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 №5

Наслідування і поліморфізм

Виконав: студент 1 курсу

група: ІО-21

Кузьменко В. З.

Номер залікової книжки № 2114

Прийняв: Невдащенко М.В.

Дата:

Оцінка:

Київ - 2013

Тема: Наслідування і поліморфізм.

Мета: Ознайомлення з механізмом наслідування та принципом поліморфізму. Використанні механізму наслідування ти принципу поліморфізму в мові Java. Здобуття навичок у використанні механізму наслідування та принципу поліморфізму.

Варіант завдання: 6

Визначити ієрархію амуніції лицаря. Екіпірувати лицаря. Порахувати вартість амуніції. Провести сортування амуніції за вагою. Знайти елементи амуніції, що відповідають заданому діапазону цін.

Опис програми:

Створено клас Ammunition та 4 класи-нащадки: Sword, Armor, Helmet, Shield. У класі Chevalier знаходяться методи для знаходження ціни амуніції та сортування за вагою(сортування методом бульбашки). У виконавчому методі реалізовано Екіпірування лицаря та знаходження елементів амуніції, що відповідають заданому діапазону цін.

Програмний код:

**import** java.util.ArrayList;

/\*\*

\*

\* **@author** Volodia Kuzmenko

\*

\*/

**public** **class** Lab5 {

**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

\*/

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

am.add(helmet);

am.add(shield);

am.add(sword);

am.add(armor);

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

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

}

/\*\*

\* перевіряємо чи набрана амуніція підходить для

\* одного лицаря

\*/

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

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");

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

System.out.println(( chevalier.getAmmunition().get(i))

.getInformation());

}

}

/\*\*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\*/

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

**if** ((am.get(i).getPrice() > minPrice)

&& (am.get(i).getPrice() < maxPrice)) {

System.out.println(am.get(i).getInformation());

}

}

}

/\*\*

\*

\* **@param** am масив амуніції для лицаря

\* **@return** boolean чи підходить амуніція для 1 лицаря

\*/

**public** **static** **boolean** isCorrectAmmunition(ArrayList<Ammunition> am) {

**int** q = 0;

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

Ammunition buf;

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

q = 0;

buf = ammunition;

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

**if** ((buf.getClass()).equals(amOne.getClass()))

q++;

}

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

bul = **false**;

**break**;

}

}

**return** bul;

}

}

**import** java.util.ArrayList;

/\*\*

\*

\* **@author** Volodia Kuzmenko

\*

\*/

**public** **class** Chevalier {

**private** ArrayList<Ammunition> ammunition;

/\*\*

\*

\* **@param** am амуніція для одного лицаря

\*/

**public** Chevalier(ArrayList<Ammunition> am) {

ammunition = am;

}

/\*\*

\*

\* **@return** ціну всієї амуніції для лицаря

\*/

**public** **int** getPriceAmmunition(){

**int** price=0;

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

price += ammun.getPrice();

}

**return** price;

}

/\*\*

\*

\* **@param** arr масив амуніції

\* **@param** i позиція для заміни

\* **@param** j позиція для заміни

\*/

**private** **void** swap(ArrayList<Ammunition> arr, **int** i, **int** j) {

Ammunition t = arr.get(i);

arr.set(i, arr.get(j));

arr.set(j, t);

}

/\*\*

\* метод сортує амуніцію по вазі

\* **@param** arr масив амуніції

\*

\*/

**public** **void** sortAmmunition(ArrayList<Ammunition> arr){

**for**(**int** i = arr.size()-1 ; i > 0 ; i--){

**for**(**int** j = 0 ; j < i ; j++){

**if**( arr.get(j).getWeight() > arr.get(j+1).getWeight())

swap(arr, j, j+1);

}

}

}

/\*\*

\*

\* **@return**

\*/

**public** ArrayList<Ammunition> getAmmunition(){

**return** ammunition;

}

}

/\*\*

\*

\* **@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);

}

}

/\*\*

\*

\* **@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();

}

}

/\*\*

\*

\* **@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();

}

}

/\*\*

\*

\* **@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();

}

}

/\*\*

\*

\* **@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();

}

}

Висновок: Я використав механізм наслідування при створенні ієрархії амуніції лицаря. Використав принцип поліморфізму при виведенні інформації про предмет амуніції лицаря. Здобув навички у використанні механізму наслідування та принципу поліморфізму.