Description of application

Abstract factory pattern

Dimitar Markov – ei4s1

***Introduction***

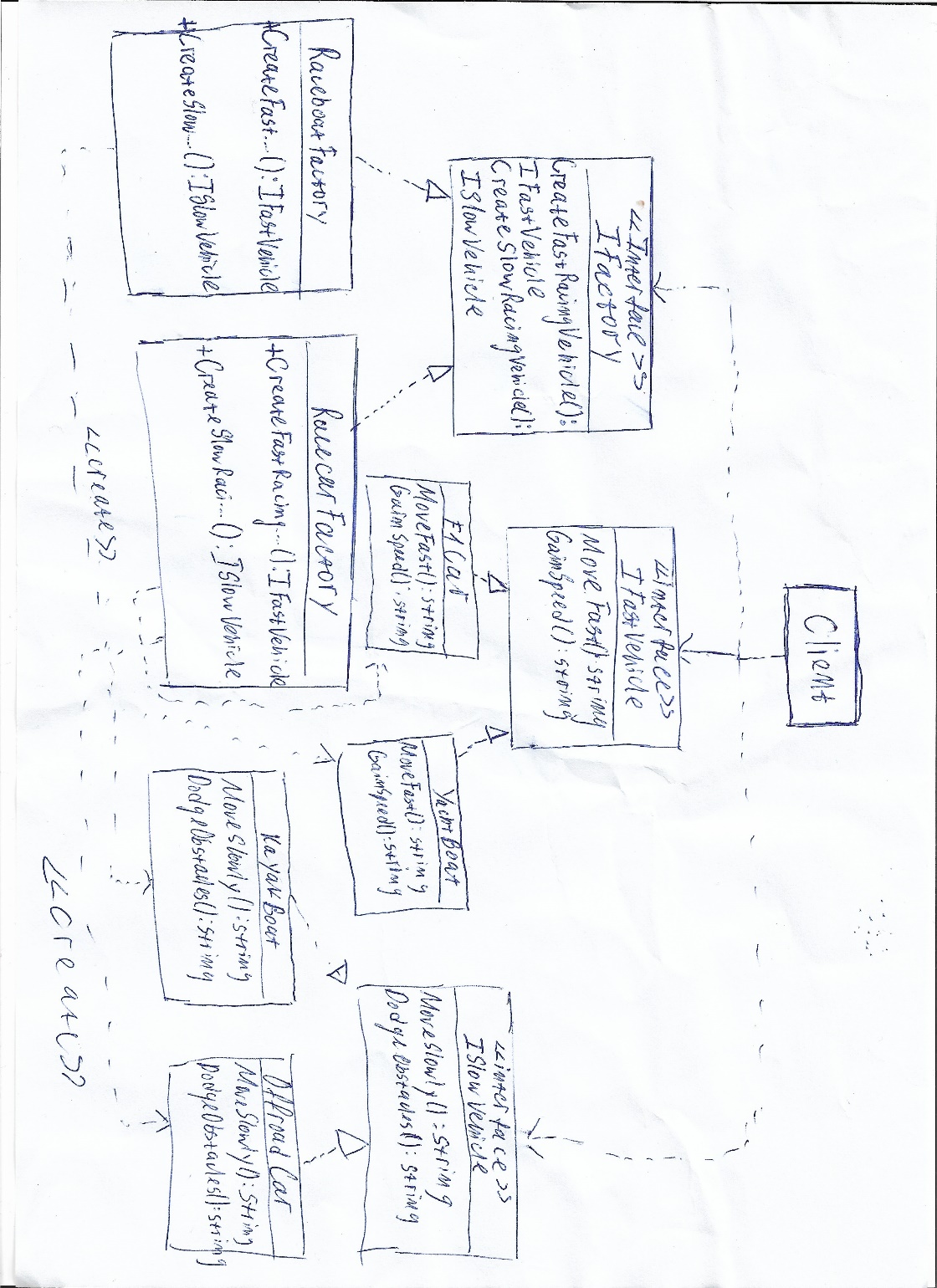
The purpose of this application is to make use of the “abstract factory pattern”, to do so I’ve develop an application or game related to racing. It is very simple and consists of 2 buttons: “Start race with boats” and “Start race with cars”, based on which one you press the program creates a different game and logs different messages.

***Organization and architecture***

The project is divided in 5 class libraries

* Client
* RacingGameFactory
* FastVehicles
* SlowVehicles
* Tests

If we explore these divisions further, we will get the following class diagram



We can also relate the classes to the abstract factory pattern like that:

* IFactory – Abstract factory
* IRaceboatFactory – Concrete factory
* IRacecarFactory – Concrete factory
* IFastVehicle – Product
* ISlowVehicle – Product
* F1Car - Concrete product
* YachBoat – Concrete product
* KayakBoat – Concrete product
* OffroadCar – Concrete product

The implementation is very basic and specific all the “concrete products” have 2 methods which return a string with a message. It is strongly related to abstract factory pattern. This enables us to have big flexibility, encapsulation and extensibility of the application