EXERCISE 7

Before running the program, study the code of the program **Exercise07_Cooperation** and answer the following questions:

- How do we know that a thread class has been declared?
- How many objects of the thread class have been created?
- Which sentence makes a thread to run?
- What do the threads do?
- Which is the purpose of the method **join**?
- Why does the method join be inside a **try {} catch** section?

Now you can run the program. Does the program run as expected? Why?

EXERCISE 8

Before running the program, study the code of the program **Exercise08_Competition** and answer the following questions:

- What is the purpose of the class **Increase.java**?
- What is the purpose of the class **Decrease.java**?
- What is the purpose of the attribute **value** of the class **Counter.java**?
- What do you think is going to happen when the program runs?

Now you can run the program. Does the program run as expected? Why?

EXERCISE 9

Before running the program, study the code of the program **Exercise09_ConsumerProducer** and answer the following questions:

- What is the purpose of the class **Consumer.java**?
- What is the purpose of the class **Producer.java**?
- What is the purpose of the class **Vault.java**?
- What do you think is going to happen when the program runs?

Now you can run the program. Does the program run as expected? Why?

EXERCISE 10

Before running the program, study the code of the program **Exercise10_TravelAgency.**

- What do you think is going to happen when the program runs?

Now you can run the program. Does the program run as expected? Why? What would you change?

EXERCISE 11

Sarah and John have a shared account in a Bank with a total ammount of 500 €. Sarah is in Valencia and has decided to get 400 €. At the same time, John is in Paterna and is trying to get 400 €. For the purposes of this scenario the cash machine only allows to get 100 € per transaction.

Every time a user gets cash from the cash machine will happen the following sequence of steps:

- The cash machine shows a message like this one:
  Kyle: You have asked for a withdrawal of 25 €

- The cash machine takes 2 seconds to prepare the money

- The cash machine informs the user with a message like this one:
  Kyle: The withdrawal of 25 € has been accepted
  Kyle: Tha balance in your account is 1275 €

If there is not enough money a message will be displayed to the user.

Kyle: Your current balance does not allow you to get 75 €

Design a program in Java to solve this scenario.