[](http://academy.telerik.com)

Доклад за отборна работа

1. Отбор: "Leonardo"

2. Участници:

2.1 Ангела Тенева - ellapt - 1000563

2.2 Валентин Пелтеков – valiyo - 9003252

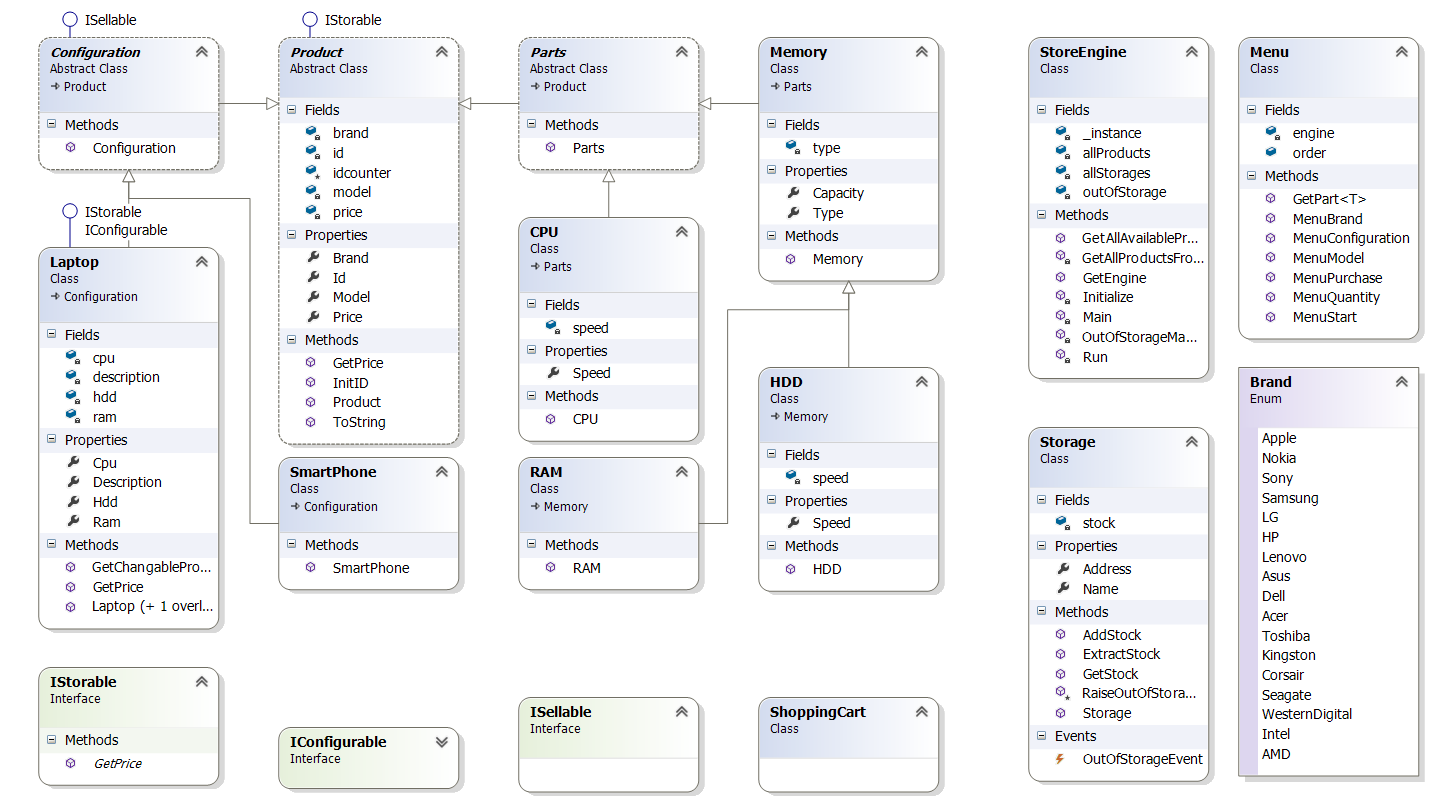
2.3 Ирена Стоянова – inlovewithlife - 7004076

2.4 Симеон Симеонов - simo.smo - 9005314

3. URL на проекта:

https://telerikleonardo.visualstudio.com/DefaultCollection/tkleonardo/

4. Диаграма на класовете:



5. Описание

Целта на проекта е изготвяне на приложение за управление продажбите на магазин за IT компоненти. Основно системата поддържа два типа продукти – конфигурирани от клиента или напълно готови. За улеснение ние сме избрали да имаме два типа крайни продукта – лаптопи и смартфони. Имаме складове за части и крайни продукти. Нашата система е направена да поддържа и управлява няколко склада, като това е видимо само за програмистите, но не и за клиентите. Това, което потребителят вижда е дали исканият от него продукт го има като наличен или не.

Логиката на програмата е реализирана в един основен клас – StoreEngine. Той изпълнява логиката на приложението като зарежда информацията в базата данни и пуска менюто, от което клиентът има възможност да избере и конфигурира продукта, който иска да закупи. При реализация сме използвали design pattern – Singleton, затова и самият клас позволява само една инстанция, само един обект. Това се подсигурява чрез метода GetEngine(), който ако имаме вече създадена инстанция от класа я връща като резултат, а в противен случай я създава и връща като резултат.

Зареждането на базата данни се извършва от метода Initialize. Той зарежда данните за типовете продукти и части както и информация за различните складове и какво налично има в тях.

Класът StoreEngine има и втори метод, който се казва Run и зарежда меню за клиента, от което той избира желания продукт. При завършване на заявката за продукт, той заедно с допълнителните му части се отписва от складовата наличност. Системата автоматично търси необходимата наличност в първия склад. Ако бройката заявени продукти е по-голяма от наличното в първия склад автоматично оставащото количество се отписва от следващия склад и т.н. В случай, че броят поръчани артикули е повече от наличното в складовете, програмата извежда съобщение за грешка. Това съобщение за грешка се реализира чрез събитието "OutOfStorageEvent". Към него е прикачен метод , който следи за това събитие и при извикването си преглежда всички останали складове за наличност от дадения продукт. Ако няма наличност от продукта в нито един склад се извежда съобщение за грешка и в подходящ списък се запазва ID-то на този артикул. Този списък съдържа ID-тата на всички артикули, които не са налични в нашите складове. При следващa поръчка, системата не визуализира артикулите, от които нямаме в наличност.

Програмата има интерфейс IStorable, който изисква от класовете, които го имплементират да имат метод GetPrice. Този метод е деклариран на ниво Product, като просто извежда стойността на пропъртито Price. За класа лаптоп метода е override-нат като извежда не само стойността на самия лаптоп, но и към нея добавя стойността на допълнителните части, които са конфигурирани за него (РАМ, Процесор, Харддиск и т.н.).

Менюто има пет нива. В началото потребителя избира дали иска лаптоп или смартфон. След това във второто меню се показват само марките, от които имаме в наличност от избрания тип продукти. Следващата стъпка е избор на модел. Ако е избран смартфон се извежда менюто за избор на желаната бройка за покупка. Освен това се извежда името на избрания продукт и неговата цена.

Ако изборът е лаптоп има опция да се добави твърд диск, процесор или RAM памет. Отново текущо се извежда цената на конфигурирания лаптоп. След това се избира какво количество да се закупи. Поисква се потвърждение от клиента за покупката и се изписва цената на заявката.

Структурата на класовете с продукти е следната – имаме родителски клас Product, на който всички останали класове са наследници. Под този клас имаме 2 подкласа – Parts и Configuration. Под класа Parts се намира дървото с различните части, които могат да бъдат добавени в готовите конфигурации. Под класа Configuration се намира дървото с различните готови конфигурации, които могат да бъдат предлагани за продажба. Всички обекти в дървото имат няколко общи полета – ID, price, brand, model. Идеята на полето ID e да е уникално за всички продукти и конфигурации в нашата система. За това се грижи методът InitID, който се извиква в конструкторите на обектите и им добавя уникален идентификатор. За да подсигурим, че този идентификатор ще е уникален използваме едно статично поле на класа Product, в което държим поредния номер на последния добавен продукт.