package Amar;

public class Sample {

//using priority queue

public static void main(String[] args) {

ToDoList todoList = new ToDoList();

todoList.addItem("Complete project proposal", 2);

todoList.addItem("Buy groceries", 1);

todoList.addItem("Call client", 3);

System.*out*.println("Initial List:");

todoList.displayItems();

System.*out*.println("\nRemoving the highest priority item:");

todoList.removeHighestPriorityItem();

todoList.displayItems();

System.*out*.println("\nUpdating an item's description:");

todoList.updateItemDescription(1, "Buy essential groceries");

todoList.displayItems();

}

}

package Amar;

public class ToDoItem {

private String description;

private int priority;

public ToDoItem(String description, int priority) {

this.description = description;

this.priority = priority;

}

public String getDescription() {

return description;

}

public void setDescription(String description) {

this.description = description;

}

public int getPriority() {

return priority;

}

public void setPriority(int priority) {

this.priority = priority;

}

}

package Amar;

public class ToDoList {

private static final int ***MAX\_SIZE*** = 10;

private ToDoItem[] items;

private int size;

public ToDoList() {

items = new ToDoItem[***MAX\_SIZE***];

size = 0;

}

public void addItem(String description, int priority) {

ToDoItem newItem = new ToDoItem(description, priority);

if (size == 0) {

items[size] = newItem;

} else {

int i;

for (i = size - 1; i >= 0; i--) {

if (items[i].getPriority() < priority) {

items[i + 1] = items[i];

} else {

break;

}

}

items[i + 1] = newItem;

}

size++;

}

public void removeHighestPriorityItem() {

if (size > 0) {

for (int i = 0; i < size - 1; i++) {

items[i] = items[i + 1];

}

size--;

} else {

System.***out***.println("List is empty. No items to remove.");

}

}

public void updateItemDescription(int index, String newDescription) {

if (index >= 0 && index < size) {

items[index].setDescription(newDescription);

} else {

System.***out***.println("Invalid index.");

}

}

public void displayItems() {

if (size == 0) {

System.***out***.println("List is empty.");

} else {

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

System.***out***.println((i + 1) + ". Priority: " + items[i].getPriority() + ", Description: " + items[i].getDescription());

}

}

}

}