**import** java.text.DecimalFormat;

**import** java.text.ParseException;

**import** java.text.SimpleDateFormat;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.Comparator;

**import** java.util.Date;

**import** java.util.List;

**import** java.util.Scanner;

**public** **class** Program4 {

**public** **static** **void** main(String[] args) **throws** NumberFormatException, ParseException {

ArrayList<Note> list=**new** ArrayList<Note>();

List<Note> outlist=**new** ArrayList<Note>();

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("Enter the No of the book:");

**int** no=Integer.*parseInt*(input.nextLine());

**for** (**int** i = 0; i < no; i++) {

String detail=input.nextLine();

Note note=**new** Note().*CreateNote*(detail);

list.add(note);

}

System.***out***.println("Enter a type to sort:\r\n" +

"1.Sort by name\r\n" +

"2.Sort by priority level\r\n" +

"3.Sort by date created");

**int** choice=Integer.*parseInt*(input.nextLine());

**while**(choice>0&&choice<4) {

**if**(choice==1) {

Collections.*sort*(list);

}

**else** **if**(choice==2) {

PriorityLevelComparer pl=**new** PriorityLevelComparer();

Collections.*sort*(list, pl);

}

**else** **if**(choice==3) {

DateComparer dc=**new** DateComparer();

Collections.*sort*(list, dc);

}

System.***out***.println("Name\tContent\tSize\tCreated\tDate\tPriorityLevel");

**for** (Note note : list) {

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

}

System.***out***.println("1.By Name\n2.By Created Date\n3.By Priority Level\nEnter your choice:");

choice=Integer.*parseInt*(input.nextLine());

}

}

}

**class** Note **implements** Comparable<Note>

{

**private** String \_name;

**private** String \_content;

**private** Double \_size;

**private** Double \_priorityLevel;

**private** Date \_createdDate;

**public** Note(String \_name, String \_content, Double \_size, Double \_priorityLevel,Date \_createdDate) {

**super**();

**this**.\_name = \_name;

**this**.\_content = \_content;

**this**.\_size = \_size;

**this**.\_priorityLevel = \_priorityLevel;

**this**.\_createdDate = \_createdDate;

}

Note(){}

**public** String get\_name() {

**return** \_name;

}

**public** **void** set\_name(String \_name) {

**this**.\_name = \_name;

}

**public** String get\_content() {

**return** \_content;

}

**public** **void** set\_content(String \_content) {

**this**.\_content = \_content;

}

**public** Double get\_size() {

**return** \_size;

}

**public** **void** set\_size(Double \_size) {

**this**.\_size = \_size;

}

**public** Double get\_priorityLevel() {

**return** \_priorityLevel;

}

**public** **void** set\_priorityLevel(Double \_priorityLevel) {

**this**.\_priorityLevel = \_priorityLevel;

}

**public** Date get\_createdDate() {

**return** \_createdDate;

}

**public** **void** set\_createdDate(Date \_createdDate) {

**this**.\_createdDate = \_createdDate;

}

@Override

**public** String toString() {

SimpleDateFormat df=**new** SimpleDateFormat("dd-MM-yyyy");

DecimalFormat ddf=**new** DecimalFormat("#.#");

**return** \_name+"\t"+\_content+"\t"+ddf.format(\_size)+"\t"+ddf.format(\_priorityLevel)+"\t"+df.format(\_createdDate)+"\t";

}

**public** **static** Note CreateNote(String detail) **throws** NumberFormatException, ParseException {

String arr[]=detail.split(",");

SimpleDateFormat df=**new** SimpleDateFormat("dd-MM-yyyy");

Note note=**new** Note(arr[0],arr[1],Double.*parseDouble*(arr[2]),

Double.*parseDouble*(arr[3]),df.parse(arr[4]));

**return** note;

}

**public** **int** compareTo(Note n) {

// **TODO** Auto-generated method stub

**return** **this**.\_name.compareTo(n.\_name);

}

}

**class** Notebook{

**private** String \_name;

**private** List<Note> \_noteList=**new** ArrayList<Note>();

**public** Notebook(String \_name, List<Note> list) {

**super**();

**this**.\_name = \_name;

**this**.\_noteList = list;

}

Notebook(){}

}

**class** DateComparer **implements** Comparator<Note>{

@Override

**public** **int** compare(Note n1, Note n2) {

// **TODO** Auto-generated method stub

**return** n1.get\_createdDate().compareTo(n2.get\_createdDate());

}

}

**class** PriorityLevelComparer **implements** Comparator<Note>{

@Override

**public** **int** compare(Note n1, Note n2) {

**if**(n1.get\_priorityLevel()>n2.get\_priorityLevel()) **return** 1;

**else** **if**(n1.get\_priorityLevel()<n2.get\_priorityLevel()) **return** -1;

**else** **return** 0;

}

}