

תאריך: \_\_\_\_ . \_\_\_\_ . \_\_\_\_

מקצוע: מדעי המחשב ב' - JAVA

שם: \_\_\_\_\_

נושא: אובייקטים

## סדר עבודה – ספר טלפונים - פתרון

\*\*\*\*\*

נתונים 4 קבצים:

Contact.java , PhoneBookDynamic.java , PhoneBookStatic.java , CheckPhoneBook.java

1. העתק/י את הקבצים ל-Eclipse.
2. השלם/י את הקווים החסרים.
3. הרץ/י את התכנית הראשית.
4. לכל פעולה הגדר/י סדר גודל.

```
import unit4.utilsLib.Date;

public class Contact
{

    private String name;
    private String phoneNumber;
    private Date bdate;

    // ***** Constructors *****

    public Contact()
    {
        this.name = "no name";
        this.phoneNumber = "000-0000000";
        this.bdate = new Date(1,1,1900);
    }

    public Contact(String name, String phoneNumber, Date bdate)
    {
        this.name = name;
        this.phoneNumber = phoneNumber;
        this.bdate=bdate;
    }

    public Contact(Contact c)
    {
        this.name = c.name;
        this.phoneNumber = c.phoneNumber;
        this.bdate = c.bdate;
    }
}
```



```
// ***** Set + Get *****

public String getName()
{
    return this.name;
}
public void setName(String name)
{
    this.name = name;
}

public String getPhoneNumber()
{
    return this.phoneNumber;
}
public void setPhoneNumber(String phoneNumber)
{
    this.phoneNumber = phoneNumber;
}

public Date getBdate()
{
    return this.bdate;
}
public void setBdate(Date bdate)
{
    this.bdate = bdate;
}

// ***** equals + toString *****

public boolean equals(Contact c) ← // Equals Contact
{
    if ( !this.name.equals(c.name))
        return false;
    if ( !this.phoneNumber.equals(c.phoneNumber))
        return false;
    return ( !this.bdate.equals(c.bdate));
}

public boolean equals(Date d) ← // Equals Date
{
    if ( this.bdate.getDay() != d.getDay())
        return false;
    if ( this.bdate.getMonth() != d.getMonth())
        return false;
    return ( this.bdate.getYear() == d.getYear());
}

public String toString()
{
    return ( "Contact info -> Name:"+this.name+
             " Phone Number:"+this.phoneNumber+
             " Birthday:"+this.bdate.toString() + "\n");
}
}
```

---



```
import unit4.utilsLib.Date;

public class PhoneBookStatic
{
    private Contact[] contacts = new Contact[1000];
    private int size;

    // ***** Constructors *****
    public PhoneBookStatic()
    {
        for(int i=0 ; i < this.contacts.length ; i++ )
            this.contacts[i] = null;
        this.size=0;
    }

    private void addPlaceToArray (int n)
    {
        Contact[] temp = new Contact[this.contacts.length+n];
        for( int i = 0 ; i < this.size ; i++ )
            temp[i] = this.contacts[i];
        this.contacts = temp;
    }

    // Add Contact to Phones Book
    public void addContact(String name ,String phone , Date bDate)
    {
        if ( this.size == this.contacts.length)           // Full Phones Book
            this.addPlaceToArray(10);                     // Add 10 places
        int i = 0;

        // Search for Contact
        While (( i < this.size ) &&
            (!this.contacts[i].getName().equals(name)))
            i++;

        if ( i == this.size )    // Contact doesn't Exist - Add Contact
        {
            this.contacts[this.size] = new Contact(name,phone,bDate);
            this.size++;
        }
        else                    // if exist - Update Contact

            this.contacts[i].setPhoneNumber(phone);
    }

    // Delete Contact From Phones Book
    public void delContact(String name)
    {
        int i = 0;
        while ((i < this.size) &&
            (!this.contacts[i].getName().equals(name)))
            i++ ;
        for( ; i < this.size - 1 ; i++ )
            this.contacts[i] = this.contacts[i+1];
        this.size--;
    }
}
```



```
// Return Phone number by Name
public String getPhone(String name)
{
    int i = 0;
    while ((i<this.size) &&
        (!this.contacts[i].getName().equals(name)))
        i++ ;
    if(i==this.size)
        return null;
    return this.contacts[i].getPhoneNumber();
}

// Returns Array of Contacts name
public String[] getAllContactsNames()
{
    String[] names = new String[ this.size ];
    for(int i=0 ; i < names.length ; i++)
        names[i]=this.contacts[i].getName();
    return names;
}

public String toString()
{
    String temp="";
    for( int i=0 ; i<this.size ; i++ )
        temp += this.contacts[i].toString();
    return temp;
}
}
```

-----

```
import unit4.collectionsLib.Node;
```

```
import unit4.utilsLib.Date;
```

```
public class PhoneBookDynamic
```

```
{
    private Node<Contact> contacts;
    private int size;
```

```
// ***** Constructors *****
```

```
public PhoneBookDynamic()
```

```
{
    this.contacts=null;
    this.size=0;
}
```



```
// Add Contact to Phones Book

public void addContact(String name, String phone , Date bDate)
{
    if (this.contacts == null) //PhoneBook is Empty(this.size==0)
    {
        this.contacts =
            new Node<Contact>(new Contact(name,phone, bDate));
        this.size++;
    }
    else
    {
        // Find Contact (if exist)
        Node<Contact> p = this.contacts;
        while ((p != null) && (!p.getInfo().getName().equals(name)))
            p = p.getNext();

        if ( p == null ) // if not exist - insert new Contact
        {
            Node<Contact> temp = new Node<Contact>
                (new Contact(name,phone, bDate));
            temp.setNext(this.contacts);
            this.contacts = temp;
            this.size++;
        }
        else // if exist - Update Contact
        {
            p.getInfo().setPhoneNumber(phone);
        }
    }
}

// Delete Contact From Phones Book
// Assumption: Contact exist

public void delContact(String name)
{
    Node<Contact> p = this.contacts;
    while ((p != null) && (!p.getInfo().getName().equals(name)))
        p = p.getNext();

    Node<Contact> temp = this.prev(p);
    if ( temp == this.contacts )
    {
        this.contacts = this.contacts.getNext();
        temp.setNext(null);
    }
    else
    {
        temp.setNext(p.getNext());
        p.setNext(null);
    }
}
```



```
// Returns Position of Previos p
private Node<Contact> prev (Node<Contact> p)
{
    if ( p == this.contacts )    // p is the first Node
        return p;

    Node<Contact> q = this.contacts;
    while ( q.getNext() != p)
        q = q.getNext();
    return q;
}

// Return Phone number by Name
public String getPhone(String name)
{
    Node<Contact> p = this.contacts;
    while ((p != null) && (!p.getInfo().getName().equals(name)))
        p = p.getNext();
    if ( p == null )
        return null ;
    return p.getInfo().getPhoneNumber();
}

// Returns Array of Contacts name
public String[] getAllContactsNames()
{
    String[] names = new String[ this.size ];
    Node<Contact> p = this.contacts;
    int i = 0 ;
    while ( p != null )
    {
        names[i] = p.getInfo().getName();
        p = p.getNext();
    }
    return names;
}

public String toString()
{
    String temp="";
    Node<Contact> p = this.contacts;
    while ( p != null )
    {
        temp += p.getInfo().toString();
        p = p.getNext();
    }
    return temp;
}
}
```

---



```
import unit4.utilsLib.Date;

public class CheckPhoneBook
{
    public static void main(String[] args)
    {
        PhoneBookDynamic phb= new PhoneBookDynamic();
        phb.addContact("Ran", "0521122334", new Date(27,10,2002));
        phb.addContact("Shiri", "0526677889", new Date(23,2,2013));
        System.out.println(phb.toString());
        phb.addContact("Oreen", "0522244668", new Date(23,2,2013));
        System.out.println(phb.getPhone("Oreen"));
        System.out.println(phb.getPhone("Shiri"));
        phb.addContact("Shiri", "0521111111", new Date(23,2,2013));
        System.out.println(phb.toString());
        phb.delContact("Shiri");
        System.out.println(phb.toString());

        PhoneBookStatic phbs= new PhoneBookStatic();
        phbs.addContact("Ran", "0521122334", new Date(27,10,2002));
        phbs.addContact("Shiri", "0526677889", new Date(23,2,2013));
        System.out.println(phbs.toString());
        phbs.addContact("Oreen", "0522244668", new Date(23,2,2013));
        System.out.println(phbs.getPhone("Oreen"));
        System.out.println(phbs.getPhone("Shiri"));
        phbs.addContact("Shiri", "0521111111", new Date(23,2,2013));
        System.out.println(phbs.toString());
        phbs.delContact("Shiri");
        System.out.println(phbs.toString());
    }
}
```



תאריך: \_\_\_\_ . \_\_\_\_ . \_\_\_\_

מקצוע: מדעי המחשב ב' - JAVA

שם: \_\_\_\_\_

נושא: אובייקטים מורכבים

## סדף עבודה מס' 4 – ספר טלפונים – יעילות

\*\*\*\*\*

נתונות 2 פעולות למחיקת איש קשר:

לכל פעולה הגדרי/ סדר גודל, נמק/!

// Delete Contact From Phones Book

```
public void delContact(String name)
{
    for ( int i = 0 ; i < this.size ; i++ )
        if (this.contacts[i].getName().equals(name))
            for ( ; i < this.size - 1 ; i++ )
                this.contacts[i] = this.contacts[i+1];
    this.size-- ;
}
```

 $O(n)$  - מצוין את מספר אנשי הקשר בספר הטלפונים.

מתבצע מעבר אחד על כל אנשי הקשר. בחלק הראשון של המעבר מחפשים את מיקום איש הקשר

בספר הטלפונים, ובחלק השני מזיזים את אנשי הקשר מימינו תא אחד שמאלה.

// Delete Contact From Phones Book

```
public void delContact(String name)
{
    int i = 0;
    while ((i < this.size) &&
        (!this.contacts[i].getName().equals(name)))
        i++ ;
    for ( ; i < this.size - 1 ; i++ )
        this.contacts[i] = this.contacts[i+1];
    this.size-- ;
}
```

 $O(n)$  - מצוין את מספר אנשי הקשר בספר הטלפונים.

מתבצע מעבר אחד על כל אנשי הקשר. בחלק הראשון של המעבר מחפשים את מיקום איש הקשר

בספר הטלפונים, ובחלק השני מזיזים את אנשי הקשר מימינו תא אחד שמאלה.

[דף\_עבודה - ספר טלפונים - פתרון]



עבודה נעימה !