**Магазин –**

**Проект по Обектно-Ориентирано програмиране**

Изготвила: Марина Дудушка, 45488

**Съдържание**

1. Увод...........................................................................................................2
2. Реализация................................................................................................2
3. Изпълнение на поставените задачи........................................................2
4. Основни концепции и проблеми............................................................3
5. Обща архитектура....................................................................................5
6. Реализация и тестване.............................................................................7
7. Примери....................................................................................................9
8. Заключение...............................................................................................9
9. **Увод**

Проектът представлява система за управление на търговски обект, която може да се използва както от големи, така и от малки вериги магазини и супермаркети. Системата поддържа различни видове продукти и възможност за добавянето на нови такива. Също дава възможност на потребители да се регистрират и да пазаруват *онлайн*.

1. **Реализация**

Реализиране на система на магазин, притежаваща списък с наличните продукти и тяхното съдържание, списък с текущите клиенти и списък с работещите в магазина продавачи. Също така системата може да извършва различни дейности като:

* Добавяне на продукт.
* Добавяне на клиент.
* Добавяне на продавач.
* Премахване на продукт.
* Продажба на съществуващи и налични продукти.
* Запазване на информацията за всеки продукт в база данни.

Клиентите на магазина могат да купуват налични продукти, да получават информация за даден продукт, информация за продукти от даденена категория или информация за всички продукти.

Продавачите могат да обслужват клиентите и да им продават стока.

Магазинът разполага с четири категории продукти: храни, облекла, технологии или козметика. Всеки продукт има уникално име, цена и бройка в наличност. Различните видове продукти съдържат и свои данни, различни за всеки вид.

1. **Изпълнение на поставените задачи:**
2. Създаване на абстрактен клас Продукт, базов за всички категории артикули, съдържащ полета за име, цена и количество.
3. Създаване на класове, наследяващи клас Продукт. Клас Облекла, съдържащ полета за размер и дължина. Клас Козметика, съдържащ полета за държава на производство и състав. Клас Техника, притежаващ полета за тегло и години гаранция. Клас Храни, притежаващ полета за калории, грамаж и съдържание.

3.Създаване на клас Човек, съдържащ полета за име и години, наследен от два класа. Клас Продавач, съдържащ поле за брой обслужени клиенти и булева променлива, показваща дали в момента продавачът е зает. Клас Клиент съдържащ поле за брой закупени продукти.

4.Създаване на клас Система на магазин съдържащ динамичен двумерен масив за продуктите, масив за продавачите и масив за клиентите. Реализиране на нужните методи на класа, изпълняващи изискванията.

1. **Основни концепции и проблеми**

За проекта ще бъдат използвани основните принципи на обектно-ориентираното програмиране:

* Инкапсулация

При съзваването на всеки от деветте класа, бива постигната инкапсулация, а именно член-данните на всеки клас са видими само за член-функциите им и до полетата няма достъп отвън.

* Полиморфизъм

В класа Система масивът с продуктите е полиморфен, защото към него могат да бъдат добавени четири вида продукти, без значение тяхната подредба. Също така при метода, който добавя продукт, полиморфизма се проявява чрез неясното подаване на артикул към функцията. При изкарване на информацията за всички продукти също има полиморфизъм.

* Наследяване

В програмата има два базови класа и шест класа-наследници.

Най- сложният проблем в проекта е създаването на връзката на клиентите със системата, т.е. по какъв начин се осъществява комуникацията между клиенти, продавачи и магазин. Също важен етап в развитието на проекта е записването на продуктите в база данни, тъй като всеки артикул е уникален и има свои собствени член-данни, освен наследените от базовия клас. Реализирането на проекта отнема 1-2 седмици.

За осъществяване на връзката между системата, продавачите, клиентите и продуктите беше нужно създаването на няколко допълнителни функции в класа Система, които ще бъдат описани в следващите глави.

При стартиране на програмата се извършва проверка: дали съществува файл с информация за продуктите. Ако такъв съществува, системата бива създадена с информацията взета от файла. В противен случай, на екрана се изписва съобщение за създаване на система и брой на продуктите, които ще се продават в нея, с възможност за добавяне на нови такива.

От потребителите се иска при влизане в системата да си направят профил, съдържащ тяхното име и години, по този начин в системата бива добавян клиент. При вече съществуващ профил се изисква вписване в акаунт за бъдещи покупки и получаване на информация за налични продукти.

В системата има възможност да се влезе като администратор, който може да управлява добавянето и премахването на продукти, както и назначаването на продавачи.

1. **Обща архитектура на проекта**

Архитектура на продуктите:

Архитектура на клиенти и продавачи:

Архитектура на системата:

1. **Реализация и тестване**

Класове за артикули:

* Клас Продукт:

Абстрактен клас съдържащ : полета за име, цена и количество, конструктори с и без параметри, копиращ конструктор, конструктор за присвояване и деструктор. Също така: методи за получаване на име, цена и количество, чисто виртуални член-функции за получаване на тип на продукта, информацията за него и за записване във файл, член-функция за изкарване информацията за продукта на конзолата и такава за продажба на продукт.

virtual Type getPrType() const = 0; - чисто виртуална-функция, връщаща стойност от изброен тип, различна за всеки клас наследник на Продукт.

virtual void infoToStream(ostream&) const = 0; - чисто виртуална-функция, предефинирана във всеки от класовете наследяващи Продукт, която взима информацията за дадения продукт, за да може тя да бъде записана в база данни.

ostream& operator<<(ostream & os, const Product & prod)

{

prod.infoToStream(os);

return os;

}

Функция, предефинираща оператор за въвеждане в изходен поток, използваща горната функция за записване на информацията във файл.

Наследници на Продукт, съдържащи голяма четворка,член-функции за достъп и реализация на чисто виртуалните функции на клас Продукт:

* Клас Облекла
* Клас Технология
* Клас Козметика
* Клас Храна
* Клас Човек

Клас съдържащ : полета за име и години, конструктори с и без параметри, копиращ конструктор, конструктор за присвояване, деструктор и член-функции за достъп.

Клас Продавач и клас Клиент, наследяващи Човек. Продавач има поле за брой обслужени от него клиенти, а Клиент има поле за брой закупени продукти.

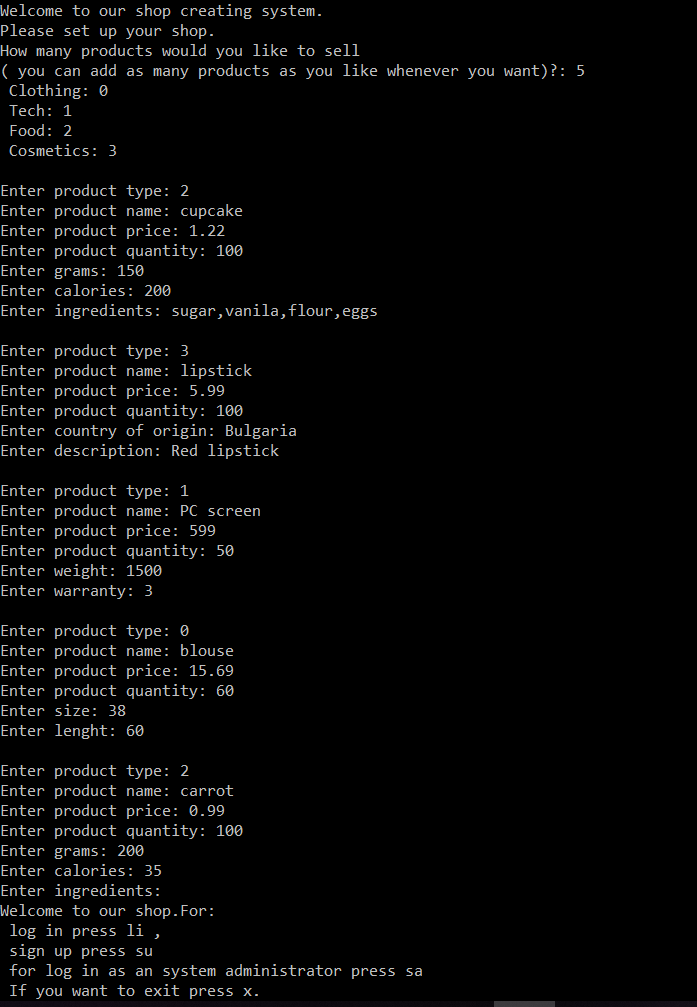
* Клас Система на магазин

Съдържащ указател към масив от указатели от тип Продукт, указател към масив от тип Продавач и указател към масив от тип Клиент, също така капацитети за масивите и текущи позиции в масивите. Освен голямата четворка, класа има член-функции за:

* Добавяне на продукт.
* Премахване на продукт.
* Получаване на цена за продукт.
* Получаване на информация за продукти.
* Продажба на продукт.
* Добавяне на клиент.
* Добавяне на продавач.
* За проверяване дали даден продукт същесвтува - използвана при заявка на клиент за закупуване продукт и при добавяне на продукт в системата от администратор.
* За проверка на наличен продавач, който да може да обслужи клиенти.
* За продаване на артикул, при наличие на продавач, чрез намаляване на количеството на продукта.
* За намиране на клиент в магазина.

Създаването на масивите се случва в конструктора, а заделената памет се освобождава в деструктора.

**Примери**:

A screenshot of a cell phone

Description automatically generated

1. **Заключение**

Системата на магазин бе разработена да удовлетвори нуждите на потенциалните клиенти относно бързо и лесно намиране на търсените продукти и тяхното закупуване. В бъдеще може да се усъвършенства, като се добави база данни за клиентите и пароли на акаунтите. Паролите на акауннтите биха допринесли за по-добра защита за потребителите на магазина.

Използвана литература:

1. Практически самоучител C++ Herbert Schildt
2. Сборник от задачи по програмиране втора част М. Тодорова, П. Армянов