**Задание.**

1. Написать приложение «Склад». Склад служит хранилищем для товаров, товары добавляются на склад, берутся со склада, изменяются через накладную. Необходимо написать соответствующие классы, реализующие функционал для регистрации операция с товарами. Использовать коллекции ArrayList, LinkedList, HashSet, HashMap (выбор обосновать для того или иного контейнера, выбрать нужные коллекции самостоятельно).

Регистрация поступления или отгрузки товаров осуществляется по накладной (документ).

Вид накладной (взять данные отсюда):



2. Реализовать:

* добавление/удаление/изменение товаров в накладной
* добавление/удаление/изменение накладных на складе
* вывод товаров накладной (+ поиск/сортировка по разным полям)
* вывод накладных товаров (+ поиск/сортировка по разным полям)

3. Реализовать для двух любых сущностей Comparable и Comparator-интерфейсы (в случае компаратора сделать соответствующий класс).

4. Написать CLI-часть (в достаточно мере для тестирования).

***class Product implements Comparable<Product>***

package com.company.warehouse;  
  
//Класс - товар  
public class Product implements Comparable<Product> {  
  
 //поля  
 private String titleProduct; // наименование товара  
 private String unit; // единица измерения товара  
 private double price; // цена товара  
 private double quantityProduct; // количество товара  
  
 // Конструкторы  
 // 1. Конструктор без параметров  
  
 public Product() {  
 this.titleProduct = "title";  
 this.unit = "unit";  
 this.price = 0.0;  
 this.quantityProduct = 1.0;  
 }  
  
 // 2. Конструктор с 3 параметрами  
 public Product(String *titleProduct*, String *unit*, double *quantityProduct*, double *price*) {  
 this.titleProduct = *titleProduct*;  
 this.unit = *unit*;  
 this.price = *price*;  
 this.quantityProduct = *quantityProduct*;  
 }  
  
 //getters and setters  
  
 public String getTitleProduct() {  
 return titleProduct;  
 }  
  
 public void setTitleProduct(String *titleProduct*) {  
 this.titleProduct = *titleProduct*;  
 }  
  
 public String getUnit() {  
 return unit;  
 }  
  
 public void setUnit(String *unit*) {  
 this.unit = *unit*;  
 }  
  
 public double getQuantityProduct() {  
 return quantityProduct;  
 }  
  
 public void setQuantityProduct(double *quantityProduct*) {  
 this.quantityProduct = *quantityProduct*;  
 }  
  
 public double getPrice() {  
 return price;  
 }  
  
 public void setPrice(double *price*) {  
 this.price = *price*;  
 }  
  
 // Методы  
 // 1. метод toString  
  
 @Override  
 public String toString() {  
 // добавим информацию о товаре в строку  
 StringBuilder *sb* = new StringBuilder();  
 *sb*.append("Продукт \"").append(titleProduct).append("\":");  
 *sb*.append(" цена - ").append(price).append(" руб.");  
 *sb*.append(" ").append(unit).append(".");  
 return *sb*.toString();  
 }  
  
 // 2. метод compare - сравнивает товар по количеству в накладной  
 public int compareTo(Product *product*) {  
 //сравниваем количество товара  
 return Double.*compare*(quantityProduct, *product*.quantityProduct);  
 }  
  
}//close class

***class ComparatorProduct implements Comparator<Product>***

package com.company.warehouse;  
  
import java.util.Comparator;  
  
//Класс- компаратор для "Товара"  
public class ComparatorProduct implements Comparator<Product> {  
  
 //метод сравнения товаров по сумме  
 @Override  
 public int compare(Product *o1*, Product *o2*) {  
 if (*o1*.getPrice() > *o2*.getPrice()) {  
 return 1;  
 } else if (*o1*.getPrice() < *o2*.getPrice()) {  
 return -1;  
 } else return Double.*compare*(*o1*.getPrice(), *o2*.getPrice());  
 }  
}

***class ConsignmentNote implements Comparable<ConsignmentNote>***

package com.company.warehouse;  
  
import java.util.\*;  
  
  
//Класс - "Накладная"  
public class ConsignmentNote implements Comparable<ConsignmentNote> {  
  
 //поля  
 private String fromWhom; // поле накладной "От кого"  
 private String toWhom; // поле накладной "Кому"  
 private String theReason; // поле накладной "Основание выдачи"  
 private LinkedList<Product> products = new LinkedList<>(); // список товаров в накладной  
 private int numberDocument; // номер накладной  
  
  
 // Конструкторы  
 // 1. Конструктор без параметров  
  
 public ConsignmentNote() {  
 this.fromWhom = "\_\_\_\_\_\_\_\_\_\_\_";  
 this.toWhom = "\_\_\_\_\_\_\_\_\_\_\_\_\_\_";  
 this.theReason = "Пустой бланк";  
 this.products = new LinkedList<>();  
 this.numberDocument = 1;  
 }  
  
 // 2. Конструктор с 5 параметрами  
 public ConsignmentNote(int *numberDocument*, String *fromWhom*, String *toWhom*, String *theReason*) {  
 this.fromWhom = *fromWhom*;  
 this.toWhom = *toWhom*;  
 this.theReason = *theReason*;  
 this.numberDocument = *numberDocument*;  
 }  
  
 //getters and setters  
  
 public String getFromWhom() {  
 return fromWhom;  
 }  
  
 public void setFromWhom(String *fromWhom*) {  
 this.fromWhom = *fromWhom*;  
 }  
  
 public String getToWhom() {  
 return toWhom;  
 }  
  
 public void setToWhom(String *toWhom*) {  
 this.toWhom = *toWhom*;  
 }  
  
 public String getTheReason() {  
 return theReason;  
 }  
  
 public void setTheReason(String *theReason*) {  
 this.theReason = *theReason*;  
 }  
  
 public int getNumberDocument() {  
 return numberDocument;  
 }  
  
 public LinkedList<Product> getProducts() {  
 return products;  
 }  
  
 public void setProducts(LinkedList<Product> *products*) {  
 this.products = *products*;  
 }  
  
 public void setNumberDocument(int *numberDocument*) {  
 this.numberDocument = *numberDocument*;  
 }  
  
 // Методы  
 // 1. метод toString  
  
 @Override  
 public String toString() {  
 Calendar *calendar* = Calendar.*getInstance*();  
 // добавим информацию из накладной в строку  
 StringBuilder *sb* = new StringBuilder();  
 *sb*.append(" Накладная № ").append(numberDocument);  
 *sb*.append("\n \" ").append(*calendar*.get(Calendar.*DAY\_OF\_MONTH*)).append(" \"");  
 *sb*.append(" ").append(*calendar*.get(Calendar.*MONTH*));  
 *sb*.append(" ").append(*calendar*.get(Calendar.*YEAR*));  
 *sb*.append("\nОт кого: ").append(fromWhom);  
 *sb*.append("\nКому: ").append(toWhom);  
 *sb*.append("\nОснование: ").append(theReason).append("\n");  
 return *sb*.toString();  
 }  
  
 // 2. метод вывода шапки таблицы накладной  
 public void invoiceTable() {  
 int *count* = 1;  
 System.*out*.println("+-------+--------------+-------+--------+---------+---------+");  
 System.*out*.println("| № п/п | Наименование |Ед.изм.| Кол-во | Цена | Сумма |");  
 System.*out*.println("+-------+--------------+-------+--------+---------+---------+");  
 for (Product *list* : products) {  
 System.*out*.format("| %-6d| %-13s| %-6s| %-7.2f| %-8.2f| %-8.2f|%n", *count*++, *list*.getTitleProduct(),  
 *list*.getUnit(), *list*.getQuantityProduct(), *list*.getPrice(), amountProduct(*list*));  
 //добавили сумму спомощью метода amountProduct()  
 }  
 }  
  
 // 3. метод добавления товара в накладную  
 public void addProduct(Product *product*) throws Exception {  
 for (Product *p* : products) {  
 if (*p* == null) {  
 throw new Exception("Товар не существует!");  
 }  
 }  
 products.add(*product*);  
  
 }  
  
  
 // 4. метод удаления товара из накладную  
 public void removeProduct(Product *product*) throws Exception {  
 for (Product *p* : products) {  
 if (*p* == null) {  
 throw new Exception("Товар не существует!");  
 }  
 }  
 products.remove(*product*);  
 }  
  
  
 // 5. метод изменения полей товара  
 public void changeProduct() throws Exception {  
 Scanner *in* = new Scanner(System.*in*);  
 String *menu*;//меню  
 String *valueTitle*;  
 double *valuePrice*;  
 double *valueQuantity*;  
 String *valueUnit*;  
 for (Product *product* : products) {  
 if (*product* == null) {  
 throw new Exception("Товар не существует!");  
 }  
 }  
 for (Product *p* : products) {  
 System.*out*.println("0 - Выбрать следующий продукт ");  
 System.*out*.println("1 - Изменить имя товара " + *p*.getTitleProduct() + " ? ");  
 System.*out*.println("2 - Изменить цену товара " + *p*.getTitleProduct() + " - " + *p*.getPrice() + " ? ");  
 System.*out*.println("3 - Изменить количество товара " + *p*.getTitleProduct() + " - " + *p*.getQuantityProduct() + " ? ");  
 System.*out*.println("4 - Изменить единицу измерения товара " + *p*.getTitleProduct() + " - " + *p*.getUnit() + " ? ");  
 System.*out*.println("5 - Назад ");  
 *menu* = *in*.nextLine();  
 if ("0".equals(*menu*)) {  
 continue;//выбор следующей накладной  
 }  
  
 if ("1".equals(*menu*)) {  
 System.*out*.println("Введите новое имя товара: ");  
 *valueTitle* = *in*.nextLine();  
 *p*.setTitleProduct(*valueTitle*);  
 System.*out*.println(*p*);  
 break;  
 }  
 if ("2".equals(*menu*)) {  
 System.*out*.println("Введите новую цену товара ");  
 *valuePrice* = *in*.nextInt();  
 *p*.setPrice(*valuePrice*);  
 System.*out*.println(*p*);  
 break;  
 }  
 if ("3".equals(*menu*)) {  
 System.*out*.println("Введите количество товара ");  
 *valueQuantity* = *in*.nextInt();  
 *p*.setQuantityProduct(*valueQuantity*);  
 System.*out*.println(*p*);  
 break;  
 }  
 if ("4".equals(*menu*)) {  
 System.*out*.println("Введите единицу измерения товара ");  
 *valueUnit* = *in*.nextLine();  
 *p*.setUnit(*valueUnit*);  
 System.*out*.println(*p*);  
 break;  
 }  
 }  
 }  
  
 // 6. метод вывода всех товаров накладной  
 public void printProductOfConsignment() throws Exception {  
 StringBuilder *sb* = new StringBuilder();  
 for (Product *p* : products) {  
 if (*p* == null) {  
 throw new Exception("Товар не существует!");  
 }  
 *sb*.append(*p*.getTitleProduct());  
 *sb*.append("\n");  
 }  
 System.*out*.print(*sb*);  
 }  
  
  
 // 7. метод расчёта суммы товара  
 public double amountProduct(Product *product*) {  
 return *product*.getQuantityProduct() \* *product*.getPrice();  
 }  
  
 // 8. метод compare - сравнивает накладные по номеру  
 public int compareTo(ConsignmentNote *product*) {  
 //сравниваем накладные по номеру  
 return Double.*compare*(numberDocument, *product*.numberDocument);  
 }  
  
}//class close

***class ComparatorConsignmentNote implements Comparator<ConsignmentNote>***

package com.company.warehouse;  
  
import java.util.Comparator;  
  
//Класс- компаратор для "Накладной"  
public class ComparatorConsignmentNote implements Comparator<ConsignmentNote> {  
  
 //метод сравнения накладных по полю "От кого"  
 @Override  
 public int compare(ConsignmentNote *o1*, ConsignmentNote *o2*) {  
  
 String *fromWhom* = *o1*.getFromWhom().toUpperCase();  
 String *fromWhom2* = *o2*.getFromWhom().toUpperCase();  
  
 // в порядке возрастания  
 return *fromWhom*.compareTo(*fromWhom2*);  
 //в порядке убывания  
 // вернуть fromWhom2.compareTo (fromWhom);  
 }  
}

***class PowerOfAttorney***

package com.company.warehouse;  
  
//Класс "доверенность"  
public class PowerOfAttorney {  
  
 //поля  
 private final String name = "Доверенность"; // доверенность  
 private int number; // номер доверенности  
 private int day\_of\_month; // день выдачи доверенности  
 private int month; // месяц выдачи доверенности  
 private int year; // год выдачи доверенности  
  
 // Конструкторы  
 // 1. Конструктор без параметров  
  
 public PowerOfAttorney() {  
 this.number = 1;  
 this.day\_of\_month = 1;  
 this.month = 1;  
 this.year = 2000;  
 }  
  
 // 2. Конструктор с 4 параметрами  
  
 public PowerOfAttorney(int *number*, int *day\_of\_month*, int *month*, int *year*) {  
 this.number = *number*;  
 this.day\_of\_month = *day\_of\_month*;  
 this.month = *month*;  
 this.year = *year*;  
 }  
  
 @Override  
 public String toString() {  
 StringBuilder *sb* = new StringBuilder();  
 *sb*.append(name).append(" № ").append(number).append(" от \" ").append(day\_of\_month).append(" \"");  
 *sb*.append(" ").append(month).append(" ").append(year);  
 return *sb*.toString();  
 }  
}//class close

***class Stock***

package com.company.warehouse;  
  
import java.util.\*;  
  
//Класс - Склад  
public class Stock {  
  
 //поля  
 private ConsignmentNote note; // накладная на складе  
 private LinkedList<ConsignmentNote> document = new LinkedList<>();//список накладных на складе  
 private LinkedList<Product> products = new LinkedList<>(); // список товаров на складе  
 // Конструкторы  
 // 1. Конструктор без параметров  
  
 public Stock() {  
 this.note = new ConsignmentNote();  
 }  
  
 // 2. Конструктор с одним параметром  
 public Stock(LinkedList<ConsignmentNote> *document*) {  
 this.document = *document*;  
 }  
  
 //getters  
 public LinkedList<ConsignmentNote> getDocument() {  
 return document;  
 }  
  
 // Методы  
 // 1. метод toString  
 @Override  
 public String toString() {  
 // добавим информацию о товаре в строку  
 StringBuilder *sb* = new StringBuilder();  
 for (ConsignmentNote *list* : document) {  
 *sb*.append("\nНакладная № ").append(*list*.getNumberDocument());  
 }  
 return *sb*.toString();  
 }  
  
 // 2. метод добавления товара на склад  
 public void addProductInStock(Product *product*) throws Exception {  
 for (Product *p* : products) {  
 if (*p* == null) {  
 throw new Exception("Товар не существует!");  
 }  
 }  
 products.add(*product*);  
  
 }  
  
 // 3. метод добавления накладной на склад  
 public void addConsignmentNote(ConsignmentNote *doc*) throws Exception {  
 for (ConsignmentNote *note* : document) {  
 if (*note* == null) {  
 throw new Exception("Накладная не существует!");  
 } else if (*note*.getNumberDocument() == *doc*.getNumberDocument()) {  
 throw new Exception("Накладная с номером " + *doc*.getNumberDocument() + " уже существует, измените номер!!!");  
 }  
 }  
 document.add(*doc*);  
 }  
  
 // 4. метод удаления накладной из склада  
 public void removeConsignmentNote() throws Exception {  
 Scanner *in* = new Scanner(System.*in*);  
 String *menu*;  
 for (ConsignmentNote *note* : document) {  
 if (*note* == null) {  
 throw new Exception("Накладная не существует!");  
 }  
 }  
 for (ConsignmentNote *n* : document) {  
 System.*out*.println("1 - Удалить накладную № " + *n*.getNumberDocument() + " ? ");  
 System.*out*.println("2 - Выбрать следующую накладную ");  
 System.*out*.println("3 - Назад ");  
  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 document.remove(*n*);  
 break;  
 }  
 if ("2".equals(*menu*)) {  
 continue;//выбор следующей накладной  
 }  
 if ("3".equals(*menu*)) {  
 break;  
 }  
 }  
 }  
  
 // 5. метод изменения полей накладной  
 public void changeConsignmentNote() throws Exception {  
 Scanner *in* = new Scanner(System.*in*);  
 String *menu*;//меню  
 int *valueNumber*;//меню  
 String *valueStr*;//меню  
 for (ConsignmentNote *note* : document) {  
 if (*note* == null) {  
 throw new Exception("Накладная не существует!");  
 }  
 }  
 for (ConsignmentNote *n* : document) {  
 System.*out*.println("0 - Выбрать следующую накладную ");  
 System.*out*.println("1 - Изменить поле накладной " + *n*.getFromWhom() + " ? ");  
 System.*out*.println("2 - Изменить поле накладной " + *n*.getToWhom() + " ? ");  
 System.*out*.println("3 - Изменить поле накладной " + *n*.getTheReason() + " ? ");  
 System.*out*.println("4 - Изменить поле накладной " + *n*.getNumberDocument() + " ? ");  
 System.*out*.println("5 - Назад ");  
 *menu* = *in*.nextLine();  
  
 if ("0".equals(*menu*)) {  
 continue;//выбор следующей накладной  
 }  
  
 if ("1".equals(*menu*)) {  
 System.*out*.println("Введите новое поле накладной ");  
 *valueStr* = *in*.nextLine();  
 *n*.setFromWhom(*valueStr*);  
 System.*out*.println(*n*);  
 break;  
 }  
 if ("2".equals(*menu*)) {  
 System.*out*.println("Введите новое поле накладной ");  
 *valueStr* = *in*.nextLine();  
 *n*.setToWhom(*valueStr*);  
 System.*out*.println(*n*);  
 break;  
 }  
 if ("3".equals(*menu*)) {  
 System.*out*.println("Введите новое поле накладной ");  
 *valueStr* = *in*.nextLine();  
 *n*.setTheReason(*valueStr*);  
 System.*out*.println(*n*);  
 break;  
 }  
 if ("4".equals(*menu*)) {  
 System.*out*.println("Введите новое поле накладной ");  
 *valueNumber* = *in*.nextInt();  
 *n*.setNumberDocument(*valueNumber*);  
 System.*out*.println(*n*);  
 break;  
 }  
 }  
 }  
  
 // 6. метод поиска накладной  
 public void searchConsignmentNote() {  
 System.*out*.println("Введите номер накладной:");  
 Scanner *in* = new Scanner(System.*in*);  
 int *numberDocument* = *in*.nextInt();  
 for (ConsignmentNote *note* : document) {  
 if (*numberDocument* == *note*.getNumberDocument()) {  
 System.*out*.println(*note*);  
 return;  
 }  
 }  
 System.*out*.println("Накладной с таким номером нет на складе");  
 }  
  
  
 // 7. вывод меню  
 public void showMenu() throws Exception {  
 Menu.*showMenu*();  
 }  
  
 // 8. метод поиска товара  
 public void searchProduct() {  
 System.*out*.println("Введите наименование товара:");  
 Scanner *in* = new Scanner(System.*in*);  
 String *titleProduct* = *in*.nextLine();  
 for (Product *p* : products) {  
 if (Objects.*equals*(*titleProduct*, *p*.getTitleProduct())) {  
 System.*out*.println(*p*);  
 return;  
 }  
 }  
 System.*out*.println("Товара с таким наименованием нет на складе");  
 }  
  
}//class close

***class Menu***

package com.company.warehouse;  
  
import java.util.Collections;  
import java.util.Scanner;  
  
//Класс - Меню  
public class Menu {  
  
 //Вывод накладной  
 public static void showConsignmentNote(ConsignmentNote *note*, PowerOfAttorney *document*) {  
 //выводим поля накладной  
 System.*out*.println(*note*);  
 //выводим доверенность  
 System.*out*.println(*document*);  
 //выводим таблицу в накладной  
 *note*.invoiceTable();  
 System.*out*.println();  
 }  
  
 public static void showMenu() throws Exception {  
 //Создали сканер  
  
 Scanner *in* = new Scanner(System.*in*);  
 String *menu*;// переменная для ввода меню  
 System.*out*.println();  
 //Создадим склад  
 Stock *stock* = new Stock();  
  
 //Создадим товары  
  
 Product *screw* = new Product("Винты", "шт.", 10, 15.85);  
 Product *nails* = new Product("Гвозди", "кг.", 3.0, 50.15);  
 Product *paper* = new Product("Бумага", "уп.", 4.0, 90.99);  
 Product *pencil* = new Product("Карандаш", "шт.", 20.0, 9.50);  
 Product *ductTape* = new Product("Скотч", "шт.", 20.0, 70.0);  
 Product *paint* = new Product("Краска", "кг.", 7.0, 100.0);  
 Product *antifreeze* = new Product("Антифриз", "л.", 5.0, 220.0);  
 Product *monitor* = new Product("Монитор", "шт.", 1.0, 11300.0);  
  
 // Создаём накладные, и добавим в них товары  
  
 ConsignmentNote *number\_1* = new ConsignmentNote(14, "Шпак Н.С.", "Петров Д.А.",  
 "Для нужд отдела");  
 ConsignmentNote *number\_2* = new ConsignmentNote(20, "Иванов А.С.", "Королев П.А.",  
 "Распоряжение директора");  
 ConsignmentNote *number\_3* = new ConsignmentNote(32, "Абрамов А.С.", "Никулин Ю.А.",  
 "Для служебного автомобиля");  
  
 //Пустой бланк накладной  
 ConsignmentNote *number\_4* = new ConsignmentNote();  
  
 ConsignmentNote *number\_5* = new ConsignmentNote(32, " Дудкин Д.Д.", "Цветкова И.М",  
 "Черновик, накладная непонадобилась");  
  
 //создаём доверенность на накладную №14  
 PowerOfAttorney *data\_1* = new PowerOfAttorney(15, 30, 3, 2022);  
  
 //создаём доверенность на накладную №20  
 PowerOfAttorney *data\_2* = new PowerOfAttorney(3, 10, 4, 2022);  
  
 //создаём доверенность на накладную №32  
 PowerOfAttorney *data\_3* = new PowerOfAttorney(25, 17, 5, 2022);  
 //вывод накладной на экран  
  
 //создаём доверенность на пустую накладную №1  
 PowerOfAttorney *data\_4* = new PowerOfAttorney();  
  
 //добавляем товары в накладные  
 try {  
 //добавление товаров в накладную №14  
 *number\_1*.addProduct(*screw*);  
 *number\_1*.addProduct(*nails*);  
 *number\_1*.addProduct(*paper*);  
 *number\_1*.addProduct(*pencil*);  
 } catch (Exception *ex*) {  
 System.*out*.println(*ex*.getMessage());  
 }  
  
 try {  
 //добавление товаров в накладную №20  
 *number\_2*.addProduct(*ductTape*);  
 *number\_2*.addProduct(*paint*);  
 *number\_2*.addProduct(*monitor*);  
 } catch (Exception *ex*) {  
 System.*out*.println(*ex*.getMessage());  
 }  
 try {  
 //добавление товаров в накладную №32  
 *number\_3*.addProduct(*antifreeze*);  
 } catch (Exception *ex*) {  
 System.*out*.println(*ex*.getMessage());  
 }  
  
 // Меню  
 while (true) {  
 System.*out*.println("\nВыберите пункт меню: ");  
 System.*out*.println("1 - Довавить товар и накладные на склад и вывести на экран ");  
 System.*out*.println("2 - Выбрать накладную и изменить товар в накладной ");  
 System.*out*.println("3 - Инфррмация о товаре на складе ");  
 System.*out*.println("4 - Удалить товар из накладной ");  
 System.*out*.println("5 - Удалить накладную со склада ");  
 System.*out*.println("6 - Изменить накладную");  
 System.*out*.println("7 - Поиск накладной по номеру");  
 System.*out*.println("8 - Поиск товара по имени");  
 System.*out*.println("9 - выход ");  
 *menu* = *in*.nextLine();  
  
 if ("1".equals(*menu*)) {  
 System.*out*.println("Накладные на складе: ");  
 try {  
 *stock*.addConsignmentNote(*number\_2*);  
 *stock*.addConsignmentNote(*number\_3*);  
 *stock*.addConsignmentNote(*number\_1*);  
 *stock*.addConsignmentNote(*number\_4*);  
 *stock*.addConsignmentNote(*number\_5*);  
 } catch (Exception *ex*) {  
 System.*out*.println(*ex*.getMessage());  
 }  
 //Добавление товара на складе:  
 try {  
 *stock*.addProductInStock(*screw*);  
 *stock*.addProductInStock(*nails*);  
 *stock*.addProductInStock(*paint*);  
 *stock*.addProductInStock(*pencil*);  
 *stock*.addProductInStock(*ductTape*);  
 *stock*.addProductInStock(*paint*);  
 *stock*.addProductInStock(*monitor*);  
 *stock*.addProductInStock(*antifreeze*);  
 } catch (Exception *ex*) {  
 System.*out*.println(*ex*.getMessage());  
 }  
 System.*out*.println(*stock*);  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 - Сортировать накладные по номеру? ");  
 System.*out*.println("2 - Сортировать накладные по ФИО \"От кого\"? ");  
 System.*out*.println("3 - Назад? ");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 Collections.*sort*(*stock*.getDocument());  
 System.*out*.println(*stock*);  
 System.*out*.println();  
 break;  
 }  
 if ("2".equals(*menu*)) {  
 *stock*.getDocument().sort(new ComparatorConsignmentNote());  
 System.*out*.println();  
 break;  
 }  
 if ("3".equals(*menu*)) {  
 break;  
 }  
 }  
 }  
 if ("2".equals(*menu*)) {  
 System.*out*.println(*stock*);  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 - Накладная № " + *number\_4*.getNumberDocument());  
 System.*out*.println("2 - Накладная № " + *number\_1*.getNumberDocument());  
 System.*out*.println("3 - Накладная № " + *number\_2*.getNumberDocument());  
 System.*out*.println("4 - Накладная № " + *number\_3*.getNumberDocument());  
 System.*out*.println("0 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *showConsignmentNote*(*number\_4*, *data\_4*);  
  
 }  
 if ("2".equals(*menu*)) {  
 //сортировка товаров в накладной по количеству товаров  
 Collections.*sort*(*number\_1*.getProducts());  
 *showConsignmentNote*(*number\_1*, *data\_1*);  
 while (true) {  
 System.*out*.println("Используется сортировка товара по количеству ");  
 System.*out*.println("1 -Изменить товар? ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
  
 *number\_1*.changeProduct();  
 *showConsignmentNote*(*number\_1*, *data\_1*);  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }  
 }  
  
 if ("3".equals(*menu*)) {  
 new ComparatorProduct();  
 *showConsignmentNote*(*number\_2*, *data\_2*);  
 while (true) {  
 System.*out*.println("Используется сортировка товара по цене от меньшей к большей ");  
 System.*out*.println("1 -Изменить товар? ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *number\_2*.changeProduct();  
 *showConsignmentNote*(*number\_2*, *data\_2*);  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }  
 }  
 if ("4".equals(*menu*)) {  
 *showConsignmentNote*(*number\_3*, *data\_3*);  
 while (true) {  
 System.*out*.println("1 -Изменить товар? ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *number\_3*.changeProduct();  
 *showConsignmentNote*(*number\_3*, *data\_3*);  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }  
 }  
 if ("0".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }//2  
  
 if ("3".equals(*menu*)) {  
 System.*out*.println("Товары на складе : ");  
 try {  
 *number\_1*.printProductOfConsignment();  
 *number\_2*.printProductOfConsignment();  
 *number\_3*.printProductOfConsignment();  
 *number\_4*.printProductOfConsignment();  
 } catch (Exception *ex*) {  
 System.*out*.println(*ex*.getMessage());  
 }  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 - Информация о товаре: " + *screw*.getTitleProduct());  
 System.*out*.println("2 - Информация о товаре: " + *nails*.getTitleProduct());  
 System.*out*.println("3 - Информация о товаре: " + *paper*.getTitleProduct());  
 System.*out*.println("4 - Информация о товаре: " + *pencil*.getTitleProduct());  
 System.*out*.println("5 - Информация о товаре: " + *ductTape*.getTitleProduct());  
 System.*out*.println("6 - Информация о товаре: " + *paint*.getTitleProduct());  
 System.*out*.println("7 - Информация о товаре: " + *monitor*.getTitleProduct());  
 System.*out*.println("8 - Информация о товаре: " + *antifreeze*.getTitleProduct());  
 System.*out*.println("0 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 System.*out*.println(*screw*);  
  
 }  
 if ("2".equals(*menu*)) {  
 System.*out*.println(*nails*);  
  
 }  
 if ("3".equals(*menu*)) {  
 System.*out*.println(*paper*);  
  
 }  
 if ("4".equals(*menu*)) {  
 System.*out*.println(*pencil*);  
  
 }  
 if ("5".equals(*menu*)) {  
 System.*out*.println(*ductTape*);  
  
 }  
 if ("6".equals(*menu*)) {  
 System.*out*.println(*paint*);  
  
 }  
 if ("7".equals(*menu*)) {  
 System.*out*.println(*monitor*);  
  
 }  
 if ("8".equals(*menu*)) {  
 System.*out*.println(*antifreeze*);  
  
 }  
 if ("0".equals(*menu*)) {  
 break;  
 }  
 }  
 }  
  
 if ("4".equals(*menu*)) {  
 System.*out*.println(*stock*);  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("Выбирете накладную: ");  
 System.*out*.println("1 - Накладная № " + *number\_1*.getNumberDocument());  
 System.*out*.println("2 - Накладная № " + *number\_2*.getNumberDocument());  
 System.*out*.println("3 - Накладная № " + *number\_3*.getNumberDocument());  
 System.*out*.println("4 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 while (true) {  
 System.*out*.println("1 - Удалить " + *screw*.getTitleProduct());  
 System.*out*.println("2 - Удалить " + *nails*.getTitleProduct());  
 System.*out*.println("3 - Удалить " + *paper*.getTitleProduct());  
 System.*out*.println("4 - Удалить " + *pencil*.getTitleProduct());  
 System.*out*.println("5 - Назад ");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *number\_1*.removeProduct(*screw*);  
 *showConsignmentNote*(*number\_1*, *data\_1*);  
  
 }  
  
 if ("2".equals(*menu*)) {  
 *number\_1*.removeProduct(*nails*);  
 *showConsignmentNote*(*number\_1*, *data\_1*);  
  
 }  
 if ("3".equals(*menu*)) {  
 *number\_1*.removeProduct(*paper*);  
 *showConsignmentNote*(*number\_1*, *data\_1*);  
  
 }  
 if ("4".equals(*menu*)) {  
 *number\_1*.removeProduct(*pencil*);  
 *showConsignmentNote*(*number\_1*, *data\_1*);  
 }  
 if ("5".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
  
 if ("2".equals(*menu*)) {  
 while (true) {  
 System.*out*.println("1 - Удалить " + *ductTape*.getTitleProduct());  
 System.*out*.println("2 - Удалить " + *paint*.getTitleProduct());  
 System.*out*.println("3 - Удалить " + *monitor*.getTitleProduct());  
 System.*out*.println("4 - Назад ");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *number\_2*.removeProduct(*ductTape*);  
 *showConsignmentNote*(*number\_2*, *data\_2*);  
  
 }  
  
 if ("2".equals(*menu*)) {  
 *number\_2*.removeProduct(*paint*);  
 *showConsignmentNote*(*number\_2*, *data\_2*);  
  
 }  
 if ("3".equals(*menu*)) {  
 *number\_2*.removeProduct(*monitor*);  
 *showConsignmentNote*(*number\_2*, *data\_2*);  
  
 }  
 if ("4".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
 if ("3".equals(*menu*)) {  
 while (true) {  
 System.*out*.println("1 - Удалить " + *antifreeze*.getTitleProduct());  
 System.*out*.println("2 - Назад ");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *number\_3*.removeProduct(*antifreeze*);  
 *showConsignmentNote*(*number\_3*, *data\_3*);  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
 if ("4".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
 if ("5".equals(*menu*)) {  
 System.*out*.println(*stock*);  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 -Удалить накладную? ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *stock*.removeConsignmentNote();  
 System.*out*.println(*stock*);  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
 if ("6".equals(*menu*)) {  
 System.*out*.println(*stock*);  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 -Изменить накладную? ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *stock*.changeConsignmentNote();  
 System.*out*.println(*stock*);  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
  
 if ("7".equals(*menu*)) {  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 - Для поиска накладной по номеру ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *stock*.searchConsignmentNote();  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
 if ("8".equals(*menu*)) {  
 System.*out*.println();  
 while (true) {  
 System.*out*.println("1 - Для поиска товара по имени ");  
 System.*out*.println("2 - Возврат в меню");  
 *menu* = *in*.nextLine();  
 if ("1".equals(*menu*)) {  
 *stock*.searchProduct();  
 }  
 if ("2".equals(*menu*)) {  
 break;  
 }  
 }//while  
 }  
  
 if ("9".equals(*menu*)) {  
 System.*out*.println("Вы покинули склад ");  
 break;  
 }  
 }//while 1  
  
 }//main  
  
}//class close

***class ShowStock***

package com.company.warehouse;  
  
//Класс Вывода информации по складу  
public class ShowStock {  
  
 public static void main(String[] *args*) throws Exception {  
  
 //Создадим склад  
 Stock *stock* = new Stock();  
 //вызвали меню склада  
 *stock*.showMenu();  
  
 }//main  
  
}//close class

***Тесты:***



