PLPA COURSE WORK: DOCUMENTS WITH GRAPHICS AND TEXT

1. Description

A portable and scalable format for graphics with text is needed that can be stored and sent via the Internet. The format should support drawing lines, rectangles, circles and text. The most important use of the format will be to draw charts representing data, e.g., bar charts and pie charts. The graphics are drawn on a Euclidean plane with coordinates (x, y). A bounding box must be defined, inside which the drawing will be placed. Drawings outside a bounding box are clipped. Figure 2 shows the Euclidean plane with a bounding box. The following objects should be contained the format:

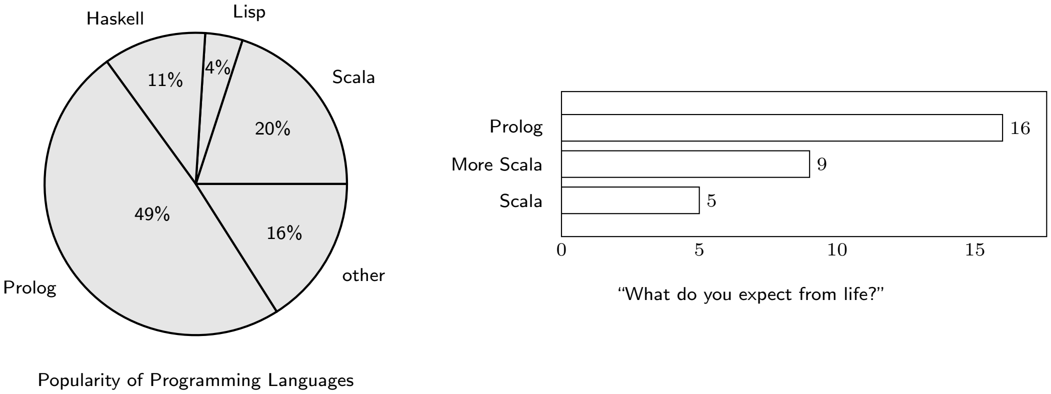


Figure 1. Examples of a pie chart and a bar chart

- `(LINE (x1 y1) (x2 y2))`: draw a line from `(x1 y1)` to `(x2 y2)`.
- `(RECTANGLE (x1 y1) (x2 y2))`: draw a rectangle with bottom left coordinate `(x1 y1)` and top right coordinate `(x2 y2)`.
- `(CIRCLE (x1 y1) r)`: draw a cicle with center `(x1 y1)` and radius `r`.
- `(TEXT-AT (x1 y1) t)`: draw the text `t` at `(x1 y1)`.
- `(BOUNDING-BOX (x1 y1) (x2 y2))`: the bounding box must be the first command.
- `(DRAW c g1 g2 g3 ... )`: draw the objects `g1`, `g2`, `g3`, ... in colour `c`, where `g1`, `g2`, `g3`, ... are any of the preceding commands like `(LINE (x1 y1) (x2 y2))`. (Outside a draw command the default colour black is used.)
- `(FILL c g)`: fill the object `g` with colour `c`, where `g` is any of the preceding commands.

Date: January 31, 2023.

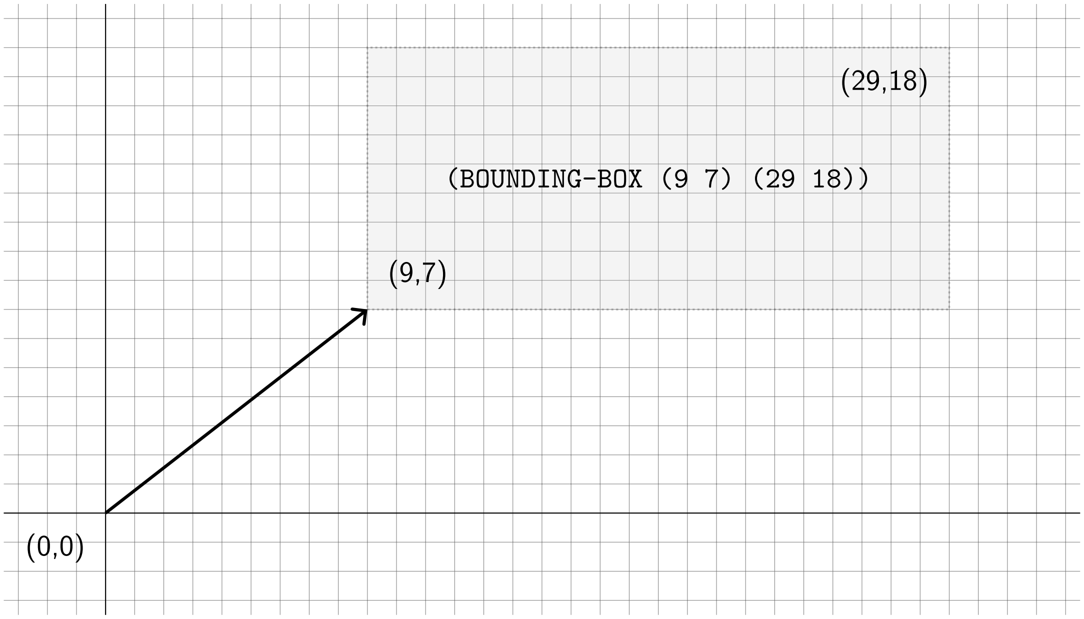


Figure 2. Drawing plane with bounding box

All coordinates are relative to the origin `(0,0)`. The objects must be drawn in the order specified. Lines are to be drawn with Bresenham's line algorithm. Circles are to be drawn with the Midpoint circle algorithm. For text, built in textual rendering is to be used.

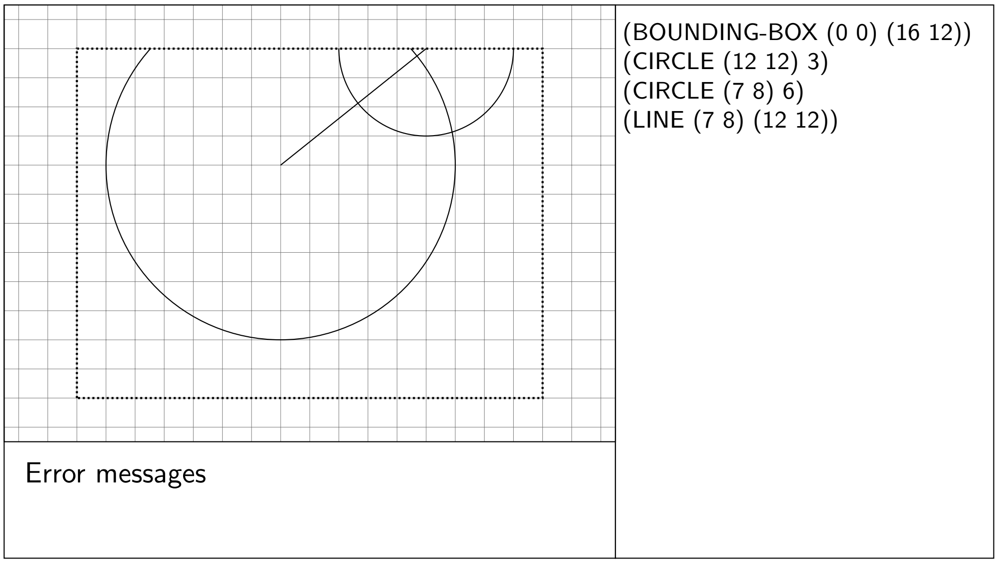
2. Instructions

Develop an IDE for creating chart diagrams. The GUI has to be implemented in Java while the graphic and the drawing have to be implemented in Scala. The GUI should have the shape indicated in Fig. 3. On the left-hand side the IDE should have the graphics viewer with grid displayed. On the right hand side a graphics language editor should permit to edit graphics images. When the graphics is drawn on the plane, the currently drawn object should be highlighted.

The Java part of the program must not contain the state of the drawing engine (bounding box, colours, ...). All of this must be contained in the Scala part. The program shown in the graphics language editor must be passed directly to the Scala interpreter and must not be modified in any way, e.g., by splitting it into separate commands.

3. Testing

Your application must be tested, both the Scala and Java portions. Apply testing and proof techniques discussed in the course. Explain your strategy and be prepared to show some of the tests during the demo. You must demonstrate your program by drawing one



```
(BOUNDING-BOX (0 0) (16 12))
(CIRCLE (12 12) 3)
(CIRCLE (7 8) 6)
(LINE (7 8) (12 12))
```

Figure 3. Sketch of the GUI

bar chart and one pie chart with text labels for the bars resp. the segments. The pie chart must have three bars. The pie chart must have 5 segments.

4. Hand-in

You have to hand-in the application and a small report (max. 6 pages) describing your application by 16. May.

5. Demonstation

On 16 May you will be expected to demonstrate your application in class. Attendance is obligatory for the assignment to be accepted.