NATIONAL SENIOR CERTIFICATE EXAMINATION

MAY 2025

INFORMATION TECHNOLOGY: PAPER I

MARKING GUIDELINES

Time: 3 hours 150 marks

These marking guidelines are prepared for use by examiners and sub-examiners, all of whom are required to attend a standardisation meeting to ensure that the guidelines are consistently interpreted and applied in the marking of candidates' scripts.

The IEB will not enter into any discussions or correspondence about any marking guidelines. It is acknowledged that there may be different views about some matters of emphasis or detail in the guidelines. It is also recognised that, without the benefit of attendance at a standardisation meeting, there may be different interpretations of the application of the marking guidelines.

IEB Copyright © 2025

SECTION A SQL

QUESTION 1

Question 1.1 (4)

Access/MySQL/JavaDB:
```
SELECT *
FROM tblPlays
WHERE PubDom = TRUE OR NumActors > 30
```

Accept PubDom = TRUE, 1 or other acceptable variants

Question 1.2 (5)

Access:
```
SELECT TName, NumStaff * INT(RND(TheatreID) * 501 ) + 1
FROM tblTheatres
ORDER BY TName
```

MySQL:
```
SELECT TName, NumStaff * FLOOR( (RAND(TheatreID)* 501) + 1
FROM tblTheatres
ORDER BY TName
```

JavaDB:
```
SELECT TName, NumStaff * FLOOR( (RAND(TheatreID) * 501) + 1
FROM tblTheatres
ORDER BY TName
```

Question 1.3 (4)

Access/MySQL/JavaDB:
```
SELECT *
FROM tblTheatres
WHERE (NumStaff > 15 OR TicketPrice < 150) AND City <> "Durban"
```
Accept 'Durban' for MySQL/JavaDB

Question 1.4 (3)

Access/MySQL/JavaDB:
```
SELECT SUM (NumStaff)
FROM tblTheatres
```

Question 1.5 (4)
Access/MySQL/JavaDB:

```
SELECT ShowDate, COUNT (*) AS TotalPlays
FROM tblSales
GROUP BY ShowDate
```

Question 1.6 (4)

IEB Copyright © 2025

Access/MySQL/JavaDB:
```
SELECT PName, ShowDate
FROM tblPlays, tblSales
WHERE tblPlays.PlayID = tblSales.PlayID
```

Alternate Solution:
```
SELECT PName, ShowDate
FROM tblPlays INNER JOIN tblSales
ON tblPlays.PlayID = tblSales.PlayID
```

Question 1.7 (6)

Access:
```
SELECT LEFT(City, 1) & Right(City, 4) & "Feb" &
DAY(ShowDate)
FROM tblTheatres, tblSales
WHERE tblTheatres.TheatreID = tblSales.TheatreID
```

MySQL:
```
SELECT CONCAT (LEFT(City, 1), RIGHT(City, 4), 'Feb',DAY(ShowDate))
FROM tblTheatres, tblSales
WHERE tblTheatres.TheatreID = tblSales.TheatreID
```

JavaDB:
```
SELECT SUBSTR(City, 1, 1)||SUBSTR(City, LENGTH(City)-3,4)|| 'Feb'
|| CAST(DAY(ShowDate) AS CHAR)
FROM tblTheatres, tblSales
WHERE tblTheatres.TheatreID = tblSales.TheatreID
```

Alternate Solution:
Accept INNER JOINs for all languages
```
SELECT LEFT(City, 1) & Right(City, 4) & "Feb" &
DAY(ShowDate)
FROM tblTheatres
INNER JOIN tblSales
ON tblTheatres.TheatreID = tblSales.TheatreID
```

Question 1.8 (4)

Access/MySQL/JavaDB:
```
SELECT TName FROM
tblTheatres
WHERE TheatreID NOT IN (SELECT TheatreID FROM tblSales)
```

Alternate Solution:
```
SELECT TName
FROM tblTheatres LEFT JOIN tblSales
ON tblTheatres.TheatreID = tblSales.TheatreID
WHERE tblSales.PlayID IS NULL
```

Question 1.9 (8)

Access/MySQL/JavaDB:
```
SELECT TName, PName, AVG(Tickets)
FROM tblTheatres, tblPlays, tblSales
```

IEB Copyright © 2025

```
WHERE tblPlays.PlayID = tblSales.PlayID AND tblTheatres.TheatreID
= tblSales.TheatreID
GROUP BY TName, PName
HAVING AVG(Tickets) > 250
```

Question 1.10 (4)

Access/MySQL/JavaDB:

```
UPDATE tblPlays
SET PubDom = TRUE, Fee = 0
WHERE PName = "When Life Gives Lemons"
```

Question 1.11 (4)

Access:

```
DELETE FROMtblTheatres
WHERE TName LIKE "*GRAND*"
```

MySQL/JavaDB:

```
DELETE FROMtblTheatres
WHERE TName LIKE'%GRAND%'
```

50 marks

IEB Copyright © 2025

SECTION B

JAVA SOLUTION

QUESTION 2

```
//Question 2
//Question 2.1 - 3
//class header
//all fields private
//all correctly typed with correct names
public class Play {

    private String name;
    private boolean trap;
    private int license;
    private LocalDate first;
    private LocalDate last;

    //Question 2.2 - 3
    //fields static
    //fields final
    //fields correctly typed, named and correct values assigned
    public static final int LOCAL = 0;
    public static final int INTERNATIONAL = 1;

    //Question 2.3 - 4
    //correct method name
    //correct parameter names and types
    //fields assigned to parameters
    //inS and inE converted to dates
    public Play (String inN, boolean inT, int inLic, String inF,
                String inLast) {
        name = inN;
        trap = inT;
        license = inL;
        first = LocalDate.parse(inF);
        last = LocalDate.parse(inL);
    }

    //Question 2.4 - 2
    //correct headers
    //correct return statements
    //if one or more accessors are missing award only 1 mark.
    public String getName() {
        return name;
    }

    public LocalDate getFirst() {
        return start;
    }

    public LocalDate getLast() {
        return end;
```

IEB Copyright © 2025

```
    }

    public boolean getTrap() {
        return trap;
    }

    //Question 2.5 - 4
    //correct header
    //switch OR nested if on license
    //use of LOCAL and INTERNATIONAL constants
    //correct return values
    public String getLicense(){
        switch (license) {
            case LOCAL:
                return "Local";
            case INTERNATIONAL:
                return "International";
            default:
                return "Other";
        }
    }
    /*Alternate Solution
    public String getLicense(){
        String message = "Other";
        if (license == LOCAL) {
            message = "Local";
        } else if (license == INTERNATIONAL) {
            message = "International";
        }
        return message;
    }*/

    //Question 2.6 - 7
    //correct header
    //check if start dates are equal
    //check if start date of parameter is between start and end date
              of object
    //check if end date of parameter is between start and end date
              of object
    //OR conditions (or non-nested ifs)
    //correctly return true if clash
    //correctly return false if otherwise
    public boolean dateClash(Play inP){
        if ( inP.getFirst().equals(first)||
             inP.getFirst().isAfter(first)                          &&
               inP.getFirst().isBefore(last)||
             inP.getLast().isAfter(first)                           &&
               inP.getLast().isBefore(last)){
            return true;
        }
        return false;
    }

/*Alternate Solutions:
```

```
    public boolean dateClash(Play inPlay) {
        return inPlay.getFirst().equals(first) ||
               inPlay.getFirst.isAfter(first)                    &&
               inPlay.getFurst().isBefore(last) ||
               inPlay.getLast().isAfter(first)                   &&
               inPlay.getLast().isBefore(last));
    }

    public boolean dateClash(Play inPlay) {
        if (inPlay.getFirst.equals(first) {
            return true;
        }
        if            (inPlay.getFirst.isAfter(first)             &&
              inPlya.getFirst().isBefore(last)) {
            return true;
        }
        if           (inPlay.getLast().isAfter(first)             &&
              inPlay.getLast().isBefore(last)) {
            return true;
        }
    }

    public boolean dateClash(Play inPlay) {
        boolean clash = false;
        if ( inPlay.getFirst ().equals(first) ||
             inPlay.getFirst().isAfter(first)                     &&
               inPlay.getFirst().isBefore(last) ||
             inPlay.getLast().isAfter(firstr)                     &&
               inPlay.getLast().isBefore(last)) {
            clash = true;
        }
        return clash;
    */

    //Question 2.7 - 2
    //call to getLicense
    //return correctly formatted string
    public String toString() {
        return getLicense()+ "\t" + name + "\t" + first+ "\t" +
               last;
    }

}
```

25 marks

QUESTION 3 & 6.1

```
//Question 3.1 - 2
//class header
//all fields private, correctly named and typed
public class Theatre {

    private String name;
    private boolean trap;
    private int seats;
    private Play[] tPlaysArr = new Play[30];
    private int size = 0;

    //Question 3.2 - 2
    //correct header
    //parameters assigned correctly
    public Theatre(String inN, boolean inT, int inS){
        name = inN;
        trap = inT;
        seats = inS;
    }

    //Question 3.3 - 2
    //correct headers
    //correct return statements
    //if one or more accessors are missing award only 1 mark.
    public String getName(){
        return name;
    }

    public boolean getTrap() {
        return trap;
    }

    public int getSeats() {
        return seats;
    }

    //Question 3.4 - 7
    //correct header
    //correct first line, with tabs
    //check if size == 0
    //correctly append "No plays scheduled."
    //else loop through array
    //add tPlaysArr[i] to array
    //return string
    public String toString(){
        String ret = name + "\t" + trap + "\t" + seats + "
               seats.\n";
        if (size == 0){
            ret = ret + "No plays scheduled.";
        } else {
            for (int i = 0; i < size; i++){
                ret = ret + tPlaysArr[i] + "\n";
```

```
            }
        }
        return ret;
    }

    //Question 6.1 - 13
    //method header returns boolean and accept Play parameter
    //loop through internal array
    //check for dateClash
    //between parameter object and array object
    //return false if clash detected
    //check if parameter play needs trap
    //and theatre has trap
    //or if parameter play does NOT need a trap
    //add parameter play to tPlaysArr
    //in correct position
    //increment size
    //return true
    //return false outside of if
    public boolean addPlay(Play inP){
        for (int i = 0; i < size; i++) {
            if (inP.dateClash (tPlaysArr[i])) {
                return false;
            }
        }
        if (inP.getTrap() && trap || !inP.getTrap()) {
            tPlaysArr[size] = inP;
            size++;
            return true;
        }

        return false;
    }

}
```

13 marks + 13 marks

QUESTION 4

```
//Question 4.1 - 1
//class header
public class TheatreManager{

    //Question 4.2 - 3
    //tArr correctly instantiated
    //pArr correctly instantiated
    //both size variables correctly created (not necessary to set
              = 0)
    private Theatre[] tArr = new Theatre[50];
    private Play[] pArr = new Play[50];
    private int tSize = 0;
    private int pSize = 0;

    //Question 4.3 - 13
    //correct header
    //check if file exists and display error message if it doesn't
    //open file for reading
    //loop through file
    //read line from file
    //split line using ";" (Scanner or .split acceptable)
    //assign tokens to variables
    //check for additional token (or array length of 5 if using
              split
    //read additional values
    //create Play object and add to array
    //else create Theatre object and add to array
    //increment sizes of arrays
    //close file correctly
    public TheatreManager(){
        try {
            Scanner scFile = new Scanner(new
               File("Production.txt"));
            while (scFile.hasNextLine()) {
                Scanner scLine = new
               Scanner(scFile.nextLine()).useDelimiter(";");
                String name = scLine.next();
                boolean trap = scLine.nextBoolean();
                int num = scLine.nextInt();
                if (scLine.hasNext()){
                    String start = scLine.next();
                    String end = scLine.next();
                    pArr[pSize] = new Play(name, trap, num, start,
               end);
                    pSize++;
                } else {
                    tArr[tSize] = new Theatre(name, trap, num);
                    tSize++;
                }
                scLine.close();
            }
            scFile.close();
```

```
        } catch (FileNotFoundException ex) {
            System.out.println("Error. File not found.");
        }
    }

    //Question 4.4 - 5
    //correct header
    //create string
    //loop through array
    //add array element details to string
    //return string
    public String toString() {
        String ret = "";
        for (int i = 0; i < tSize; i++) {
            ret = ret + tArr[i].toString() + "\n";
        }
        return ret;
    }

    //Question 4.5 - 5
    //correct boundaries of outside loop
    //correct boundaries of inside loop
    //compare correct elements seats
    //store into temp variable
    //swap elements
    public void sortTheatre() {
        for (int i = 0; i < tSize - 1; i++) {
            for (int j = i + 1; j < tSize; j++) {
                if (tArr[i].getSeats() < tArr[j].getSeats()) {
                    Theatre temp = tArr[i];
                    tArr[i] = tArr[j];
                    tArr[j] = temp;
                }
            }
        }
    }
```

27 marks

IEB Copyright © 2025

QUESTION 6.2

```
//Question 6.2 - 17
//method header returns string and accepts no parameters
//Instantiate String to "Unassigned Plays"
//loop through pArr
//initialise j to 0 (do not award if inner loop is a for loop)
//initialise assigned flag to false
//loop through theatre array
//terminating loop if assigned == true
//Check if license is local and seats are >= 200
//call addplay method using tArr[j] and pArr[i]
//assign result to flag
//otherwise, check if license is international and seats are
           less than 200
//call addPlay method using tArr[i] and pArr[i]
//assign result to flag
//increment j (do not award if inner loop is a for loop)
//check if play was assigned after inner loop
//if not, add play name to string
//return string outside of outer loop
public String allocatePlays(){
    String unassigned = "Unassigned Plays\n";
    for (int i = 0; i < pSize; i++) {
        int j = 0;
        boolean assigned = false;
        while (!assigned&& j < tSize) {
            if (pArr[i].getLicense().equals("Local") &&
           tArr[j].getSeats() >= 200)
                assigned = tArr[j].addPlay(pArr[i]);
            else if
           (pArr[i].getLicense().equals("International") &&
           tArr[j].getSeats() < 200)
                assigned= tArr[j].addPlay(pArr[i]);
            j++;
        }
        if (!assigned) {
            unassigned = unassigned + pArr[i].getName() +
           "\n";
        }

    }
    return unassigned;
}
```

17 marks

IEB Copyright © 2025

QUESTION 5 & 6.3,

```
//Question 5.1 - 1
//class header with main method
public class TheatreUI {

    public static void main(String[] args){
        //Question 5.2 - 1
        //instantiate TheatreManager object inside or outside main
        method
        TheatreManager tm = new TheatreManager();
        //Question 5.3 - 1
        //call sortTheatre method
        tm.sortTheatre();
        //Question 5.4 - print object
        System.out.println(tm);

        //Question 6.3 - 1
        //print output of allocatePlays method
        System.out.println(tm.allocatePlays());
    }

}
```

4 + 1 marks

Total: 150 marks

IEB Copyright © 2025

DELPHI SOLUTION

QUESTION 2

```
//Question 2
//Question 2.1 - 3
//class header
unit Play;

interface
uses SysUtils, DateUtils;

type TPlay = class
  //all fields private
  //all correctly typed with correct names
  Private
    name : string;
    trap : boolean;
    first : Tdate;
    last : TDate;
    license : integer;

  public
    constructor Create(inN : string; inT : boolean; inLic :
                integer; inF : string; inLast : string);
    function getName() : string;
    function getFirst() : TDate;
    function getLast() : TDate;
    function getTrap() : boolean;
    function getLicense() : string;
    function dateClash(inP : TPlay) : boolean;
    function toString() : string;

    //Question 2.2 - 3
    //fields constant
    //fields in all caps
    //fields correctly typed, named and correct values assigned
    const
      LOCAL = 0;
      INTERNATIONAL = 1;

end;

implementation
  //Question 2.3 - 4
  //correct method name
  //correct parameter names and types
  //fields assigned to parameters
  //inS and inE converted to dates
  constructor TPlay.Create (inN: string; inT: Boolean; inLic:
                Integer; inF: string; inLast: string);
  var
    y, m, d: string;
  begin
```

IEB Copyright © 2025

```
    name := inN;
    trap := inT;
    license := inLic;
    y := Copy(inF, 1, Pos('-', inF) -1);
    Delete(inF, 1, Pos('-', inF));
    m := Copy(inF, 1, Pos('-', inF) -1);
    Delete(inF, 1, Pos('-', inF));
    d := inF;
    first := EncodeDate(StrToInt(y), StrToInt(m), StrToInt(d));
    y := Copy(inLast, 1, Pos('-', inLast) -1);
    Delete(inLast, 1, Pos('-', inLast));
    m := Copy(inLast, 1, Pos('-', inLast) -1);
    Delete(inLast, 1, Pos('-', inLast));
    d := inLast;
    last := EncodeDate(StrToInt(y), StrToInt(m), StrToInt(d));
end;


//Question 2.4 - 2
//correct method names
//correct return statements
//if one or more accessores are missing award only 1 mark.
function TPlay.getName: string;
begin
  Result := name;
end;

function TPlay.getFirst: TDate;
begin
  Result := first;
end;

function TPlay.getLast: TDate;
begin
  Result := last;
end;

function TPlay.getTrap: Boolean;
begin
  Result := trap;
end;

//Question 2.5 - 4
//correct header
//case OR nested if on license
//use of LOCAL and INTERNATIONAL constants
//correct return values
function TPlay.getLicense: string;
begin
  case license of
    LOCAL : Result := 'Local';
    INTERNATIONAL : Result := 'International';
    else Result := 'Other';
  end;
```

IEB Copyright © 2025

```
  end;

  //Question 2.6 - 7
  //correct header
  //check if start dates are equal
  //check if start date of parameter is between start and end date
              of object
  //check if end date of parameter is between start and end date
              of object
  //OR conditions (or non-nested ifs)
  //correctly return true if clash
  //correctly return false if otherwise
  function TPlay.dateClash(inP: TPlay): Boolean;
  begin
    Result := false;
    if System.DateUtils.SameDate(inP.getFirst, first)or
       System.DateUtils.DateInRange(inP.getFirst, first, last) or
       System.DateUtils.DateInRange(inP.getLast, first, last) then
       begin
         Result := true;
       end

  end;

  //Question 2.7 - 2
  //call to getLicense
  //return correctly formatted string
  function TPlay.toString: string;
  begin
    Result := getLicense() + #9 + name + #9 + DateToStr(first) +
              #9 + DateToStr(last);
  end;


end.
```

25 marks

IEB Copyright © 2025

QUESTION 3 & 6.2

```
//Question 3.1 - 2
//class header
unit Theatre;

interface
uses SysUtils, Play;

type TTheatre = class
  //all fields private, correctly named and typed
  Private
    name : string;
    trap : boolean;
    seats : integer;
    tPlaysArr : array[1..30] of TPlay;
    size : integer;
  public
    constructor Create(inN : string; inT : boolean; inS :
                integer);
    function getName() : string;
    function getTrap() : boolean;
    function getSeats() : integer;
    function addPlay(inP : TPlay) : boolean;
    function toString() : string;

end;

implementation
  //Question 3.2 - 2
  //correct method name and parameters
  //parameters assigned correctly
  constructor TTheatre.Create(inN: string; inT: Boolean; inS:
                Integer);
  begin
    name := inN;
    trap := inT;
    seats := inS;
  end;

  //Question 3.3 - 2
  //correct method names
  //correct return statements
  //if one or more accessores are missing award only 1 mark.
  function TTheatre.getName(): string;
  begin
      Result := name;
  end;

  function TTheatre.getTrap(): Boolean;
  begin
    Result := trap;
  end;
```

IEB Copyright © 2025

```
function TTheatre.getSeats(): Integer;
begin
  Result := seats;
end;

//Question 3.4 - 7
//correct method name
//correct first line, with tabs - NB this mark may be awarded
             even if the Boolean has NOT been converted to
             'true' or 'false'
//check if size == 0
//correctly append "No plays scheduled."
//else loop through array
//add tPlaysArr[i] to array
//return string
function TTheatre.toString(): string;
var
  ret, strTrap : string;
  i: integer;
begin
  if trap then
  begin
    strTrap := 'true';
  end
  else
  begin
    strTrap := 'false';
  end;
  ret := name + #9 + strTrap + #9 + IntToStr(seats) + ' seats.'
            + #13#10;
  if size = 0 then
  begin
      ret := ret + 'No plays scheduled.' + #13#10;
  end
  else
  begin
    for i := 1 to size do
    begin
      ret := ret + tPlaysArr[i].toString() + #13#10;
    end;
  end;
  Result := ret;
end;
```

```
//Question 6.1 - 13
//method returns boolean and accepts Play parameter
```

IEB Copyright © 2025

```
//loop through internal array
//check for dateClash
//between parameter object and array object
//set flag / Result to false if clash detected
//check if parameter play needs trap
//and theatre has trap
//or if parameter play does NOT need a trap
//add parameter play to tPlaysArr
//in correct position
//increment size
//set flag / Result to true inside if
//set flag / Result to false inside else
Function TTheatre.addPlay(inP: TPlay): Boolean;
 var
  i: integer;
begin
  Result := true;
  for i := 1 to size do
  begin
    if inP.dateClash (tPlaysArr[i]) then
    begin
      Result := false;
    end
  end;
  if (Result and ( (inP.getTrap() and trap) or (not
             inP.getTrap()) ) ) then
  begin
    Inc(size);
    tPlaysArr[size] := inP;
    Result := true;
  end
  else
  begin
    Result := false;
  end;
end;
```

13 marks + 13 marks

QUESTION 4

```
//Question 4.1 - 1
//class header
unit TheatreManager;

interface
uses SysUtils, Theatre, Play;

type TTheatreManager = class
  private
    //Question 4.2 - 3
    //tArr created as array with 50 cells
    //pArr created as array with 50 cells
    //both size variables created
    tArr : array[1..50] of TTheatre;
    tSize : integer;
    pArr : array[1..50] of TPlay;
    pSize : integer;

  public
    constructor Create();
    function toString() : string;
    procedure sortTheatre();
    function allocatePlays() : string;

end;

implementation
  //Question 4.3 - 13
  //correct header
  //check if file exists and display error message if it doesn't
  //open file for reading
  //loop through file
  //read line from file
  //split line using ";"
  //assign tokens to variables
  //check for additional token
  //read additional values
  //increment sizes of arrays
  //create Play object and add to array
  //else create Theatre object and add to array
  //close file correctly
  constructor TTheatreManager.Create;
    var
      inFile : textfile;
      line, name, first, last : string;
      trap : boolean;
      num : integer;
    begin
      if FileExists('Production.txt') <> true then
      begin
        WriteLn('File not found.');
      end
```

IEB Copyright © 2025

```
  else
  begin
    AssignFile(inFile, 'Production.txt');
    Reset(inFile);
    tSize := 0;
    pSize := 0;
    while NOT EOF(inFile) do
    begin
      ReadLN(infile, line);
      name := Copy(line, 1, Pos(';', line) -1);
      Delete(line, 1, Pos(';', line));
      trap := StrToBool(Copy(line, 1, Pos(';', line) -1));
      Delete(line, 1, Pos(';', line));
      if Pos(';', line) > 0 then
      begin
        num := StrToInt(Copy(line, 1, Pos(';', line) -1));
        Delete(line, 1, Pos(';', line));
        first := Copy(line, 1, Pos(';', line) -1);
        Delete(line, 1, Pos(';', line));
        last := line;
        Inc(pSize);
        pArr[pSize] := TPlay.Create(name, trap, num, first,
           last);
      end
      else
      begin
        num := StrToInt(line);
        Inc(tSize);
        tArr[tSize] := TTheatre.Create(name, trap, num);
      end;
    end;
    CloseFile(inFile);
  end;
end;

//Question 4.4 - 5
//correct method name
//create string
//loop through array
//add array element details to string
//return string
function TTheatreManager.toString: string;
var
  i : integer;
  ret : string;
begin
  ret := '';
  for i := 1 to tSize do
  begin
    ret := ret + tArr[i].toString() + #13#10;
  end;
  Result := ret;
end;
```

```
//Question 4.5 - 5
//correct boundaries of outside loop
//correct boundaries of inside loop
//compare correct elements seats
//store into temp variable
//swap elements
procedure TTheatreManager.sortTheatre;
var
  i, j : integer;
  temp : TTheatre;
begin
  for i := 1 to tSize do
  begin
    for j := 1 to tSize - 1 do
    begin
      if tArr[i].getSeats() > tArr[j].getSeats() then
      begin
        temp := tArr[i];
        tArr[i] := tArr[j];
        tArr[j] := temp;
      end;

    end;

  end;
end;
```

27 marks

QUESTION 6.2

```
//Question 6.2 - 17
//method returns string and accepts no parameters
//Instantiate String to "Unassigned Plays"
//loop through pArr
//initialise j to 1 (do not award if inner loop is a for loop)
//initialise assigned flag to false
//loop through theatre array
//terminating loop if assigned = true
//Check if license is local and seats are >= 200
//call addplay method using tArr[j] and pArr[i]
//assign result to flag
//otherwise, check if license is international and seats are
            less than 200
//call addPlay method using tArr[i] and pArr[i]
//assign result to flag
//increment j (do not award if inner loop is a for loop)
//check if play was assigned after inner loop
//if not, add play name to string
//return string outside of outer loop
function TTheatreManager.allocatePlays: string;
var
  unassigned : string;
  i, j : integer;
  assigned : boolean;
begin
  unassigned := 'Unassigned Plays' + #13#10;
  for i := 1 to pSize do
  begin
    j := 1;
    assigned := false;
    while (not assigned) and (j <= tSize) do
    begin
      if (pArr[i].getLicense() = 'Local') and
           (tArr[j].getSeats() >= 200) then
      begin
        assigned := tArr[j].addPlay(pArr[i]);
      end
      else
      begin
        if (pArr[i].getLicense() = 'International') and
           (tArr[j].getSeats() < 200) then
        begin
          assigned := tArr[j].addPlay(pArr[i]);
        end;
      end;
      Inc(j);
    end;
    if (not assigned) then
    begin
      unassigned := unassigned + pArr[i].getName() + #13#10;
    end;
  end;
```

IEB Copyright © 2025

```
    Result := unassigned;
  end;
```

17 marks

IEB Copyright © 2025

QUESTION 5 & 6.3,

```
//Question 5.1 - 1
//class created
program TheatreUI;

{$APPTYPE CONSOLE}

{$R *.res}

uses
  System.SysUtils,
  Play in 'Play.pas',
  Theatre in 'Theatre.pas',
  TheatreManager in 'TheatreManager.pas';

var
  tm : TTheatreManager;

begin
  try
    { TODO -oUser -cConsole Main : Insert code here }
    //Question 5.2 - 1
    //instantiate TTheatreManager object
    tm := TTheatreManager.Create();

    //Question 5.3 - 1
    //call sortTheatre method
    tm.sortTheatre();

    //Question 5.4 - 1
    //print object
    WriteLn(tm.toString());

    //Question 6.3 - 1
    //print output of allocatePlays method.
    WriteLn(tm.allocatePlays());

    ReadLn;
  except
    on E: Exception do
      Writeln(E.ClassName, ': ', E.Message);
  end;
end.
```

4 + 1 marks

Total: 150 marks

IEB Copyright © 2025