**Имена:** Светослав Каменов

**Дата: 11.1.2017 Предмет: Програмиране с Java, част 1**

**имейл: [sv\_kamenov@yahoo.com](mailto:tihomir.krastev@mail.bg) GitHub :**

Пазарска кошница

**1. Условие**

Потребителят да може да избира продукт и желано количество от него с което да направи заявка към пазарската кошница. След като набере пожеланите заявки и види стойността на цялата кошница, потребителят да може да се откаже от някои от заявките , за да може да влезе в бюджета на пазаруването.

**Задачата трябва да има визуален интерфейс.**

**2. Въведение**

Воден от разбирането, че целта на този проект е да се направи едно полезно упражнение върху графичния интерфейс, авторът се е опитал да комбинира различни компоненти от java.Swing, за които си е мислил, че има някакво понятие :)

Така в центъра на вниманието беше графичният интерфейс. Всички друго трябваше го обслужва.

**3. Теория и история ...**

Въпреки, че домашните любимци са уникални явления и не би трябвало да се пазаруват на бройка, тук е реализирана точно тази идея, поради предположението, че използването на радио бутони за потребителски интерфейс е твърде елементарно и трябват комбо-боксове, спинери и други главоболни неща.

Проектът започна с опит задачата да бъде решена само с графични компоненти, които пазят цялата необходима информация и си взаимодействат. Веднага се видя, че трябва и "база данни" и веднага беше реализирана с паралелни масиви, в опит да се избегне използването на "класове" и "обекти", които още не са преподавани. После се стигна до JList и се видя, че му е трудно да помни количеството на всяка заявка и почна писането на класове за всичко, което стои зад графичния интерфейс.

**4. Използвани технологии**

Като среда за разработка е използван Eclipse. Графичният интерфейс е реализиран с библиотеката java.Swing, като всичко е писано by hand, за да се упражни работата с отделните графични компоненти. Визията на графичния интерфейс умишлено е пренебрегната, с идеята по-нататък проектът да се преработи с NetBeans, за което в момента няма време. Вниманието беше насочено към Listener интерфейсите.

**5. Инсталация и настройки**

Инсталирайте си Java , след което просто стартирайте ShoppingCartSK.java файла, който се намира в главната директория.

**6. Кратко ръководство на потребителя**

Стартирайте програмата . Интерфейсът е интуитивен и самопонятен.

От комбо-бокса си изберете продукт. Ще видите изображение и кратка информация за продукта.

От етикета под комбо-бокса ще видите наличното количество в склада.

От спинера изберете количество, натиснете бутона Add.

Така правите трансакция към заявката. В дясно ще видите оформящата се заявка като списък.

Под нея в етикетите ще видите количеството на всяка трансакция и акумулираното количество за съответния артикул от заявката.

С бутона Remove може да премахнете трансакции от заявката.

Няма Submit. Информацията си е за нас, макар, че е отразена в база данни.

**7. Примерни данни**

Следвай графичния интерфейс. Започни с въвеждане в левия панел. Довърши с десния. Готово.

**8. Описание на програмния код**

Идеята на отделните класове:

PetArticle

- характеризира артикулите, които се създават в базата данни - изображение, име, цена ...други характеристики, които така и не се ползват засега. Тук се пазят и наличното количество в склада, както и общото продадено количество (по време на сесията).

PetWarehouse

- представлява базата данни - предоставя информация за атикулите - описания, изображения, количесва. Тегли количества при трансакция и ги връща в склада при отказ на трансакция.

OrderItem

- представлява една трансакция - това, което потребителя добавя в заявката с всяко натискане на Add бутона- описва артикул и количество.

Order

- събира всички OrderItem обекти в списък и ги обслужва - добавя, премахва, вади информация за количеството на трансакцията, която е на фокус. Вади информация от склада за общото заявено количество от артикула на фокус, смята общата стойност. Order обектът е синхронизиран с JList- a, който прави потребителския интерфейс.

**9. Приноси на курсиста, ограничения и възможности за бъдещо разширение**

Доколкото в проекта има и класове , различни от стандартните класове на Java, то те биха могли да се използват и за надстройване на функционалността. Мисля, че е добра илюстрация на организацията на счетоводството - складови операции, касови операции. Остава само да се сметнат приходи, разходи, данъци, печалба и да се свърже с НАП :)

**10. Използвани източници**

- [www.oracle.com](http://www.java2s.com/Tutorial/Java/0240__Swing/UsingJPanelasacontainer.htm)

How to use GUI with Swing...