Exercise 1

This exercise is meant to provide a first look at the `Finmath` library implementation for the simulation of a LIBOR Market Model $L_i := L(T_i, T_{i+1})$, $0 \leq i \leq n-1$, with

$$dL_i(t) = L_i(t)\sigma_i(t)dW_i(t), \quad 0 \leq t \leq T_i, \quad i = 0, \dots, n-1, \tag{1}$$

where $d\langle W_i, W_j\rangle(t) = \rho_{i,j}(t)dt$. You can find an already implemented class

`com.andreamazzon.exercise9.LIBORMarketModelConstruction`,

with a method `createLIBORMarketModel` which returns an object of type

`net.finmath.montecarlo.interestrate.LIBORModelMonteCarloSimulationModel`.

An object of a class implementing this latter interface is obtained by linking together an object of type `IndependentIncrements` (for example Brownian motion) with one of type `net.finmath.montecarlo.interestrate.LIBORMarketModel`. This can be done by using the constructor of the class

`net.finmath.montecarlo.process.EulerSchemeFromProcessModel`.

As you can see, the method `createLIBORMarketModel` is mainly devoted to construct an object of type `LIBORMarketModel`. Have a look at the code and at the `Finmath` library classes which it involves, in order to get what is needed to implement the LIBOR market model. To have a further help understanding it, have a look also at the PDF file *Libor market model simulation* you find in `com.andreamazzon.exercise9`. Note that in our case, the term $\sigma_i(t)$ in (1) is given by a volatility structure

$$\sigma_i(t) := (a + b(T_i - t))\exp\left(-c(T_i - t)\right) + d, \quad t \geq 0, \quad i = 0, \dots, n-1,$$

$a, d \in \mathbb{R}$, $b, c > 0$. Moreover, we define a correlation

$$\rho_{i,j}(t) := \exp\left(-\alpha|T_i - T_j|\right), \quad t \geq 0, \quad i, j = 0, \dots, n-1,$$

$\alpha > 0$. Do then the following:

- Taking inspiration for example from `net.finmath.montecarlo.interestrate.products.Caplet`, write a class `myDigitalCaplet` implementing

  `net.finmath.montecarlo.interestrate.products.AbstractLIBORMonteCarloProduct`.

  The method `getValue`, taking as inputs the evaluation time and an object of type `LIBORModelMonteCarloSimulationModel`, must in this case return the discounted payoff of a digital caplet with underlying $L(T_i, T_{i+1})$. The dates T_i and T_{i+1}, or one of those and the period length, must be given in the constructor of the class.

- Complete where needed (look at the capitalized instructions) the implementation of the class `LMMDigitalCapletTest`, that you find in `com.andreamazzon.handout9`, under tests.

  Here we use our `createLIBORMarketModel` method in order to construct and simulate a LIBOR Market Model with tenure structure

  $$T_0 = 0 < T_i = 0.5 < T_{i+1} = 1 < \cdots < T_{20} = 10,$$

  correlation decay parameter $\alpha = 0.5$, volatility parameters $a = 0.2$, $b = 0.1$, $c = 0.15$, $d = 0.3$, and initial forwards $L_i = 0.05$ (note that, thanks to the method `ForwardCurveInterpolation.createForwardCurveFromForwards`, we don't have to provide all the initial forwards as the missing ones are interpolated).
  In particular, for every T_i, we consider the digital caplet with underlying L_i, notional $N = 10000$, strike $K = 0.05$ and maturity T_i, and compare its Monte Carlo price to the analytical one. Here your duty is to compute the Monte Carlo prices.

Exercise 2

Note: This exercise has nothing to do with the current lecture content, so it is completely optional to do it or not. However, it can be good training to test your skills with object-oriented programming.

Write a program to randomly match sixteen football teams, in order to form eight games. You must comply with the following rules:

- The teams are divided in two big groups, the *top seeded* and the *second seeded*. Every first seeded team has to play against (only) one second seeded team.
- Teams of the same nationality cannot play against each other.
- Every first seeded team has already played in a former stage against a second seeded team (i.e., *they have already been matched*): these teams cannot play against each other.

Example: team 1 is first seeded. So it can play only with one of the second seeded teams. However, it cannot play against teams B and C because they are all Spanish, and against team E because they have already played against each other in the first stage.

Remark: since you don't want to have to make a second draw (this would be a shame, right? :)) you have to avoid to end up in situations where you have no possible choice. This is the most difficult part of the program. For example, you must avoid situations where:

- At least one team cannot play against any other team: say for example four teams are left, two first seeded and two second seeded. Team 1 is first seeded, and is an italian team. Both the second seeded teams are Italian, or one is Italian and the other has already played against team 1 in the first stage.
- At least two teams, both first seeded or both second seeded, have only one possible choice, and is the same. For example, teams 1 and 2 are first seeded, and because of the rules above can both play only against team A.
- .. And so on, similar situations.

How to test your program: you can try to consider 16 teams with nationalities of your choice, taking into consideration that every nation cannot have more than 4 (in extreme cases 5) teams. Try to find some scenarios that can challenge your program (like having 2 first seeded teams from Spain, 2 from England and 2 second seeded teams from Spain, 2 from England, etc). Execute the draw many times, maybe printing the nations of the teams that are matched, printing some error message if it is not possible to find a match for at least one team, etc.

If you are quite into football, you probably know that such a program failed when drawing the Champions League last-sixteen last week, so if yours succeed, you can say you have done better than the company working for UEFA!

If instead you really don't want to think in terms football, you can imagine having to match eight women with eight men, with the rules above: every woman already knows one and only one man and you do not want to match them; additionally, you don't want to match women and men of the same nationality (this is some kind of language exchange maybe?).