1. Result has the value “false “.

2. The header would look like this: `public class K<T>  {..}`

    This syntax is called “Generics”.

3. The name of the pattern is Singleton.

4. Decimals to binary: 12 is 001100 & 55 is 110111.

    AND: 000100 (binary), 4 (decimal)

    OR: 111111 (binary), 63 (decimal)

    XOR: 111011 (binary), 55 (decimal)

5. The problem with the code is that it leads to an infinite loop.

    Adding items to the list increases list count that is evaluated every iteration (list.Count).

6. I never heard of deadlocks before, but my intuition (and some google knowledge) tells me, that if I say “Yes” I’d decide for heaven and all 3 programmers go to heaven, if I say “No” no one goes to heaven, but I’d make an assumption, because the other programmers have not given their information. If no decision is made, they once again get stuck in a deadlock. To avoid getting stuck in a deadlock & to avoid making assumptions (and simply deciding for yes or no), I’d say “I don’t know either, but I’d like to go to heaven (or don’t in case I don’t, but I assume I know my own state) & if we make another circle where you share your own sentiment/state, the programmer who heard all other answers can give Petrus the desired answer.” Basically, making the information accessible to all. As long the information is not shared no answer can be given without assumption. But assumption is still better than deadlock, so if no information can be obtained at all after a full circle, making an assumption (based on the own sentiment so at least the wish of one agent is fulfilled) would solve the issue, too.