Learning to Program with F#
Exercises
Department of Computer Science
University of Copenhagen

Jon Sporring, Martin Elsman, Torben Mogensen, Christina Lioma

October 21, 2022

0.1 Vec

0.1.1 Teacher's guide

Emne Moduler, namespaces og afprøvning

Sværhedsgrad Middel

0.1.2 Introduction

This assignment is about 2-dimensional vectors. A 2-dimensional vector (henceforth just called a vector) is a geometrical object consisting of a length and a direction. Typically, a vector is represented as a pair of numbers, $\vec{v} = (x, y)$, where its length and direction are found as,

$$\mathrm{len}(\vec{v}) = \sqrt{x^2 + y^2} \tag{1}$$

$$\mathrm{ang}(\vec{v}) = \mathrm{atan2}(y, x) \tag{2}$$

Vectors are often drawn as arrows with a head and a tail. In the Cartesian coordinate system, if the tail is placed at $(0, 0)$, then the head will be at (x, y). Addition of vectors is performed elementwise:

$$\vec{v}_1 = (x_1, y_1) \tag{3}$$

$$\vec{v}_2 = (x_2, y_2) \tag{4}$$

$$\vec{v}_1 + \vec{v}_2 = (x_1 + x_2, y_1 + y_2) \tag{5}$$

Addition can also be drawn, as shown in Figure 1.

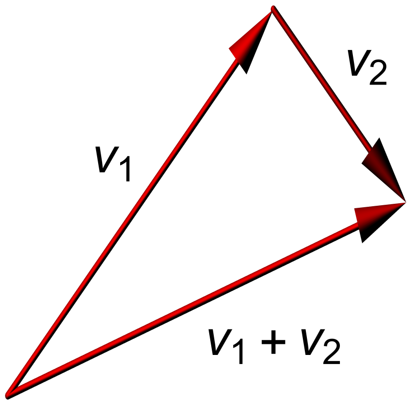

Figure 1: An illustation of vector addition.

0.1.3 Exercise(s)

0.1.3.1: Consider the signature file given in **??** which contains some of the standard operations for vectors.

Listing 1 vec2dsmall.fsi:
A signature file for vector operations.

```
/// A 2 dimensional vector library.
/// Vectors are represented as pairs of floats
module vec2d
/// The length of a vector
val len : float * float -> float
/// The angle of a vector
val ang : float * float -> float
/// Addition of two vectors
val add : float * float -> float * float -> float * float
```

Solve the following sub-tasks:

a) Extend the signature file with documentation using the documentation standard.

b) Write a library `vec2dsmall.fs` implementing the signatures.

c) Compile the signature and the implementation into `vec2dsmall.dll` demonstrating that there are no syntax errors.

0.1.3.2: Write a White-box test of the library.

0.1.3.3: Points on a circle of radius 1 can be calculated as $(\cos\theta, \sin\theta)$, $\theta \in [0, 2\pi)$. Consider the closed polygon consisting of $n > 1$ points on a circle, where $\theta_i = \frac{2\pi i}{n}$, $i = 0..(n-1)$, and where neighbouring points are conected with straight lines.

Write a program with the function,

```
polyLen : n:int -> float
```

which uses the above library to calculate the length of the polygon. The length is calculated as the sum of line pieces connecting neighbouring points. The program should further write a table of lengths for increasing number of points n, and the results should be compared with the circumference of a circle with radius 1. What appears to be the limit, when $n \to \infty$?

0.1.3.4: The library `vec2d` is based on the representations of vectors as pairs (2-tuples). Make a sketch of a signaturfil for a variant of the library, which avoids tuples. Discuss possible challenges and major changes, which the variant will require both for the implementation of the library and the application program.

0.1.3.5: Consider the following application

Listing 2: Simple usage of the Color library.

```
let v = (1.3, -2.5)
printfn "Vector %A: (%f, %f)" v (vec2d.len v) (vec2d.ang v)
let w = (-0.1, 0.5)
printfn "Vector %A: (%f, %f)" w (vec2d.len w) (vec2d.ang w)
let s = vec2d.add v w
printfn "Vector %A: (%f, %f)" s (vec2d.len s) (vec2d.ang s)

```

First run the code with `fsharpc`. Then perform a tracing by hand of the above code and the implementation of your library. Did you discover any errors? Do you get the same output?

0.1.3.6: Write a White-box test of the library.

0.1.3.7: Write a Black-box test of the library.