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 Owls and mice

0.1.1 Teacher's guide

Emne classes and objects

Sværhedsgrad Hard

0.1.2 Introduction

This assignment is about simulating a preditor-prey relationship in a simplified setting.

0.1.3 Exercise(s)

0.1.3.1: You are to simulate owls and mice in a closed environment. The owls are immortal and hunt mice to eat, and the mice run around randomly and multiply (propagate). The overall rules for the simulation are:

(a) The environment must consist of $n \times n$ fields organized as a checkerboard.

(b) Alive animals have a coordinate in the environment, and there can only be one animal per coordinate.

(c) The simulation updates in ticks, and after each tick, all animals perform an action.

(d) The simulation runs for T ticks.

(e) There must initially be O owls and M mice.

The possible actions are:

(f) A mouse can move to a neighbouring empty field.

(g) A mouse must have a counter, such that after p ticks, the mouse will not move but multiply. The effect is that an offspring is created in an empty neighbouring field. If there is no empty neighbouring field, then the mouse waits a turn.

(h) An owl can move to all neighbouring fields not occopied by another owl. If an owl moves to a field with a mouse, then the mouse is eaten and the mouse is removed from the board.

You are to:

(a) Use the object-oriented programming paradigm and include inheritance in your solution.

(b) Create a program `simulate.fsx`, which runs the simulation and prints the tick number and the total number of mice after each tick to the textfile `simulation.txt`. The program must accept the parameters n, T, p, M, O, at the command-line when the simulation starts.

(c) Collect the main classes in an implementation file called `preditorPrey.fs`, which `simulate.fsx` links to.

(d) Make a white-box test of the implementation file, `testPreditorPrey.fsx`.

(e) Find parameters, where the mice population diminishes to zero quickly, explodes, and is seemingly in balance, and demonstrate this by copying and renaming the textfile `simulation.txt` to the three corresponding files

- `simulationExtinction.txt`,
- `simulationOverpopulation.txt`, and
- `simulationBalance.txt` respectively.

You are also to write a report:

(f) The report must as a minimum include the sections: Introduction, Problem analysis and design, Program description, Testing, Experiments, and Conclusion. Include a User guide and your source code as appendices.

(g) The report must be no longer than 10 pages excluding the appendices.