

# TJa-biblioteket (studentversion)

---

---

De olika delarna i TJa-biblioteket som ni studenter kan använda är följande:

TJa.Calendar

TJa.File\_IO

TJa.Keyboard key\_codes

TJa.Keyboard.Keys

TJa.Lists

TJa.Lists.Unchecked.Double\_Linked.

General\_List.Checked\_Data

Sorted\_List.Checked\_Data

Unsorted\_List.Checked\_Data

TJa.Sockets

TJa.Window.Elementary

TJa.Window.Text

TJa.Window.Graphic

TJa.Misc

# Trevliga saker om strängar

---

Det finns en strängtyp som heter **Unbounded\_String**. Denna anpassar sig efter vad man lagrar i den. Kan vara trevligt att använda i vissa fall.

För att hantera detta i samband med de vanliga strängarna kan det vara bra att kolla in de rutiner som hör till denna datatyp. Kolla i paketet **Ada.Strings.Unbounded**.

Några bra saker (till att börja med) är:

```
S : String(1 .. 10);
US : Unbounded_String;

begin
  ...
  US := To_Unbounded_String(S);
  S := To_String(US);
  L := Length(US);
  ...
end ...;
```

Vill man hantera vanliga String lite mer kan man titta i paketet **Ada.Strings.Fixed**.

Några speciella saker som hör till stränghanteringen finns också i paketet **Ada.Strings**.

# Grafiska delen i TJa-biblioteket

---

---

```
with Ada.Text_IO;           use Ada.Text_IO;
with TJa.Window.Elementary; use ...
with TJa.Window.Text;      use ...
with TJa.Window.Graphic;   use ...
```

```
procedure Test_TJa is
```

```
    Msg : constant String := " Hello world ";
    X_UL : constant Integer := 10;
    Y_UL : constant Integer := 2;
```

```
begin
```

```
    Reset_Colours;  -- Standard (black & white)
    Clear_Window;
    Set_Graphical_Mode(On);
```

```
    -- Draw a rectangle on screen ...
```

```
    Goto_XY(X_UL, Y_UL);
    Put(Upper_Left_Corner);
    Put(Horizonta_Line, Times => Msg'Length);
    Put(Upper_Right_Corner);
```

```
    Goto_XY(X_UL, Y_UL + 1);
    Put(Vertical_Line);
    Goto_XY(X_UL + Msg'Length + 1, Y_UL + 1);
    Put(Vertical_Line);
```

```
    Goto_XY(X_UL, Y_UL + 2);
    Put(Lower_Left_Corner);
    Put(Horizonta_Line, Times => Msg'Length);
    Put(Lower_Right_Corner);
```

```
    Set_Graphical_Mode(Off);
```

```
-- Prints the message inside the rectangle ...
Goto_XY(X_UL + 1, Y_UL + 1);

Set_Background_Colour(Blue);
Set_Foreground_Colour(White);
Set_Bold_Mode(On);

Put(Msg);

Reset_Colours;
Reset_Text_Modes;  -- Resets bold mode ...

Goto_XY(1, Y_UL + 4);
end Test_TJa;
```

**OBS! Detta är inget äkta grafiskt paket utan en utökning av den vanliga texthanteringen i ett terminalfönster med t.ex. färger, några grafiska tecken m.m.**

**Varje grafiskt tecken motsvarar ett vanligt tecken, men det ser ut som ett vågrätt/lodrätt streck eller ett hörn.**

**Det man kan få fram är dock figurer som innehåller räta linjer horisontellt och vertikalt. Ett exempel skulle kunna vara att man kan rita ett vanligt husfönster.**

# Tangenthanteringsdelen i TJa-biblioteket

---

För att göra det lite roligare och användarvänligare i projekten är det meningen att ni skall undvika saker som att användaren matar in koordinater för att placera ut skepp, kryss m.m.

Användaren skall kunna köra ert projekt enkelt. T.ex. genom att använda sig av piltangenter för att flytta sig till en ruta man vill placera ut ett kryss på ett luffarschacksbräde eller vinkla någon kanon så att man kan skjuta i en viss vinkel.

För detta behövs något för att hantera tangentbordet lite smidigare. Till detta använder ni **TJa.Keyboard**-paketet.

```
with Ada.Text_IO;                use Ada.Text_IO;
with TJa.Window.Elementary;     use ...
with TJa.Keyboard;              use ...
```

```
procedure Move_Cursor is
  X, Y : Positive := 10;
  Key  : Key_Type;
begin
  loop
    Goto_XY(X, Y);
    Get_Immediate(Key);
    exit when Is_Esc(Key) or (X = 1);
    if Is_Left_Arrow(Key) then
      X := X - 1;
    end if;
  end loop;
end Move_Cursor;
```

# Innan ni kommer allt för långt i ert projekt

**VIKTIGT! Hur skall ni lagra era data i projektet? Vad har ni för olika möjligheter?**

**Enstaka variabler**

**Poster**

**Fält**

**Listor**

**Filer**

**Kombinationer av ovanstående**

**Hur skall ni strukturera ert program? Skall det finnas paket/underprogram/...? En tanke man bör ha är att man skall se om det finns saker som används både i server och klient. Isåfall är det praktiskt att lägga sådana saker i ett (eller flera) paket. Vad kan det vara?**

**Nätverksdelen (socketshanteringen)**

**Lagring/hantering av spelplan**

**Interaktion med användaren**

**Att dela upp projektet i olika delar gör att ni kan arbeta parallellt. T.ex. en person tar hand om ett paket, en annan tar hand om hur man spelar själva spelet och en ser till att användardialogen blir snygg och funktionell.**

**VIKTIGT! Planera så att ni vet hur ni skall anropa de olika sakerna i paketen och så att ni inte gör så att ni inte får ihop sakerna på slutet.**