Model

=========

IEnt.java

package model;

public interface IEnt {

int GetWeight();

}

Apple.java

package model;

public class Apple implements IEnt {

int density, volume;

public Apple(int density, int volume) {

this.density = density;

this.volume = volume;

}

@Override

public int GetWeight() { return this.density \* this.weight; }

public int GetDensity() { return this.density; }

public void SetDensity(int density) {

this.density = density;

}

@Override

public String toString() {

return “Apple with density:”+ this.density + ” with volume:” + this.volume;

}  
}

Repository

=========

IRepository.java

package Repository;

import Model.\*;

public interface IRepository{

public void addElement(IEnt elementToAdd) throws MyExc;

public IEnt[] getElements();

public IEnt getElementFromPosition(int position) throws MyExc;

public Void updateElement(int position, IEnt newElement) throws MyExc;

}

ArrayRepository.java

package Repositorya

import Model.IEnt

public class ArrayRepository implements IRepository {

IEnt[] things;

int size;

public ArrayRepository(int maxCapacity) {

things = new IEnt[maxCapacity];

size = 0;

}

@Override

public void addElement(IEnt elementToAdd) throws MyExc {

If (size == things.length)

throw MyExc(“Limit of array reached!”);

things[size++] = elementToAdd;

}

@Override

public IEnt[] getElements() {

return things;

}

@Override

public IEnt getElementFromPosition(int position) throws MyExc {

if (position < 0 || positon >= size)

throws MyExc(“Index specified doesn’t exist!”);

return things[position];

}

@Override

public void updateElement(int position, IEnt newElement) throws MyExc {

if (position < 0 || positon >= size)

throws MyExc(“Index specified doesn’t exist!”);

things[position] = newElement;

}

}

Controller

========

Controller.java

package Controller;

import Repository.\*

public class Controller {

IRepository repository;

public Controller(IRepository repository) {

this.repository = repository;

}

public void addElement(IEnt element) throws MyExc {

this.repository.addElement(element);

}

public String[] filterItems(int minimumWeight) {

String[] s = new String[this.repository.getElements().length];

int size = 0;

for (IEnt item : this.repository.getElements()) {

If (item == null)

break;

if (item.getWeight() >= minimumWeight)

s[size++] = item.toString();

}

String[] newArray = new String[size];

System.arraycopy(s, 0, newArray, 0, size);

return newArray;

}

/\*public String[] filterItems(int minimumWeight) {

var s = new String[this.repository.getElements().length];

var size = 0;

for (var item : this.repository.getElements()) {

If(item == null)

break;

if (item.getWeight() >= minimumWeight)

s[size++] = item.toString();

var newArray = new String[size];

System.arraycopy(s, 0, newArray, 0, size);

return newArray;

}\*/

}

View

========

//AppView.java

package view;

public class AppView {

public static void main(String[] args) {

IRepository repository = new ArrayRepository(15);

Controller controller = new Controller(repository);

Apple a1 = new Apple(2, 3);

Book b1 = new Book(...);

Cake c1 = new Cake(…);

try {

controller.addElement(a1);

controller.addElement(b1);

controller.addElement(c1);

} catch (MyExc exception) {

System.out.println(exception);

}

String[] array = controller.filterItems(200);

for (String el : array) {

System.out.println(el);

}

}

}