***Barracks***

I used classes for Archer, Gold and Treasure instead of record in order to be able to change the properties since the record properties are immutable.

**Archer** class with properties:

* ID auto generated
* Available as Boolean which is true by default and when the archer is sent for plundering the it turns false so that archer can’t be sent again
* Dead as Boolean it’s false by default so when the archer dies (when applying the possibility) when we apply the possibility for archer to die 50% or 25%
* Upgraded as Boolean it’s false by default if barracks did not upgrade archers yet (we check a Boolean variable called archersUpgraded in the main app) and it’s going to be true by default if that archersUpgraded variable is true.

I create constructor to give values to all properties when I create a new instance for Archer, in addition methods to get the properties values for checking and set properties for modifying.

**Gold** class with properties:

goldCoin, bagOfGold, treasureChest and TOTAL as integer type, then when creating an instance for Gold it fires switch to add randomly 5, 25 or 50 values respectively to the properties and to add the whatever value to TOTAL.

**Treasure** class with properties:

Monuments as integer type so when an archer plunder (which means creating an instance for Treasure) the constructor adds 1 to monuments or creates an instance of Gold randomly with 25% and 75% respectively.

**app** class has **buildArcher** method which requires food from towncenter, the program gets food as a RESTful request by which return Boolean if the amount of food requested is available.

I apply the same logic for getting wood from workshop and getting workers from towncenter.

In addition if we got food and there is no enough wood then we have to return wood by a separated method.

**App** class has **plunder** method which will be fired automatically every (5 to 8) seconds randomly.

First the method checks if there is available archers then the for loop breaks so plunder method doesn’t send more than one archer.

When the archer sent it will be plundering treasure in an infinite thread every (5 to 8) seconds.

The thread will check if the archer is upgraded to apply the die possibility and if archer dies then toggle the archer isDead property to true and remove it from the array and inform twoncenter.

If the archer isDead property is still false then the archer will stay in the infinite loop to plunder treasure again.

**App** class has **checkWin** method which will be fired automatically every (5 to 8) seconds since it is inside an infante loop in the main method.

This method will check if monuments >= 3 and if the twoncenter is upgraded.

To check if the twoncenter is upgraded it is a RESTful API get request from the twoncenter as a Boolean