



Rules:

- **You are not allowed to use books, notes, or other material.**
 - **You can answer in Italian or English.**
 - **Total time for the test: 1.5 hour.**
-

1. Implement a `DoubleBuffer` class in Java. The class encapsulates two arrays of integers: `Alpha` and `Beta` (size determined at class instantiation time) and offers four methods: `addAlpha` and `addBeta` to add data (they suspend the caller if the arrays are full), `clear`, to clear both arrays, and `compute` that applies a complex and time consuming computation that involves the elements of both arrays `alpha` and `beta` (imagine a static method `void Foo.calc(int[], int[])` exists that performs such computation and writes the result on screen). Put your synchronization code to maximize parallelism in case the four methods would be called by different threads. Notice that it is the caller of the `compute` method which executes the complex computation, not an additional, ad-hoc thread.
2. Describe the main differences between RMI and Corba as two middleware systems offering a RPC-like form of communication. solution at RMI vs CORBA
3. Suppose you have to implement a videogame organized in rounds (like a table game, e.g., monopoly). Imagine the different participants have to share some state (e.g, who owns what, where is each participant on the table, etc.). Which middleware would you prefer among those described during the course? Why? Provide also an architectural sketch of the solution (use UML, fragments of code and any other notation you think may help your description).
4. Write in Erlang a process (i.e., a function to be spawned) that receives some data, spawns a new process to compute it (the new process executes the pre-defined function `f○○`), and receives back the result (an integer), which it accumulates (i.e., sums) and prints every 10 results received.