### Οντοκεντρικός Προγραμματισμός Πρότζεκτ

# Συγγραφείς:

# Χαλκιόπουλος Νικόλαος 1072572

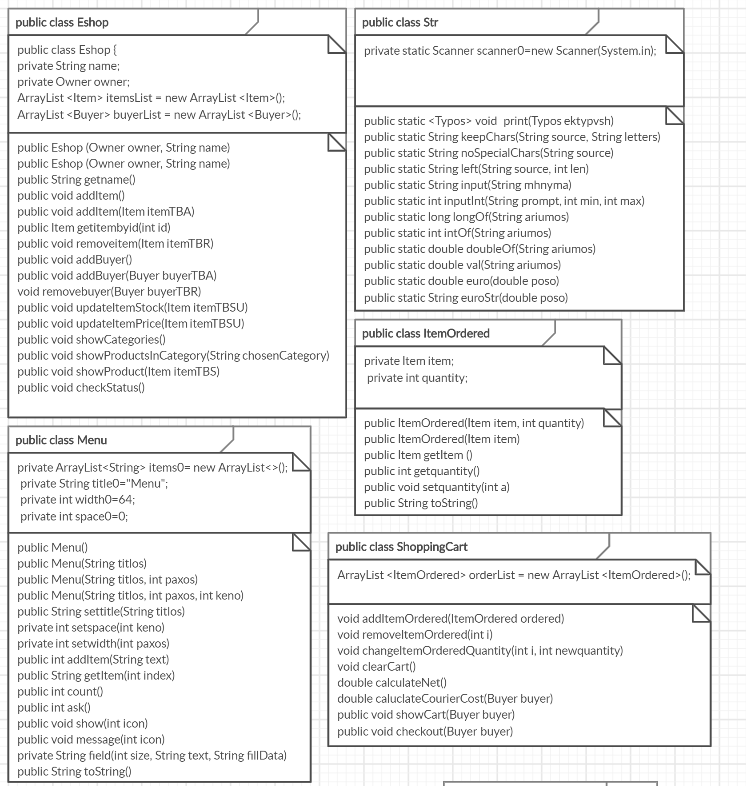
# Σφήκας Θεόδωρος 1072550

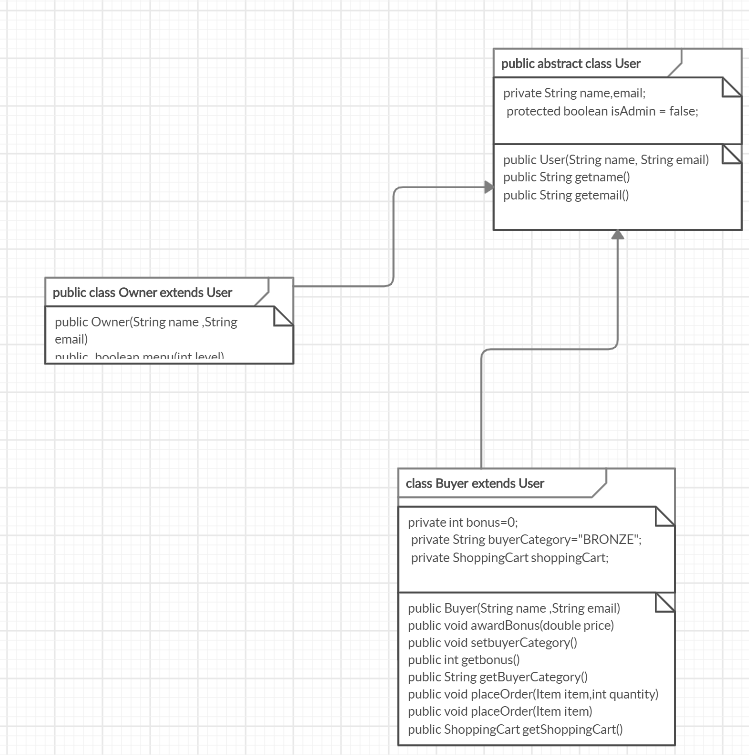
# Μουστάκη Αλέξανδρος 1072532

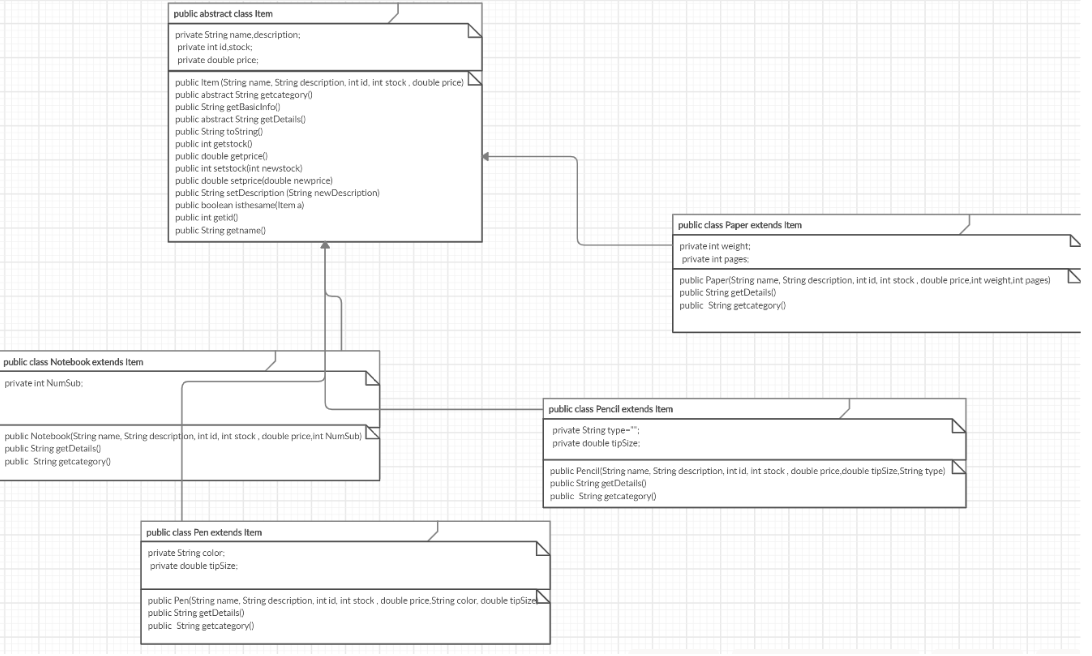
Ο Κώδικας στο github μπορεί να βρεθεί εδώ https://github.com/UnburdenSoul/java-project-2020-ardin-team.git

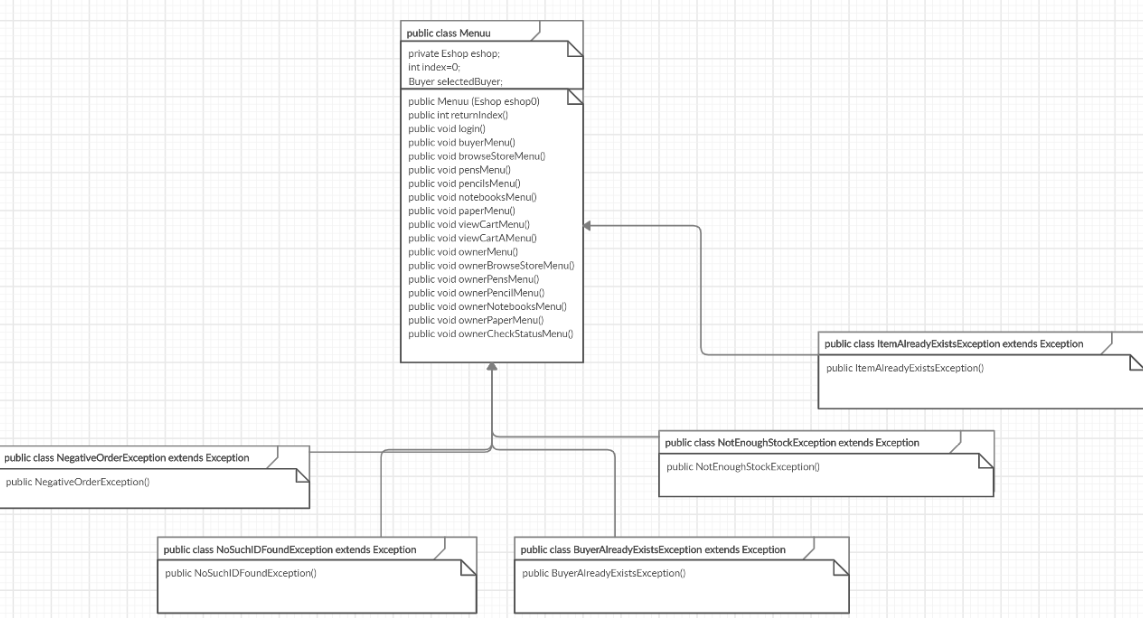
Μας ζητείται να σχεδιάςουμε και να υλοποιήσετε ένα απλοποιημένο ηλεκτρονικό κατάστημα (eshop). To ηλεκτρονικό κατάστημα εμπορεύεται προϊόντα γραφείου και υποστηρίζει διαχείριση χρηστών, διαχείριση προϊόντων, καταχώρηση και τιμολόγηση παραγγελιών. Η υλοποίηση έγινε αποκλειστικά σε Java

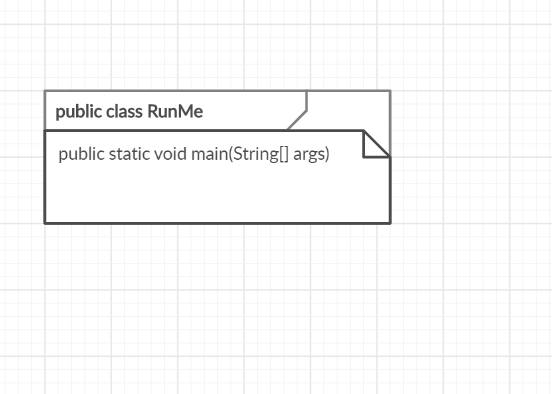
Το πρόγραμμα που δημιουργήσαμε ακολουθεί τις κλάσεις που απεικονίζονται στα παρακάτω UML διάγραμματα αφου δεν φαινεται καλά σε ένα UML διαγραμμα :











Αρχικά παρατηρούμε την Δημιουργία της abstract κλάςης User, η οποία δεν μπορεί να έχει στιγμιότυπα αλλα κληρομονεί τα χαρακτηριστικά της στις κλάσεις Owner (piou με email piou@ceid.upatras.gr) και Buyers ( Οι δύο αρχικοποιημένοι Buyers είναι ο [teo@ceid.upatras.gr](mailto:teo@ceid.upatras.gr) και ο [alex@ceid.upatras.gr](mailto:alex@ceid.upatras.gr) με usernames teo , alex αντιστοιχα. )

Ακολουθεί ο κώδικας των από πάνω κλάσεων:

**User:**

**package** Project2020;

**public** **abstract** **class** User {

**private** String name,email;

**protected** **boolean** isAdmin = **false**;

**public** User(String name, String email) {

**this**.name=name;

**this**.email=email;

}

**public** String getname()

{

**return** **this**.name;

}

**public** String getemail()

{

**return** **this**.email;

}

}

Σχόλιο: Χρησιμοποιείται μια μεταβλητή IsAdmin για την ταυτοποίηση του Owner.

**Owner:**

**package** Project2020;

**public** **class** Owner **extends** User {

**public** Owner(String name ,String email){

**super**(name , email);

**this**.isAdmin=**true**;

}

H IsAdmin παιρνει την τιμη false

**Buyer:**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Menuu:**

Το μενού αποτέλεσε μία από τις πιο αφαιρετικές και σημαντικές κλάσεις για την υλοποίηση του προβλήματος καθως και ηταν η μοναδική στην οποία αποκλείναμε σημαντικά από οποιαδήποτε άλλη κλάση. Χρησιμοποιήθηκε μια μεταβλητή Index για να καθοδηγεί το προγραμμα στην εκάστοτε μέθοδο της κλάσης menuu ώστε να εμφανίζεται η κατάλληλη διεργασία

Συγκεκριμένα ο index κάνει τα επόμενα:

|  |  |
| --- | --- |
| Index : |  |
| -1 | Κλεισιμο προγράμματος |
| 0 | Login |
| 1 | Owner |
| 2 | Buyer |
| 3 | BrowseStoreBuyer |
| 4 | ViewCartBuyer |
| 5 | PensBuyer |
| 6 | PencilBuyer |
| 7 | NotebookBuyer |
| 8 | PaperBuyer |
| 9 | ViewCartA |
| 10 | OwnerBrowseStore |
| 11 | CheckStatus |
| 12 | OwnerPen |
| 13 | OwnerPencil |
| 14 | OwnerNotebook |
| 15 | OwnerPaper |

**ShoppingCart**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**ItemOrdered**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Item**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Pen**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Pencil**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Notebook**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Paper**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**Eshop:**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

**RunMe:**

**package** Project2020;

**class** Buyer **extends** User {

**private** **int** bonus=0;

**private** String buyerCategory="BRONZE";

**private** ShoppingCart shoppingCart;

**public** Buyer(String name ,String email){

**super**(name , email);

shoppingCart= **new** ShoppingCart();

}

**public** **void** awardBonus(**double** price)

{

**this**.bonus=**this**.bonus+(**int**)price/10;

**this**.setbuyerCategory();

}

**public** **void** setbuyerCategory()

{

**if** (**this**.bonus < 100)

{**this**.buyerCategory="BRONZE";}

**else** **if** ((**this**.bonus>=100) && (**this**.bonus<200))

{**this**.buyerCategory="SILVER";}

**else**

{**this**.buyerCategory="GOLD";}

}

**public** **int** getbonus()

{

**return** **this**.bonus;

}

**public** String getBuyerCategory()

{

**return** **this**.buyerCategory;

}

**public** **void** placeOrder(Item item,**int** quantity) // εχει δυο ορισματα για ειδος αντικειμενου και ποσοτητα

{

ItemOrdered ordered= **new** ItemOrdered(item,quantity);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** **void** placeOrder(Item item) // Εχει ενα ορισμα για το ειδος του αντικειμοένου

{ //και υπονοεί πως παραγγέλνει 1 μονο τεμάχιο του ειδους

ItemOrdered ordered= **new** ItemOrdered(item);

**try** {

**this**.shoppingCart.addItemOrdered(ordered);

}

**catch**(NotEnoughStockException e){

//Θέλω να με γυρναει ένα βήμα πίσω.

}

**catch**(NegativeOrderException e) {

//Θέλω να με γυρναει ένα βήμα πίσω.

}

}

**public** ShoppingCart getShoppingCart()

{

**return** **this**.shoppingCart;

}

}

Κλάση Str : Χρησιμοποιήθηκε και δημιουργήθηκε για την ορθή λειτουργία της Scanner που είναι buggy και την διευκολυνση των προγραμματιστών στο να παιρνουν τιμές από το πληκτρολόγιο.Υπάρχουν διάφορες Static μέθοδοι τις Οποίες Δημιούργησε ο Κ.Σφήκας καθως προσπαθούσαμε να υλοποιήσουμε

Το eshop υπάρχει εκτενής σχολιασμός μέσα στο κώδικα της κλάσης Str

.