

מחלקה Train

עליכם לכתוב מחלקה בשם Train המייצגת נסיעה.
למחלקה 5 תכונות, `_booked`, `_seats`, מסוג מספר שלם, `_TrainDate`,
מסוג `Date`, `_origin`, `_destination` מסוג `String`.
כתבו בנאי למחלקה המקבל את המספר המקומות המקסימלי בנסיעה,
תאריך הנסיעה, יעד ומוצא. מספר המקומות התפוסים יקבל ערך אפס.
כמוכן כתבו בנאי העתקה המקבל נסיעה ומעתיק את התכונות לנסיעה
חדשה.

כתבו שיטת `get` ו-`set` לכל תכונה.
כתבו שיטה `toString` המחזירה מחרוזת עם כל פרטי הנסיעה, כל פרט
בשורה נפרדת. כתבו גם שיטה `equals` המקבלת נסיעה נוספת ובודקת אם
הטיסות שוות (השוו בין התאריכים, היעד והמוצא של הטיסות).
כמוכן כתבו שיטה בוליאנית בשם `full` הבודקת אם הנסיעה מלאה, שיטה
`placesLeft` המחזירה את מספר המקומות הפנויים בנסיעה, ושיטה
`book` המקבלת מספר מקומות להזמין, מאפשרת להזמין מקומות
בנסיעה ומחזירה אמת אם ניתן לבצע את ההזמנה.
כתבו API למחלקה.

הוסיפו לפרוייקט מחלקה חדשה בשם `TrainTester` שתכיל רק שיטת
`main`. קוד ההרצה שתכתבו בשביל לבדוק את המחלקה `Train` ייכתב
בשיטה זו.
הגדירו 2 תאריכים `date1, date2` עם תאריכים 1/3/2011 ו-
23/3/2011.
הגדירו 2 טיסות בעלי 100 מקומות כל אחת בתאריך הראשון והשני מת"א,
הראשונה לרומא והשניה ללונדון.
הגדירו נסיעה שלישית עם אותן התכונות של הנסיעה הראשונה.
השוו בין הנסיעה הראשונה לשניה והדפיסו הודעה מתאימה אם הטיסות
שוות או לא.
השוו בין הנסיעה הראשונה לשלישית והדפיסו הודעה מתאימה אם הטיסות
שוות או לא.
הדפיסו את פרטי שלושת הטיסות.
שינו את היום של התאריך הראשון ל-2.
השוו בין הנסיעה הראשונה לשלישית והדפיסו הודעה מתאימה.

שינו את תאריך הנסיעה של הנסיעה הראשונה לתאריך הראשון.
השוו בין הנסיעה הראשונה לשלישית והדפסו הודעה מתאימה.
עדכנו את החודש של הנסיעה הראשונה לחודש הבא.
הדפסו את הנסיעה הראשונה.
הזמינו 5 מקומות בנסיעה הראשונה.
הדפסו כמה מקומות פנויים יש בנסיעה הראשונה.
בדקו אם יש מקומות בנסיעה הראשונה.
כתבו API למחלקה.
ניסו לבצע בדיקות של המחלקה בעזרת ספסל האובייקטים של
blue j.

```

public class Train
{
    private int _seats;
    private int _booked;
    private Date _TrainDate;
    private String _destination;
    private String _origin;

    public Train (int seats, Date date,String dest,String origin)
    {
        _seats=seats;
        _booked=0;
        _TrainDate=new Date(date);
        _destination=dest;
        _origin=origin;
    }
    //Copy constructor
    public Train (Train other)
    {
        _seats=other._seats;
        _booked=other._booked;
        _TrainDate=new Date(other._TrainDate);
        _destination=other._destination;
        _origin=other._origin;
    }

    public int getSeats()
    {
        return _seats;
    }
    public int getBooked()
    {
        return _booked;
    }
    public Date getTrainDate()
    {
        return new Date (_TrainDate);
    }
    public String getDestination()
    {
        return _destination;
    }
    public String getOrigin()

```

```

{
    return _origin;
}
public void setSeats(int seats)
{
    _seats=seats;
}
public void setBooked(int booked)
{
    _booked=booked;
}
public void setTrainDate(Date newDate)
{
    _TrainDate=new Date(newDate);
}
public void setDestination(String dest)
{
    _destination=dest;
}
public void setOrigin(String origin)
{
    _origin=origin;
}

public String toString()
{
    return (" Date : "+_TrainDate+"\nDestination: "+_destination+
"\nOrigin: "+_origin+ "\nCapacity :"+ _seats+ "\nBooked: "+_booked);
}
public boolean equals(Train other){
    if
(_TrainDate.equals(other._TrainDate)&&_destination.equals(other._destination
)&&_origin.equals(other._origin))
        return true;
    return false;
}
public boolean full(){
    if (_booked==_seats )
        return true;
    return false;
}
public int placesLeft(){
    return _seats-_booked;
}

```

```
    }  
    public boolean book(int places)  
    {  
        if (places <=placesLeft())  
        {  
            _booked+=places;  
            return true;  
        }  
        return false;  
    }  
}
```

```

public class TrainTester
{
    public static void main (String[]args)
    {
        Date date1=new Date(1,3,2011);
        Date date2=new Date(23,3,2011);
        Train elal1=new Train(100,date1,"Rome","Tel-Aviv");
        Train elal2=new Train(100,date2,"London","Tel-Aviv");
        Train elal3=new Train(elal1);
        if (elal1.equals(elal2))
            System.out.println("elal1,elal2 alternative Trains");
        else
            System.out.println("elal1,elal2 not alternative Trains");
        if (elal1.equals(elal3))
            System.out.println("elal1,elal3 alternative Trains");
        else
            System.out.println("elal1,elal3 not alternative Trains");
        System.out.println(elal1.toString());
        System.out.println(elal2.toString());
        System.out.println(elal3.toString());
        date1.setDay(2);
        if (elal1.equals(elal3))
            System.out.println("elal1,elal3 alternative Trains");
        else
            System.out.println("elal1,elal3 not alternative Trains");
        elal1.setTrainDate(date1);
        if (elal1.equals(elal3))
            System.out.println("elal1,elal3 alternative Trains");
        else
            System.out.println("elal1,elal3 not alternative Trains");

        System.out.println(elal1.getTrainDate().getYear());
        Date tempDate=elal1.getTrainDate();
        int tempMonth=tempDate.getMonth();
        if (tempMonth==12){
            tempMonth=1;
            int tempYear=tempDate.getYear();
            tempYear++;
            tempDate.setYear(tempYear);
        }
        else
            tempMonth++;
        tempDate.setMonth(tempMonth);
    }
}

```

```

elal1.setTrainDate(tempDate);
System.out.println(elal1.toString());
elal1.book(5);
System.out.println("Places left on ElAl1 Train :"+elal1.placesLeft());
System.out.println("Train Elal1 is: " + (elal1.full()?"full":"not full"));

```

הפלט:

elal1,elal2 not alternative Trains

elal1,elal3 alternative Trains

Date : 1/3/2011

Destination: Rome

Origin: Tel-Aviv

Capacity :100

Booked: 0

Date : 23/3/2011

Destination: London

Origin: Tel-Aviv

Capacity :100

Booked: 0

Date : 1/3/2011

Destination: Rome

Origin: Tel-Aviv

Capacity :100

Booked: 0

elal1,elal3 alternative Trains

elal1,elal3 not alternative Trains

2011

Date : 2/4/2011

Destination: Rome

Origin: Tel-Aviv

Capacity :100

Booked: 0

Places left on ElAl1 Train :95

Train Elal1 is: not full