

'ITrainController' is the interface (contract) containing the actions for a train journey. In this case only method 'void NextStation()'.

'ITrainJourney' is the interface (contract) containing the methods of a train journey. Since the train journey is the model, this interface has at least the methods AddObserver() and RemoveObserver(). These 2 methods receive a ITrainDisplay object as parameter.

'ITrainDisplay' is the interface (contract) for all displays showing information of a train journey. Method 'Update(Station station)' has to be there, that will be called from within the model.

'TrainController' is the class that implements interface 'ITrainController'. So, method 'void NextStation' has to be present. The constructor of TrainController receives the train journey (ITrainJourney) as parameter. Method 'NextStation' delegates the action to: journey.NextStation();

'TrainJourney' is the class that implements interface 'ITrainJourney'. TrainJourney is the model, containing a list of Station objects, and a list of observers: private List<ITrainDisplay> observers; if the model changes (next station) then all observers are notified with: Update(currentStation);

'TrainDisplay' is a Windows form showing the current state (station) of the train journey. This is an observer, and therefor this form/class needs to implement interface 'ITrainDisplay'.

'ControlPanel' is a Windows form with a button to indicate that the next station has been arrived. The eventhandler of the button 'Next Station' will call: controller.NextStation();

