1.INVENTORY MANAGEMENT SYSTEM

**package** inventory;

**import** java.util.\*;

**public** **class** inventory {

**static** **class** Product {

String id;

String name;

**int** qty;

**double** price;

Product(String id, String name, **int** qty, **double** price) {

**this**.id = id;

**this**.name = name;

**this**.qty = qty;

**this**.price = price;

}

**public** String toString() {

**return** name + " " + qty + " " + price;

}

}

**static** **class** Inventory {

HashMap<String, Product> map = **new** HashMap<>();

**void** add(Product p) {

map.put(p.id, p);

}

**void** update(String id, **int** qty, **double** price) {

**if** (map.containsKey(id)) {

Product p = map.get(id);

p.qty = qty;

p.price = price;

}

}

**void** delete(String id) {

map.remove(id);

}

**void** show() {

**for** (Product p : map.values()) {

System.***out***.println(p);

}

}

}

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

Inventory inv = **new** Inventory();

inv.add(**new** Product("p1", "Mouse", 10, 299));

inv.add(**new** Product("p2", "Keyboard", 5, 499));

inv.show();

inv.update("p1", 15, 279);

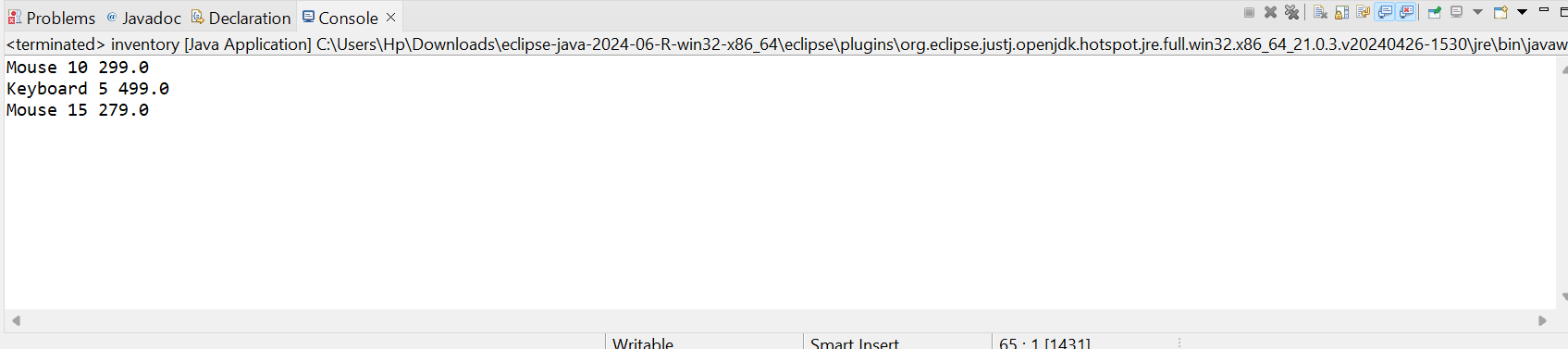
inv.delete("p2");

inv.show();

}

}

OUTPUT:



2.SEARCHING

**package** searching;

**import** java.util.\*;

**public** **class** searching {

**static** **class** P {

String id, name, cat;

P(String id, String name, String cat) {

**this**.id = id;

**this**.name = name;

**this**.cat = cat;

}

}

**static** **int** linear(P[] arr, String name) {

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

**if** (arr[i].name.equalsIgnoreCase(name)) **return** i;

}

**return** -1;

}

**static** **int** binary(P[] arr, String name) {

**int** l = 0, r = arr.length - 1;

**while** (l <= r) {

**int** m = (l + r) / 2;

**int** cmp = arr[m].name.compareToIgnoreCase(name);

**if** (cmp == 0) **return** m;

**if** (cmp < 0) l = m + 1;

**else** r = m - 1;

}

**return** -1;

}

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

P[] list = {

**new** P("1", "Bag", "Acc"),

**new** P("2", "Phone", "Elec"),

**new** P("3", "Shoe", "Foot")

};

**int** f1 = *linear*(list, "Shoe");

System.***out***.println(f1 >= 0 ? "Found" : "Not Found");

Arrays.*sort*(list, Comparator.*comparing*(p -> p.name));

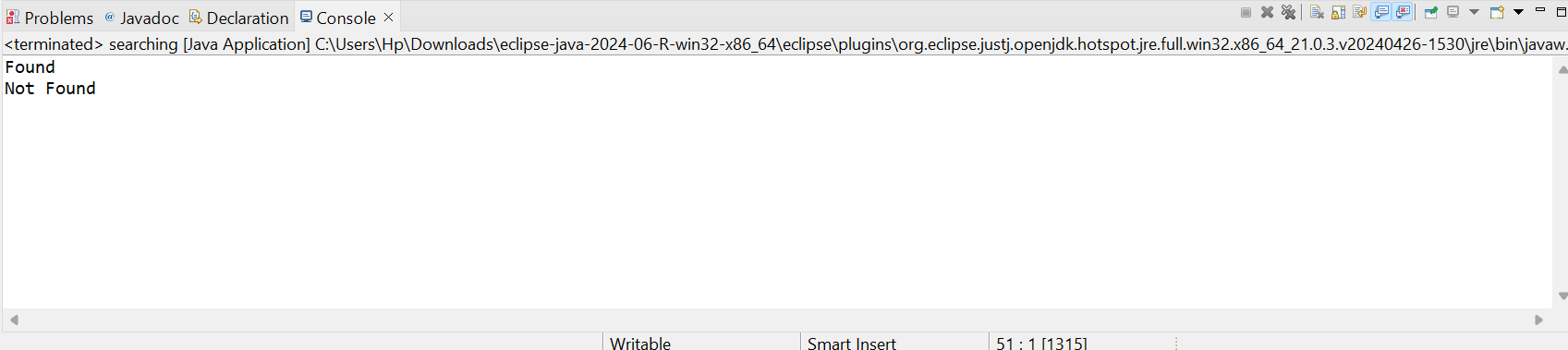
**int** f2 = *binary*(list, "Pen");

System.***out***.println(f2 >= 0 ? "Found" : "Not Found");

}

}

OUTPUT:



3.SORTING

**package** sorting;

**public** **class** sorting {

**static** **class** O {

String id, name;

**double** price;

O(String id, String name, **double** price) {

**this**.id = id;

**this**.name = name;

**this**.price = price;

}

**public** String toString() {

**return** name + " " + price;

}

}

**static** **void** bubble(O[] a) {

**for** (**int** i = 0; i < a.length - 1; i++) {

**for** (**int** j = 0; j < a.length - i - 1; j++) {

**if** (a[j].price > a[j + 1].price) {

O t = a[j];

a[j] = a[j + 1];

a[j + 1] = t;

}

}

}

}

**static** **void** quick(O[] a, **int** l, **int** h) {

**if** (l < h) {

**int** p = *part*(a, l, h);

*quick*(a, l, p - 1);

*quick*(a, p + 1, h);

}

}

**static** **int** part(O[] a, **int** l, **int** h) {

**double** key = a[h].price;

**int** i = l - 1;

**for** (**int** j = l; j < h; j++) {

**if** (a[j].price <= key) {

i++;

O t = a[i];

a[i] = a[j];

a[j] = t;

}

}

O t = a[i + 1];

a[i + 1] = a[h];

a[h] = t;

**return** i + 1;

}

**static** **void** show(O[] a) {

**for** (O o : a) {

System.***out***.println(o);

}

}

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

O[] a = {

**new** O("1", "Asha", 400),

**new** O("2", "Bala", 150),

**new** O("3", "Chitra", 300)

};

System.***out***.println("Bubble sort:");

O[] x = a.clone();

*bubble*(x);

*show*(x);

System.***out***.println("Ouick sort:");

O[] y = a.clone();

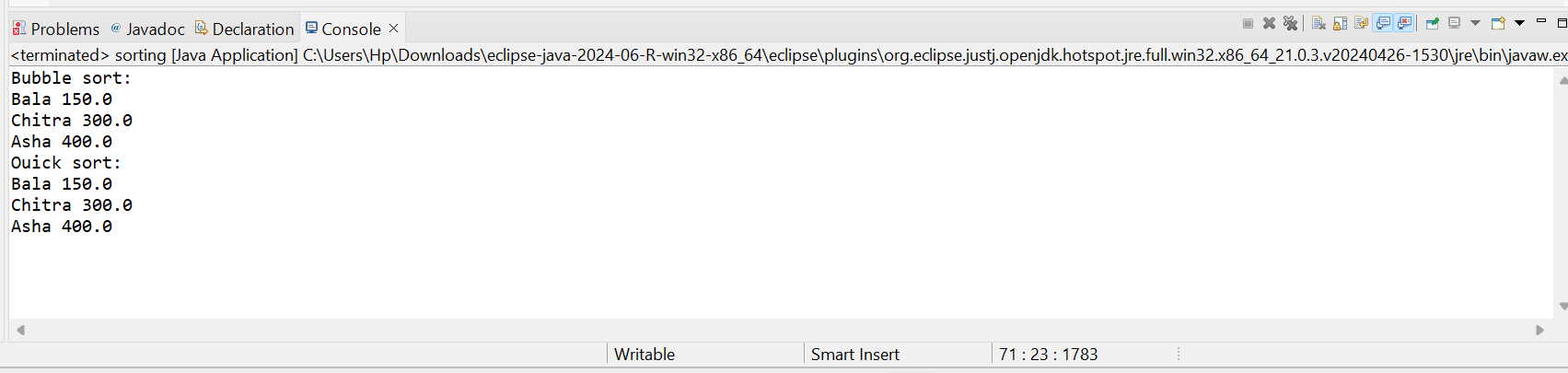
*quick*(y, 0, y.length - 1);

*show*(y);

}

}

OUTPUT:



5.DIFFERENT OPERATION

**package** operation;

**public** **class** operation {

**static** **class** Emp {

String id, name, pos;

**double** sal;

Emp(String id, String name, String pos, **double** sal) {

**this**.id = id;

**this**.name = name;

**this**.pos = pos;

**this**.sal = sal;

}

**public** String toString() {

**return** name + " " + pos + " " + sal;

}

}

**static** **class** EmpStore {

Emp[] arr = **new** Emp[10];

**int** count = 0;

**void** add(Emp e) {

**if** (count < arr.length) {

arr[count++] = e;

}

}

**void** show() {

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

System.***out***.println(arr[i]);

}

}

**void** search(String id) {

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

**if** (arr[i].id.equals(id)) {

System.***out***.println(arr[i]);

**return**;

}

}

System.***out***.println("Not found");

}

**void** delete(String id) {

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

**if** (arr[i].id.equals(id)) {

**for** (**int** j = i; j < count - 1; j++) {

arr[j] = arr[j + 1];

}

arr[--count] = **null**;

**return**;

}

}

}

}

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

EmpStore s = **new** EmpStore();

s.add(**new** Emp("E1", "Ravi", "Manager", 50000));

s.add(**new** Emp("E2", "Kiran", "Clerk", 25000));

s.add(**new** Emp("E3", "Geeta", "HR", 30000));

s.show();

System.***out***.println("Search:");

s.search("E2");

System.***out***.println("Delete:");

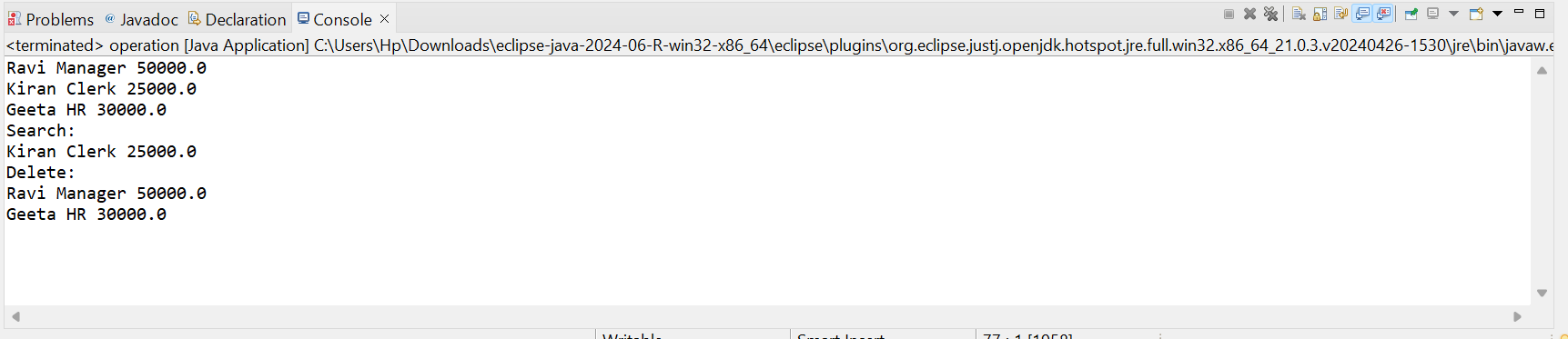
s.delete("E2");

s.show();

}

}

OUTPUT:



5.TASK LIST

**package** list;

**public** **class** tasklist {

**static** **class** Task {

String id, name, status;

Task next;

Task(String id, String name, String status) {

**this**.id = id;

**this**.name = name;

**this**.status = status;

}

**public** String toString() {

**return** name + " " + status;

}

}

**static** **class** TaskList {

Task head;

**void** add(Task t) {

**if** (head == **null**) head = t;

**else** {

Task cur = head;

**while** (cur.next != **null**) cur = cur.next;

cur.next = t;

}

}

**void** show() {

Task cur = head;

**while** (cur != **null**) {

System.***out***.println(cur);

cur = cur.next;

}

}

**void** find(String id) {

Task cur = head;

**while** (cur != **null**) {

**if** (cur.id.equals(id)) {

System.***out***.println(cur);

**return**;

}

cur = cur.next;

}

System.***out***.println("Not found");

}

**void** del(String id) {

**if** (head == **null**) **return**;

**if** (head.id.equals(id)) {

head = head.next;

**return**;

}

Task cur = head;

**while** (cur.next != **null**) {

**if** (cur.next.id.equals(id)) {

cur.next = cur.next.next;

**return**;

}

cur = cur.next;

}

}

}

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

{

TaskList t = **new** TaskList();

t.add(**new** Task("1", "Design", "ToDo"));

t.add(**new** Task("2", "Code", "Done"));

t.add(**new** Task("3", "Test", "ToDo"));

t.show();

System.***out***.println("Find:");

t.find("2");

System.***out***.println("Delete:");

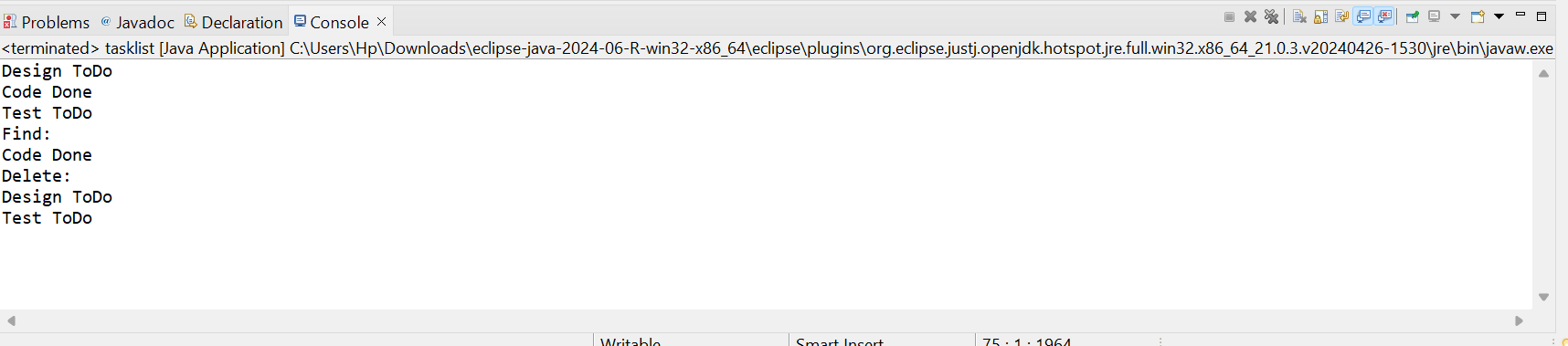
t.del("2");

t.show();

}

}

OUTPUT:



6.LINEAR AND BINARY SEARCHING

**package** linearbinary;

**import** java.util.\*;

**public** **class** linearbinary {

**static** **class** B {

String id, title, author;

B(String id, String title, String author) {

**this**.id = id;

**this**.title = title;

**this**.author = author;

}

**public** String toString() {

**return** title + " by " + author;

}

}

**static** **int** lin(B[] a, String key) {

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

**if** (a[i].title.equalsIgnoreCase(key)) **return** i;

}

**return** -1;

}

**static** **int** bin(B[] a, String key) {

**int** l = 0, r = a.length - 1;

**while** (l <= r) {

**int** m = (l + r) / 2;

**int** cmp = a[m].title.compareToIgnoreCase(key);

**if** (cmp == 0) **return** m;

**if** (cmp < 0) l = m + 1;

**else** r = m - 1;

}

**return** -1;

}

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

B[] list = {

**new** B("1", "Java", "zaara"),

**new** B("2", "DS", "vicky"),

**new** B("3", "Web", "gopinath")

};

**int** i = *lin*(list, "Web");

System.***out***.println(i >= 0 ? list[i] : "Not found");

Arrays.*sort*(list, Comparator.*comparing*(b -> b.title));

**int** j = *bin*(list, "DS");

System.***out***.println(j >= 0 ? list[j] : "Not found");

}

}

OUTPUT:



7.RECURSIVE

**package** recursive;

**public** **class** recursive {

**static** **double** getFuture(**double** val, **double** rate, **int** time) {

**if** (time == 0) {

**return** val;

} **else** {

**double** prev = *getFuture*(val, rate, time - 1);

**return** prev \* (1 + rate);

}

}

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

**double** start = 1000;

**double** grow = 0.10;

**int** year = 3;

**double** ans = *getFuture*(start, grow, year);

System.***out***.println("Future value is " + ans);

}

}

OUTPUT:

