anropa *Get* igen tills att användaren matar in ett korrekt datum.

**KRAV:** *Get* ska anropa underprogrammet ifrån **del 2** för att läsa in datumet som en 10 tecken lång sträng. Du får **inte** modifiera underprogrammet från **del 2** för att lösa detta problem.

**Datumformatet** YYYY-MM-DD är exakt 10 tecken långt där YYYY är exakt fyra siffror som motsvarar år, MM är exakt två siffror som motsvarar månad och DD är exakt två siffror som motsvarar dag. YYYY och MM respektive MM och DD skall separeras med exakt ett bindestreck.<sup>1</sup>

*Info om månader och skottår:*

Månad:	Jan	Feb	Mar	Apr	Maj	Jun	Jul	Aug	Sep	Okt	Nov	Dec
Dagar:	31	28/29	31	30	31	30	31	31	30	31	30	31

Ett år är ett skottår om året är jämnt delbart på 4 men inte jämnt delbart med 100. Dock är år som är jämnt delbara med 400 alltid skottår. Om det är skottår så har februari 29 dagar istället för 28.

---

<sup>1</sup> En liten berättelse om kött- och grönsakshandlaren som beställde en skylt till affären (har alltså inte med uppgiften att göra, men kan vara lite kul att ha hört talas om). Han ville ha texten "Kött och grönsaker" på skylten och när skylten kom så var han lite missnöjd med hur långt det var mellan orden på skylten. Hans kommentar till dem som gjort skylten var: "Jag skulle vilja att ni har lite större avstånd mellan kött och och och och grönsaker". Skulle ni kunna lösa detta vore det tacknämligt.

Detta var anledningen till att vi inte skrev "YYYY och MM och MM och DD skall separeras med exakt ett bindestreck." ovan. :-)

#### **Del 4:**

I denna del ska du dela upp själva huvudprogrammet i underprogram. Uppdelningen ska vara logisk och uppfylla kravet om som mest ca 20 rader kod mellan "begin" och "end" per underprogram/huvudprogram. När du har delat upp programmet så det ser bra ut kan du skicka in detta för den slutgiltiga rättningen.

#### **Körexempel 1:**

```
Del 1:
Mata in Min och Max: 1 102
Mata in värde (1 - 102): sju
Fel datatyp. Mata in värde (1 - 102): !
Fel datatyp. Mata in värde (1 - 102): MassorMedText
Fel datatyp. Mata in värde (1 - 102): hejsan3
Fel datatyp. Mata in värde (1 - 102): 103
För stort värde. Mata in värde (1 - 102): 0
För litet värde. Mata in värde (1 - 102): 102
Du matade in heltalet 102.

Del 2:
Mata in en sträng på exakt 5 tecken: hello
Du matade in hello
Mata in en sträng på exakt 6 tecken:

HI !!!
Du matade in HI !!!
Mata in en sträng på exakt 10 tecken:


Pasta food
Du matade in Pasta food

Del 3:
Mata in datum 1: AAAA-BB-CC
Felaktigt format! Mata in datum 1: 123-45-67
Felaktigt format! Mata in datum 1:

2000-02-29
Mata in datum 2: 2100-02-29
Felaktig dag! Mata in datum 2: 2104-02-29
Mata in datum 3: 1998-13-11
Felaktig månad! Mata in datum 3: 1874-00-01
Felaktig månad! Mata in datum 3: 1999-03-32
Felaktig dag! Mata in datum 3: 1999-03-00
Felaktig dag! Mata in datum 3: 1999-03-31
Mata in datum 4: 1000-05-20
Felaktigt år! Mata in datum 4: 9001-05-23
Felaktigt år! Mata in datum 4: 1998-05-23
Mata in datum 5: 1977-04-31
Felaktig dag! Mata in datum 5: 1977-04-30
Datum nummer 1: 2000-02-29
Datum nummer 2: 2104-02-29
Datum nummer 3: 1999-03-31
Datum nummer 4: 1998-05-23
Datum nummer 5: 1977-04-30
```

**Körexempel 2 (Användarens inmatning är understruken. Detta för att visa på de blanktecken som är med i inmatningen):**

Del 1:  
Mata in Min och Max: **11 23**  
Mata in värde (11 - 23): **15**  
Du matade in heltalet 15.

Del 2:  
Mata in en sträng på exakt 5 tecken:

**plant**

Du matade in plant  
Mata in en sträng på exakt 6 tecken: \_\_\_\_\_  
\_\_\_\_\_ **A B CD**

Du matade in A B CD  
Mata in en sträng på exakt 10 tecken: **123456789**  
För få inmatade tecken!

**Körexempel 3:**

Del 1:  
Mata in Min och Max: **10 23**  
Mata in värde (10 - 23): **10**  
Du matade in heltalet 10.

Del 2:  
Mata in en sträng på exakt 5 tecken: **hej**  
För få inmatade tecken!

**Körexempel 4:**

Del 1:  
Mata in Min och Max: **2 300**  
Mata in värde (2 - 300): **234**  
Du matade in heltalet 234.

Del 2:  
Mata in en sträng på exakt 5 tecken: **1 2 3**  
Du matade in 1 2 3  
Mata in en sträng på exakt 6 tecken: **no**  
För få inmatade tecken!

**Körexempel 5:**

Del 1:  
Mata in Min och Max: **10 50**  
Mata in värde (10 - 50): **44**  
Du matade in heltalet 44.

Del 2:  
Mata in en sträng på exakt 5 tecken: **12345**  
Du matade in 12345  
Mata in en sträng på exakt 6 tecken: **234567**  
Du matade in 234567  
Mata in en sträng på exakt 10 tecken: **123456789**  
För få inmatade tecken!

### Körexempel 6:

Del 1:

Mata in Min och Max: **10 20**

Mata in värde (10 - 20): **10**

Du matade in heltalet 10.

Del 2:

Mata in en sträng på exakt 5 tecken: **hejsa**

Du matade in hejsa

Mata in en sträng på exakt 6 tecken: **hoppsa**

Du matade in hoppsa

Mata in en sträng på exakt 10 tecken: **1234567890**

Du matade in 1234567890

Del 3:

Mata in datum 1: **9999-01-011991-01-011992-03-04**

Felaktigt år! Mata in datum 1: Mata in datum 2: Mata in datum 3: **2022-04-05**

Mata in datum 4: **1875-33-011875-03-03**

Felaktig månad! Mata in datum 4: Mata in datum 5: **2000-02-29**

Datum nummer 1: 1991-01-01

Datum nummer 2: 1992-03-04

Datum nummer 3: 2022-04-05

Datum nummer 4: 1875-03-03

Datum nummer 5: 2000-02-29

### Körexempel 7:

Del 1:

Mata in Min och Max: **10 20**

Mata in värde (10 - 20): **13**

Du matade in heltalet 13.

Del 2:

Mata in en sträng på exakt 5 tecken: **hejsa**

Du matade in hejsa

Mata in en sträng på exakt 6 tecken: **hoppsa**

Du matade in hoppsa

Mata in en sträng på exakt 10 tecken: **1234567890**

Du matade in 1234567890

Del 3:

Mata in datum 1: **2#1#--1-01**

Felaktigt format! Mata in datum 1: **2022--1-01**

Felaktigt format! Mata in datum 1: **2022-01--1**

Felaktigt format! Mata in datum 1: **2022-01-01**

Mata in datum 2: **2022-4#-2#**

Felaktigt format! Mata in datum 2: **2021-03-04**

Mata in datum 3: **2022-12-31**

Mata in datum 4: **2023-01-01**

Mata in datum 5: **2000-04-04**

Datum nummer 1: 2022-01-01

Datum nummer 2: 2021-03-04

Datum nummer 3: 2022-12-31

Datum nummer 4: 2023-01-01

Datum nummer 5: 2000-04-04