Practice 08-09
Problem solving

1. Tickix.

A new e-ticket seller has emerged and is in desperate need of an interactive website where people can look for concerts and buy tickets. Since web design is not in our field, we can build the core of their system. A must have functionality is registering new accounts, creating new events and selling tickets.

- An **account** on the website is defined by a *unique nickname* (no passwords at this stage of the system). Each account should have an e-wallet, for simplicity we'll implement it as a number representing the account's funds. There must be a way to add and withdraw funds to and from the account. Of course there should be a list of all the tickets bought by the account, as well as the count for each ticket (we can buy more than one ticket for the same event).
- An **event** is defined by its *unique name*, the date of the event. the number of people that can attend it and its price. Provide a way to change all its "properties".
- Each **ticket** must have a datetime of when it's been bought, price and it should somehow be linked to the event for which it's been bought.
- The **system** should be able to sell tickets to accounts, create new events, create new accounts, display all events, display events in a certain order - price (ascending or descending), most recent and least recent.
- Create a simple console dialog input for the system.
- After closing the application all of the information about the system should be saved such that when the application is opened once again the system must load its previous data.