МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ

"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

ЛАБОРАТОРНА РОБОТА №6

Робота з колекція в мові прогрaмування Java

Виконав: студент 1 курсу

група: ІО-21

Кузьменко В. З.

Номер залікової книжки № 2114

Прийняв: Невдащенко М.В.

Дата:

Оцінка:

Київ - 2013

Тема: Робота з колекція в мові програмування Java.

Мета: Здобуття навичок у створенні власних та використанні стандартних колекцій в мові програмування Java.

Варіант завдання: Інтерфейс List

Внутрішня структура колекції: Масив із початковою кількістю елементів на збільшенням її кількості на 30%.

Опис програми. Принцип роботи програми з лабораторної роботи №5 не змінився. Програма працює з типізованою колекцією та виконує ті ж самі завдання.

Програмний код:

Клас з головним методом:

***package*** *lab6;*

***import*** *java.util.ListIterator;*

***public******class*** *Lab6 {*

*/\*\*Головний метод*

*\** ***@param*** *args*

*\*/*

***public******static******void*** *main(String[] args) {*

*/\*\**

*\* створюємо різну амуніцію*

*\*/*

*Armor armor =* ***new*** *Armor("armor1", 100, 20.5);*

*Shield shield =* ***new*** *Shield("shield1", 90, 56.7);*

*Sword sword =* ***new*** *Sword("sword1", 120, 10.8);*

*Helmet helmet =* ***new*** *Helmet("helmet1", 60, 3.8);*

*/\*\**

*\* ствоорюємо масив амуніції для екіпіровки*

*\* одного лицаря*

*\*/*

*MyCollection am =* ***new*** *MyCollection();*

*am.add(helmet);*

*am.add(shield);*

*am.add(armor);*

*am.add(sword);*

*ListIterator<Ammunition> itr = am.listIterator(0);*

*Helmet hel =* ***new*** *Helmet("sdfsdf",20 ,20);*

***while*** *(itr.hasNext()) {*

*Ammunition ammunition = itr.next();*

*System.out.println(ammunition.getInformation());*

*}*

*am.remove(helmet);*

*am.add(hel);*

*/\*\**

*\* перевіряємо чи набрана амуніція підходить для*

*\* одного лицаря*

*\*/*

***if*** *(am.isCorrectAmmunition()) {*

*Chevalier chevalier =* ***new*** *Chevalier(am);*

*System.out.print("Вартість амуніції лицаря: ");*

*/\*\*рахуємо вартість амуніції лицаря\*/*

*System.out.println(chevalier.getPriceAmmunition());*

*/\*\*сортуємо за вагою\*/*

*chevalier.sortAmmunition(am);*

*System.out.println("Відсортована амуніція:");*

*ListIterator<Ammunition> itr1 = chevalier.sortAmmunition(am).listIterator();*

***while*** *(itr1.hasNext()) {*

*System.out.println(itr1.next().getInformation());*

*}*

*System.out.println();*

*}*

*/\*\*діапазон цін для амуніції\*/*

***int*** *minPrice = 15;* ***int*** *maxPrice = 100;*

*System.out.println("Aмуніція із вибраного діапазону цін: " + minPrice*

*+ "-" + maxPrice);*

*/\*\*пошук амуніції по заданому діапазоні\*/*

*itr = am.listIterator();*

***while*** *(itr.hasNext()) {*

*Ammunition ammunition = itr.next();*

***if*** *((ammunition.getPrice()<=maxPrice)&&(ammunition.getPrice()>=minPrice)){*

*System.out.println(ammunition.getInformation());*

*}*

*}*

*}*

*}*

Клас MyCollection

***package*** *lab6;*

*/\*\**

*\* Клас MyCollection реалізує інтерфейс List*

*\** ***@author*** *Lenovo*

*\*/*

***import*** *java.util.ArrayList;*

***import*** *java.util.Collection;*

***import*** *java.util.Iterator;*

***import*** *java.util.List;*

***import*** *java.util.ListIterator;*

***public******class*** *MyCollection* ***implements*** *List<Ammunition> {*

*/\*\**

*\* Масив Амуніції*

*\*/*

***private*** *Ammunition [] ammunitions;*

*/\*\**

*\** ***@param*** *кількість елементів*

*\*/*

***private******int*** *REALSIZE;*

*/\*\**

*\* Клас MyListIterator реалізовує інтерфейс ListIterator<Ammunition>*

*\** ***@author*** *Lenovo*

*\**

*\*/*

***public******class*** *MyListIterator* ***implements*** *ListIterator<Ammunition> {*

*/\*\**

*\* next наступний елемент*

*\* previous попередній елемент*

*\* item теперішшній елемент*

*\*/*

*Ammunition next;*

*Ammunition previous;*

*Ammunition item;*

***int*** *index=0;*

*/\*\**

*\* Конструктор класу, отримує параметр і*

*\** ***@param*** *i*

*\* ітератор встановлюється в позицію і*

*\*/*

***public*** *MyListIterator(****int*** *i) {*

*index = i;*

*item=ammunitions[index];*

*index++;*

*next=ammunitions[index];*

***if****(i==0){*

*previous=****null****;*

*}*

***else****{*

*previous=ammunitions[i-1];*

*}*

*}*

*/\*\**

*\* конструктор класу*

*\* ітератор встановлений в позицію і=0*

*\*/*

***public*** *MyListIterator() {*

*index = 0;*

*item=ammunitions[index];*

*index++;*

*next=ammunitions[index];*

*previous=****null****;*

*}*

*/\*\**

*\* метод додає до колекції елемент е*

*\*/*

*@Override*

***public******void*** *add(Ammunition e) {*

*item=e;*

***if*** *(index ==ammunitions.length){*

*next=****null****;*

*previous=ammunitions[index];*

*index++;*

*}*

***else****{*

*index++;*

*previous=ammunitions[index-1];*

*}*

*MyCollection.****this****.add(e);*

*}*

*/\*\**

*\* метод перевіряє чи є у колекції наступний елемент*

*\*/*

*@Override*

***public******boolean*** *hasNext() {*

***return*** *ammunitions[index-1]!=****null****;*

*}*

*/\*\**

*\* метод перевіряє чи є у колекції попередній елемент*

*\*/*

*@Override*

***public******boolean*** *hasPrevious() {*

***return*** *previous!=****null****;*

*}*

*/\*\**

*\* повертає наступний елемент*

*\*/*

*@Override*

***public*** *Ammunition next() {*

*Ammunition it = item;*

*item=next;*

*previous=it;*

***if*** *(index ==ammunitions.length){*

*next=****null****;*

*}*

***else****{*

*index++;*

*next = ammunitions[index];*

*previous=ammunitions[index-1];*

*}*

***return*** *it;*

*}*

*/\*\**

*\* повертає позицію наступного елемента*

*\*/*

*@Override*

***public******int*** *nextIndex() {*

***return*** *index;*

*}*

*/\*\*повертає попередній елемент*

*\**

*\*/*

*@Override*

***public*** *lab6.Ammunition previous() {*

***return*** *previous;*

*}*

*/\*\**

*\* повертає позицію попереднього елемента*

*\*/*

*@Override*

***public******int*** *previousIndex() {*

***return*** *index-1;*

*}*

*/\*\**

*\* видаляє теперішній елемент*

*\*/*

*@Override*

***public******void*** *remove() {*

*MyCollection.****this****.remove(index-1);*

*}*

*/\*\**

*\* встановлює на опзицію елемент е*

*\*/*

*@Override*

***public******void*** *set(Ammunition e) {*

*MyCollection.****this****.set(index-2, e);*

*}*

*}*

*/\*\**

*\* конструктор класу, створює пусту колекцію*

*\*/*

***public*** *MyCollection() {*

*Ammunition [] am =* ***new*** *Ammunition[15+ (****int****)(15\*0.3)];*

*ammunitions=am;*

*REALSIZE=0;*

*}*

*/\*\**

*\* конструктор класу*

*\* створює колекцію з одним елементом е*

*\** ***@param*** *e*

*\*/*

***public*** *MyCollection(Ammunition e) {*

*Ammunition [] am =* ***new*** *Ammunition[15+ (****int****)(15\*0.3)];*

*REALSIZE=0;*

*am[REALSIZE]=e;*

*REALSIZE=1;*

*ammunitions=am;*

*}*

*/\*\**

*\* конструктор класу,*

*\* створює колекцію з масиву елементів am*

*\** ***@param*** *am*

*\*/*

***public*** *MyCollection(Ammunition [] am) {*

***int*** *i=0;*

*Ammunition [] ambuf =* ***new*** *Ammunition [15+ (****int****)(15\*0.3)];*

***for*** *(Ammunition ammunition : am) {*

*ambuf[i]= ammunition;*

*i++;*

*}*

*REALSIZE=i;*

*ammunitions = ambuf;*

*ambuf=****null****;*

*}*

*/\*\**

*\* Конструктор класу, створює колекцію з колекції типу*

*\* ArrayList<Ammunition>*

*\** ***@param*** *arList*

*\*/*

***public*** *MyCollection (ArrayList<Ammunition> arList) {*

*Ammunition [] ambuf =* ***new*** *Ammunition [15+ (****int****)(15\*0.3)];*

***int*** *k=0;*

***for*** *(Ammunition ammunition : arList) {*

*ambuf[k]=ammunition;*

*k++;*

*}*

*REALSIZE = k;*

*ammunitions = ambuf;*

*}*

*/\*\**

*\* додає до колекції елемент e*

*\*/*

***public******boolean*** *add(Ammunition e) {*

***if*** *(REALSIZE>=0){*

*ammunitions[REALSIZE]= e;*

*REALSIZE++;*

***return******true****;*

*}*

***else***

***return******false****;*

*}*

*/\*\**

*\* додає до колекції елемент element*

*\* на позицію index*

*\*/*

*@Override*

***public******void*** *add(****int*** *index, Ammunition element) {*

***if*** *(REALSIZE <ammunitions.length){*

***int*** *i=index;*

*Ammunition buf2 ;*

*Ammunition buf1 = ammunitions[index];*

***while*** *(ammunitions[i]!=****null****) {*

*buf2=ammunitions[i+1];*

*ammunitions[i+1]=buf1;*

*buf1=ammunitions[i+2];*

*ammunitions[i+2]=buf2;*

*i++;*

*}*

*ammunitions[index]=element;*

*REALSIZE++;*

*}*

*}*

*/\*\**

*\* додає до колекції колекцію с*

*\*/*

*@Override*

***public******boolean*** *addAll(Collection<?* ***extends*** *Ammunition> c) {*

*Object [] am = c.toArray();*

***if*** *((ammunitions.length-REALSIZE)>=am.length){*

***for*** *(Object ammunition : am) {*

*ammunitions[REALSIZE]=(Ammunition)ammunition;*

*REALSIZE++;*

*}*

***return******true****;*

*}****else***

***return******false****;*

*}*

*/\*\**

*\* додає до колекції колекцію с*

*\* з позиції i*

*\*/*

*@Override*

***public******boolean*** *addAll(****int*** *index, Collection<?* ***extends*** *Ammunition> c) {*

*Object [] am = c.toArray();*

***if*** *((ammunitions.length-REALSIZE)>=am.length){*

***int*** *i=index;*

*Ammunition buf2 ;*

*Ammunition buf1 = ammunitions[index];*

***for*** *(****int*** *j = 0; j < am.length; j++) {*

***while*** *(ammunitions[i]!=****null****) {*

*buf2=ammunitions[i+1];*

*ammunitions[i+1]=buf1;*

*buf1=ammunitions[i+2];*

*ammunitions[i+2]=buf2;*

*i++;*

*}*

*}*

***for*** *(Object ammunition : am) {*

*ammunitions[index]=(Ammunition)ammunition;*

*index++;*

*REALSIZE++;*

*}*

***return******true****;*

*}*

***else***

***return******false****;*

*}*

*/\*\**

*\* видаляє всі елементи колекції*

*\*/*

*@Override*

***public******void*** *clear() {*

*Ammunition [] am =* ***new*** *Ammunition[15+ (****int****)(15\*0.3)];*

*ammunitions=am;*

*}*

*/\*\**

*\* перевіряє чи об'єкт e*

*\* належить колекції*

*\*/*

*@Override*

***public******boolean*** *contains(Object e ) {*

***boolean*** *q=* ***false****;*

***for*** *(Ammunition ammunition : ammunitions) {*

***if*** *(ammunition.equals(e)){*

*q=****true****;*

*}*

*}*

***return*** *q;*

*}*

*/\*\**

*\* перевіряє чи колекція с належить колекції*

*\*/*

*@Override*

***public******boolean*** *containsAll(Collection<?> c) {*

***boolean*** *q =* ***true****;*

***int*** *i=0;*

*Object [] amm = c.toArray();*

***for****(****int*** *j = 0; j<amm.length; j++){*

*i=0;q=****true****;*

***while*** *((ammunitions[i]!=****null****)&(q)) {*

***if*** *(ammunitions[i].equals(amm[j])){*

*q=****false****;*

*i++;*

*amm[j]=****null****;*

*}*

*}*

*}*

*q=****true****;*

***for*** *(Object ammunition : amm) {*

***if*** *(ammunition==****null****){*

*q=****true****;*

*}*

***else*** *q=****false****;*

*}*

***return*** *q;*

*}*

*/\*\**

*\* повертає елемент колекції за позицією index*

*\*/*

*@Override*

***public*** *Ammunition get(****int*** *index) {*

***return*** *ammunitions[index];*

*}*

*/\*\**

*\* повертає позицію першого об'єкта е*

*\* який зустрівся в колекції*

*\*/*

*@Override*

***public******int*** *indexOf(Object e) {*

***boolean*** *q=****true****;*

***int*** *positon = 0;*

*ListIterator<Ammunition> itr = listIterator();*

***if*** *(q)*

***while*** *((itr.hasNext())) {*

*Ammunition ammunition = itr.next();*

***if*** *(e.equals(ammunition)){*

*q=****false****;*

*positon = itr.nextIndex()-2;*

*}*

*}*

***if*** *(q){*

***return*** *-1;*

*}*

***else******return*** *positon;*

*}*

*/\*\**

*\* преевіряє чи колекція пуста*

*\*/*

*@Override*

***public******boolean*** *isEmpty() {*

***boolean*** *q=* ***true****;*

***for*** *(Ammunition ammunition : ammunitions) {*

***if*** *(!(ammunition==****null****)){*

*q=****false****;;*

*}*

*}*

***return*** *q;*

*}*

*/\*\**

*\* повертає ітератор*

*\*/*

*@Override*

***public*** *Iterator<Ammunition> iterator(){*

*MyListIterator itr =* ***new*** *MyListIterator();*

***return*** *itr;*

*}*

*/\*\**

*\* повертає останнє входження*

*\* об'єкта е в колекцію*

*\*/*

*@Override*

***public******int*** *lastIndexOf(Object e) {*

***boolean*** *q=****true****;*

***int*** *i=REALSIZE;*

***while*** *((q)&(ammunitions[i]!=****null****)) {*

***if****(ammunitions[i].equals(e)){*

*q=****false****;*

*}*

***else*** *i--;*

*}*

***if*** *(i==0)* ***return*** *-1;*

***else******return*** *i-1;*

*}*

*/\*\**

*\* повертає ListIterator для колекції*

*\*/*

*@Override*

***public*** *ListIterator<Ammunition> listIterator() {*

*MyListIterator itr =* ***new*** *MyListIterator();*

***return*** *itr;*

*}*

*/\*\**

*\* повертає ListIterator для колекції з позиції index*

*\*/*

*@Override*

***public*** *ListIterator<Ammunition> listIterator(****int*** *index) {*

*MyListIterator itr =* ***new*** *MyListIterator(index);*

***return*** *itr;*

*}*

*/\*\**

*\* видаляє обєкт e з колекції*

*\*/*

*@Override*

***public******boolean*** *remove(Object e) {*

***int*** *i=0;*

***if*** *(indexOf(e)!=-1){*

*Ammunition [] am =* ***new*** *Ammunition[15+ (****int****)(15\*0.3)];*

*ListIterator<Ammunition> itr = listIterator();*

***while*** *(itr.hasNext()) {*

*Ammunition ammunition = itr.next();*

***if*** *(!ammunition.equals(e)){*

*am[i]=ammunition;*

*i++;*

*}*

*}*

*ammunitions=****null****;*

*ammunitions= am;*

*REALSIZE--;*

***return******true****;*

*}*

***else******return******false****;*

*}*

*/\*\**

*\* видаляє об'єкт за позицією index*

*\*/*

*@Override*

***public*** *Ammunition remove(****int*** *index) {*

*Ammunition [] newam =* ***new*** *Ammunition[ammunitions.length];*

***int*** *k=0;*

***for*** *(****int*** *i = 0; i < ammunitions.length; i++) {*

***if****(i!=index){*

*newam[k]= ammunitions[i];*

*k++;*

*}*

*}*

*Ammunition buf = ammunitions[index];*

*ammunitions=****null****;*

*ammunitions = newam;*

*REALSIZE--;*

***return*** *buf;*

*}*

*/\*\**

*\* видаляє з колекції всі обєкти колекції с*

*\*/*

*@Override*

***public******boolean*** *removeAll(Collection<?> c) {*

*Object [] am = c.toArray();*

*ListIterator<Ammunition> itr1 = listIterator();*

***while*** *(itr1.hasNext()) {*

*Ammunition ammunition = itr1.next();*

***for*** *(Object object : am) {*

***if*** *(ammunition.equals(object)){*

*remove(ammunition);*

*}*

*}*

*}*

***return******true****;*

*}*

*/\*\**

*\* видаляє з колекції всі об'єкти що не належить до*

*\* колекції c*

*\*/*

*@Override*

***public******boolean*** *retainAll(Collection<?> c) {*

*Object [] am = c.toArray();*

*Ammunition [] newam =* ***new*** *Ammunition [ammunitions.length];*

***int*** *k=0;*

***for*** *(****int*** *i = 0; i < am.length; i++) {*

***if*** *(indexOf(am[i])!=-1){*

*newam[k]=(Ammunition)am[i];*

*k++;*

*}*

*}*

***if*** *(newam[0]!=****null****){*

*ammunitions=****null****;*

*ammunitions=newam;*

***return******true****;*

*}*

***else***

***return******false****;*

*}*

*/\*\**

*\* встановлює елемент element*

*\* на позицію index*

*\*/*

*@Override*

***public*** *Ammunition set(****int*** *index, Ammunition element) {*

***if*** *((index>0)&(index<ammunitions.length))*

*ammunitions[index]=element;*

***return*** *element;*

*}*

*/\*\**

*\* повертає кількість елементів колекції*

*\*/*

*@Override*

***public******int*** *size() {*

***return*** *REALSIZE;*

*}*

*/\*\**

*\* повертає список ArrayList*

*\* колекції з позиції fromIndex до позиції toIndex*

*\*/*

*@Override*

***public*** *ArrayList<Ammunition> subList(****int*** *fromIndex,* ***int*** *toIndex) {*

*ArrayList<Ammunition> am =* ***new*** *ArrayList<Ammunition>();*

***for*** *(****int*** *i = fromIndex; i < toIndex; i++) {*

*am.add(ammunitions[i]);*

*}*

***return*** *am;*

*}*

*/\*\**

*\* повертає масив з елементів колекції*

*\*/*

*@Override*

***public*** *Ammunition [] toArray() {*

***int*** *i=0;*

*Ammunition [] am =* ***new*** *Ammunition [size()];*

*ListIterator<Ammunition> itr = listIterator();*

***while*** *(itr.hasNext()) {*

*Ammunition ammunition = itr.next();*

*am[i]=ammunition;*

*i++;*

*}*

***return*** *am;*

*}*

*/\*\**

*\* повертає масив з елементів колекції з елементами*

*\* клас яких співпав з елементами масиву array*

*\*/*

*@SuppressWarnings({ "hiding", "unchecked", "null" })*

*@Override*

***public*** *<Ammunition> Ammunition[] toArray(Ammunition[] array) {*

***int*** *reconciliation\_counter = 0;*

***for*** *(lab6.Ammunition ammunition : ammunitions) {*

***for*** *(Ammunition ams : array) {*

***if*** *(ammunition.getClass().equals(ams.getClass())){*

*reconciliation\_counter++;*

*}*

*}*

*}*

***if*** *(reconciliation\_counter==array.length){*

***return*** *array;*

*}*

***else****{*

*Ammunition [] newam =* ***null****;*

***int*** *k=0;*

***for*** *(lab6.Ammunition ammunition : ammunitions) {*

***for*** *(Ammunition ams : array) {*

***if*** *(ammunition.getClass().equals(ams.getClass())){*

*newam[k]=(Ammunition)ammunition;*

*}*

*}*

*}*

***return*** *newam;*

*}*

*}*

*/\*\**

*\* перевіряє чи підходить даний набір амуніції для*

*\* одного лицаря*

*\*/*

***public******boolean*** *isCorrectAmmunition() {*

***boolean*** *bul =* ***true****;*

***int*** *i=0;*

*ListIterator<Ammunition> itr = listIterator();*

***while*** *(itr.hasNext()) {*

*Ammunition ammunition = itr.next();*

*ListIterator<Ammunition> itr1 = listIterator();*

*i=0;*

***while*** *(itr1.hasNext()) {*

*Ammunition ammunition2 = itr1.next();*

***if*** *((ammunition.getClass()).equals(ammunition2.getClass())){*

*i++;*

*}*

*}*

***if*** *(i>=2){*

*bul=****false****;*

*}*

*}*

***return*** *bul;*

*}*

*}*

Клас Ammuniton

***package*** *lab6;*

*/\*\**

*\* Батьківський клас для Амуніції*

*\** ***@author*** *Volodia Kuzmenko*

*\**

*\*/*

***public******class*** *Ammunition {*

*/\*\*ціна \*/*

***private******int*** *price;*

*/\*\*вага\*/*

***private******double*** *weight;*

*/\*\*назва\*/*

***private*** *String name\_am;*

*/\*\**

*\**

*\** ***@param*** *name назва амуніції*

*\** ***@param*** *p ціна амуніції*

*\** ***@param*** *w вага амуніції*

*\*/*

***public*** *Ammunition (String name,* ***int*** *p ,* ***double*** *w) {*

*name\_am=name;*

*price=p;*

*weight=w;};*

*/\*\**

*\**

*\** ***@return*** *ціну*

*\*/*

***public******int*** *getPrice(){*

***return*** *price;*

*}*

*/\*\**

*\**

*\** ***@return*** *вагу*

*\*/*

***public******double*** *getWeight(){*

***return*** *weight;*

*}*

*/\*\**

*\**

*\** ***@return*** *назву амуніції*

*\*/*

***public*** *String getName() {*

***return*** *name\_am;*

*}*

*/\*\**

*\**

*\** ***@return*** *інформацію про предмет амуніції*

*\*/*

***public*** *String getInformation(){*

***return*** *"Name:" +name\_am+" Price:"+String.valueOf(price)+" Weight:"+String.valueOf(weight);*

*}*

*}*

Клас Armor

***package*** *lab6;*

*/\*\**

*\* лас-Нащадок від класу Ammunition*

*\* описує об'єкт Кольчуга*

*\** ***@author*** *Volodia Kuzmenko*

*\**

*\*/*

***public******class*** *Armor* ***extends*** *Ammunition {*

*/\*\**

*\**

*\** ***@param*** *n назва кольчуги*

*\** ***@param*** *p ціна кольчуги*

*\** ***@param*** *w вага кольчуги*

*\*/*

***public*** *Armor(String n,* ***int*** *p,* ***double*** *w) {*

***super****(n, p, w);*

*}*

*/\*\*тип кольчуга\*/*

***private*** *String name = "Кольчуга";*

*/\*\**

*\* @ return інформацю*

*\*/*

***public*** *String getInformation(){*

***return*** *"Тип "+ name + " "+****super****.getInformation();*

*}*

*}*

Клас Shield

***package*** *lab6;*

*/\*\**

*\* лас-Нащадок від класу Ammunition*

*\* описує об'єкт Щит*

*\** ***@author*** *Volodia Kuzmenko*

*\**

*\*/*

***public******class*** *Shield* ***extends*** *Ammunition {*

*/\*\**

*\**

*\** ***@param*** *n назва щита*

*\** ***@param*** *p ціна щита*

*\** ***@param*** *w вага щита*

*\*/*

***public*** *Shield(String n,* ***int*** *p,* ***double*** *w) {*

***super****(n, p, w);*

*}*

*/\*\**

*\**

*\** ***@param*** *тип амуніції - Щит*

*\*/*

***private*** *String name = "Щит";*

*/\*\**

*\* @ return інформацю*

*\*/*

***public*** *String getInformation(){*

***return*** *"Тип "+ name + " "+****super****.getInformation();*

*}*

*}*

Клас Sword

***package*** *lab6;*

*/\*\**

*\* лас-Нащадок від класу Ammunition*

*\* описує об'єкт Меч*

*\** ***@author*** *Volodia Kuzmenko*

*\**

*\*/*

***public******class*** *Sword* ***extends*** *Ammunition{*

*/\*\**

*\**

*\** ***@param*** *n назва меча*

*\** ***@param*** *p ціна меча*

*\** ***@param*** *w вага меча*

*\*/*

***public*** *Sword(String n,* ***int*** *p,* ***double*** *w) {*

***super****(n, p, w);*

*}*

*/\*\*тип Меч\*/*

***private*** *String name = "Меч";*

*/\*\**

*\* @ return інформацю*

*\*/*

***public*** *String getInformation(){*

***return*** *"Тип "+ name + " "+****super****.getInformation();*

*}*

*}*

Клас Helmet

***package*** *lab6;*

*/\*\**

*\* Клас-Нащадок від класу Ammunition*

*\* описує об'єкт Шолом*

*\** ***@author*** *Volodia Kuzmenko*

*\**

*\*/*

***public******class*** *Helmet* ***extends*** *Ammunition {*

*/\*\**

*\**

*\** ***@param*** *n назва шолома*

*\** ***@param*** *p ціна шолома*

*\** ***@param*** *w вага шолома*

*\*/*

***public*** *Helmet(String n,* ***int*** *p,* ***double*** *w) {*

***super****(n, p, w);*

*}*

*/\*\*Тип шолом \*/*

***private*** *String name = "Шолом";*

*/\*\**

*\* @ return інформацю*

*\*/*

***public*** *String getInformation(){*

***return*** *"Тип "+ name + " "+****super****.getInformation();*

*}*

*}*

Висновки: Я створив клас MyCollection що описує типізовану колекцію, внутрішньою структурою якої є масив. Клас реалізує інтерфейс List. Я створив 4 конструктори - пустий, з одним параметром типу Ammunition, з одним параметром - масивом типу Ammunition та з параметром - колекцією ArrayList.

Для реалізації ітератора я створив внутрішній клас MyListIterator який реалізовує інтерфейс ListIterator.