

**دانشگاه صنعتی امیر کبیر**  
**( پلی تکنیک تهران )**

دانشکده مهندسی کامپیوتر

**کارگاه برنامه نویسی پیشرفته**

نیم سال دوم ۰۲-۰۳

عنوان:

شبیه سازی دفترچه تلفن

فروردین 1403

## دفترچه تلفن با جاوا

توضیح کلاس‌ها به صورت زیر است:

### - Address

این کلاس جهت ذخیره‌سازی آدرس مخاطبین استفاده می‌شود و دارای فیلدهای زیر است:

- zipCode

کد پستی مخاطب را به صورت یک رشته ذخیره می‌کند (می‌تواند خالی باشد).

- country

کشور مربوط به مخاطب را به صورت یک رشته ذخیره می‌کند (می‌تواند خالی باشد).

- city

هر مخاطب می‌تواند یک ایمیل داشته باشد (یا ایمیلی نداشته باشد) که به صورت یک رشته ذخیره می‌شود.

توضیحات متدهای این کلاس نیز به صورت زیر است:

- String toString()

تمامی اطلاعات آدرس را با فرمت مناسب در یک رشته باز می‌گرداند.

### - PhoneNumber

این کلاس به منظور ذخیره‌سازی شماره‌ی تلفن است. توضیحات فیلدهای آن به صورت زیر است:

- countryCode

کد کشور شماره‌ی مربوطه است. برای مثال برای ایران، این مقدار برابر با «+98» است.

- number

شماره‌ی تلفن مربوطه است، که در یک رشته با طول دقیقاً ۱۲ ذخیره می‌شود. برای مثال این مقدار می‌تواند

برابر با «9121234567» باشد.

توضیحات متدهای این کلاس این به شرح زیر است:

- String toString()

تمامی اطلاعات شماره‌ی تلفن را با فرمت مناسب در یک رشته بر می‌گرداند.

## - Contact

این کلاس درواقع یک شیء از مخاطب می‌باشد و شامل تمام اطلاعات مورد نیاز برای یک مخاطب است. توضیح فیلدهای آن به صورت زیر است:

- group

هر مخاطب می‌تواند عضو یک گروه خاص باشد (یا عضو هیچ گروهی نباشد) که به صورت یک رشته ذخیره می‌شود.

- email

هر مخاطب می‌تواند یک ایمیل داشته باشد (یا هیچ ایمیلی نداشته باشد) که به صورت یک رشته ذخیره می‌شود.

- firstName

یک رشته که نام مخاطب در آن ذخیره می‌شود.

- lastName

یک رشته که نام خانوادگی مخاطب در آن ذخیره می‌شود.

- phoneNumber

یک شیء از کلاس PhoneNumber که اطلاعات مربوط به شماره تلفن مخاطب را ذخیره می‌کند.

- address

یک شیء از کلاس Address که اطلاعات مربوط به آدرس مخاطب را ذخیره می‌کند.

## - PhoneBook

در این کلاس تمامی مخاطبین ذخیره می‌شوند. توضیح فیلدهای آن به صورت زیر است:

- `Contacts` و `contactCount`

یک لیست از تمامی مخاطبین است و تعداد آن در `contactCount` ذخیره می شود.

توضیح متدهای این کلاس نیز به صورت زیر است:

- `boolean addContact(Contact contact)`

یک مخاطب را به لیست اضافه می کند. اگر مخاطب در آرایه موجود بود، `false` برمی گرداند و کاری انجام نمی دهد، در غیر این صورت مخاطب را به آرایه اضافه می کند و `true` برمی گرداند.

- `boolean deleteContact(String firstName, String lastName)`

براساس نام داده شده به متد، مخاطب را پیدا می کند و آن را حذف می کند و `true` برمی گرداند. اگر مخاطب در آرایه موجود نبود، `false` برمی گرداند.

- `Contact[] findContact(String inputStr)`

براساس ورودی داده شده به متد، مخاطبین را پیدا می کند و آن را برمی گرداند. اگر مخاطب در آرایه موجود نبود، `null` برمی گرداند.

- `Contact[] findContacts(String searchType)`

تمامی مخاطبین یافت شده از روی `searchType` تحت عنوان یک آرایه برمی گرداند. اگر هیچ مخاطبی وجود نداشت، `null` برمی گرداند. `searchType` ممکن است گروه یا شهر یا کشور یا کد کشور باشد.

- `void printContacts()`

لیست مخاطبین را در قالبی مناسب نمایش می دهد.

در شکل زیر می توانید یک دیاگرام از تمام کلاس های مورد نیاز را ببینید: (پیاده سازی متدهای گفته شده الزامیست، طبیعی ست که در صورت نیاز می توانید متدهای دیگری نیز علاوه بر این متدها پیاده سازی کنید. درضمن هر جا که نیاز بود، متدهای `getter` و `setter` را نیز اضافه کنید)

## Parser

وظیفه ی این کلاس این است که دستور وارد شده را بخش بندی کند و سپس خواسته دستور را با استفاده از متد های سایر کلاس ها برآورده سازد.

## Main

در متد main این کلاس برنامه ای بنویسید که ورودی های زیر را بگیرد و خروجی مورد انتظار را تولید کند.

### ورودی:

- 1- contacts -a <contact firstName> <contact lastName>
- 2- contacts -r <contact firstName> <contact lastName>
- 3- show -g <search type> <search word >
- 4- show -c <contact info>
- 5- show
- 6- exit

۱. به منظور اضافه کردن مخاطب (در ادامه یک مثال از این فرایند خواهیم دید)

۲. به منظور حذف یک مخاطب (در صورتی که کاربر وجود داشت، Ok و در غیر این صورت Not found چاپ شود)

۳. به منظور نمایش دادن اطلاعات مخاطبان در یک گروه خاص خواسته شده

۴. به منظور نمایش اطلاعات مخاطبین با اطلاعات داده شده

۵. نمایش نام تمام مخاطبان

۶. خروج از برنامه

### خروجی:

دست شما برای تعیین فرمت خروجی برنامه باز است و می توانید هرگونه که خواستید خروجی ها را نمایش دهید. صرفاً خروجی باید تمیز و خوانا باشد.

به مثال‌های خروجی زیر دقت کنید. مثال برای دستور show:

```
Contact {  
  group: aut  
  email: -  
  firstName: Professor  
  lastName: Bakhshi  
  phoneNumber: (+98) 9123456789  
  address: 1234567890 - IRN - Tehran  
}
```

(نمونه‌ای از خروجی دستور show)

دقت کنید که در صورت خالی بودن هر یک از فیلدها، کاراکتر مناسبی چاپ کنید. (مانند مثال بالا، برای فیلد

email)

مثال برای دستور اضافه کردن مخاطب:

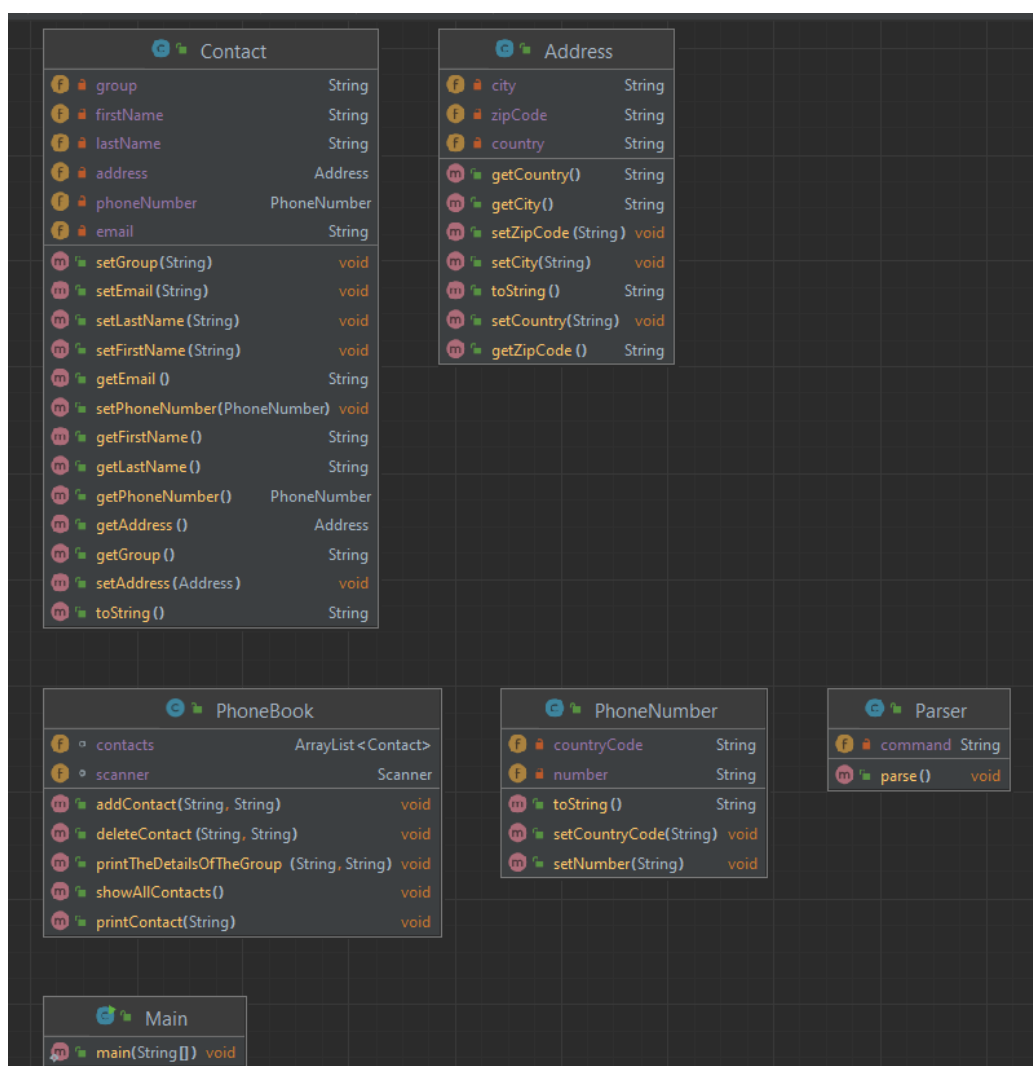
```
Input: contacts -a Professor Bakhshi  
Output: "Please enter contact's group: "  
Input: AUT  
Output: "Please enter contact's email: "  
Input:  
Output: "Please enter contact's country code: "  
Input: +98  
Output: "Please enter contact's phone number: "  
Input: 9123456789  
Output: "Please enter contact's zip code: "  
Input: 1234567890  
Output: "Please enter contact's country: "  
Input: IRN  
Output: "Please enter contact's city: "  
Input: Tehran  
Output: "Contact saved!"
```

(نمونه‌ای از خروجی دستور اضافه کردن مخاطب)

## نکات تحویل

- 1- محدودیت تعداد کاربران توسط **final** یا **static** کنترل شود - با توجه به کاربرد این دو -
- 2- لیست کاربران با استفاده از **Collection** پیاده سازی شود و استفاده از آرایه مجاز نیست.
- 3- مواقع پیمایش روی لیست ها از **Iterator** استفاده شود و همچنین استفاده از حلقه **for** مجاز نیست.
- 4- جستجو مخاطبین جزییاتی دارد که باید به آن توجه داشت از جمله ممکن است بخشی از اسم فرد وارد شود یا حتی ورودی ترکیبی از حروف بزرگ و کوچک باشد که با روش های گوناگون باید این را هندل کنید تا سرچ شما به اصطلاح **case\_insensitive** باشد.
- 5- تحویل پروژه حتما با استفاده از گیت و کامیت های مناسب انجام شود.

دیاگرام کلاس ها برای توضیحات بیشتر



## دیاگرام ارتباط کلاس ها با یکدیگر

