

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 12** 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 شئ 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 شئ از کلاس PhoneNumber که اطلاعات مربوط به شماره تلفن مخاطب را ذخیره می‌کند.

* address

یک شئ 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 می‌کند. اگر مخاطب در آرایه موجود بود، false برمی‌گرداند و کاری انجام نمی‌دهد، در غیر این‌صورت مخاطب را به آرایه اضافه می‌کند و true برمی‌گرداند.

* boolean deleteContact(String firstName, String lastName)

براساس نام داده شده به متد، مخاطب را پیدا می‌کند و آن را حذف می‌کند و true برمی‌گرداند. اگر مخاطب در آرایه موجود نبود، false برمی‌گرداند.

* Contact[] findContact(String inputStr)

براساس ورودی داده شده به متد، مخاطبین را پیدا می‌کند و آن را برمی‌گرداند. اگر مخاطب در آرایه موجود نبود، null برمی‌گرداند.

* Contact[] findContacts(String searchType)

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

* void printContacts()

لیست مخاطبین را در قالبی مناسب نمایش می‌دهد.

در شکل زیر می‌توانید یک دیاگرام از تمام کلاس‌‌های مورد نیاز را ببینید: (پیاده‌سازی متد‌های گفته شده الزامیست، طبیعی‌ست که در صورت نیاز می‌توانید متد‌های دیگری نیز علاوه بر این متد‌ها پیاده‌سازی کنید. درضمن هرجا 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 برنامه‌ای بنویسید که ورودی‌های 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

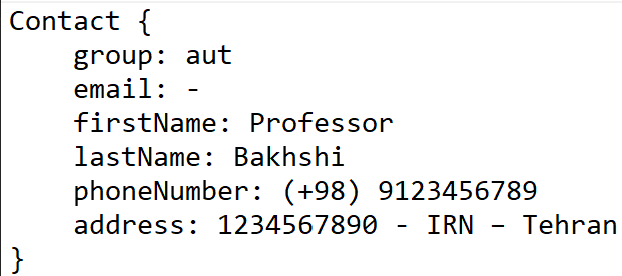
1. به‌منظور اضافه‌کردن مخاطب (در ادامه یک مثال از این فرایند خواهیم دید)
2. به‌منظور حذف یک مخاطب (در صورتی که کاربر وجود داشت، Ok و در غیر این صورت Not found چاپ شود)
3. به‌منظور نمایش دادن اطلاعات مخاطبان در یک گروه خاص خواسته شده
4. به‌منظور نمایش اطلاعات مخاطبین با اطلاعات داده شده
5. نمایش نام تمام مخاطبان
6. خروج از برنامه

**خروجی:**

دست شما برای تعیین فرمت خروجی برنامه باز است و می‌توانید هرگونه که خواستید خروجی‌ها را نمایش دهید.

صرفاً خروجی باید تمیز و خوانا باشد.

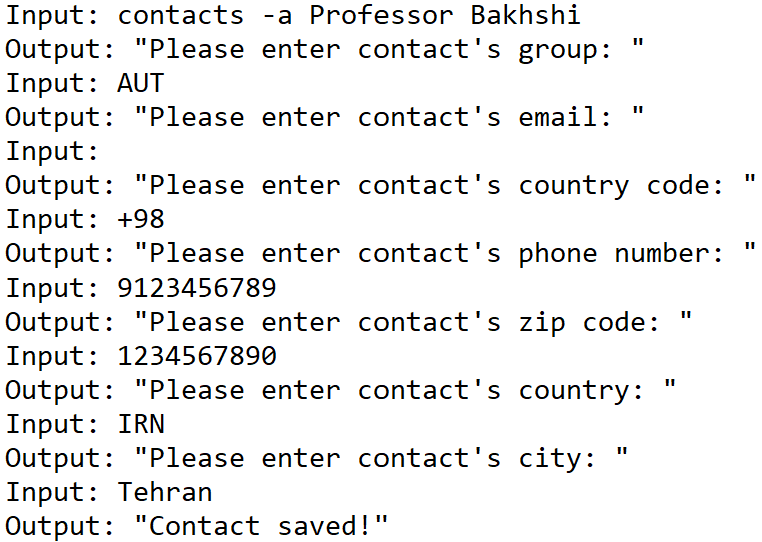
به مثال‌های خروجی زیر دقت کنید. مثال برای دستور show:



(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:

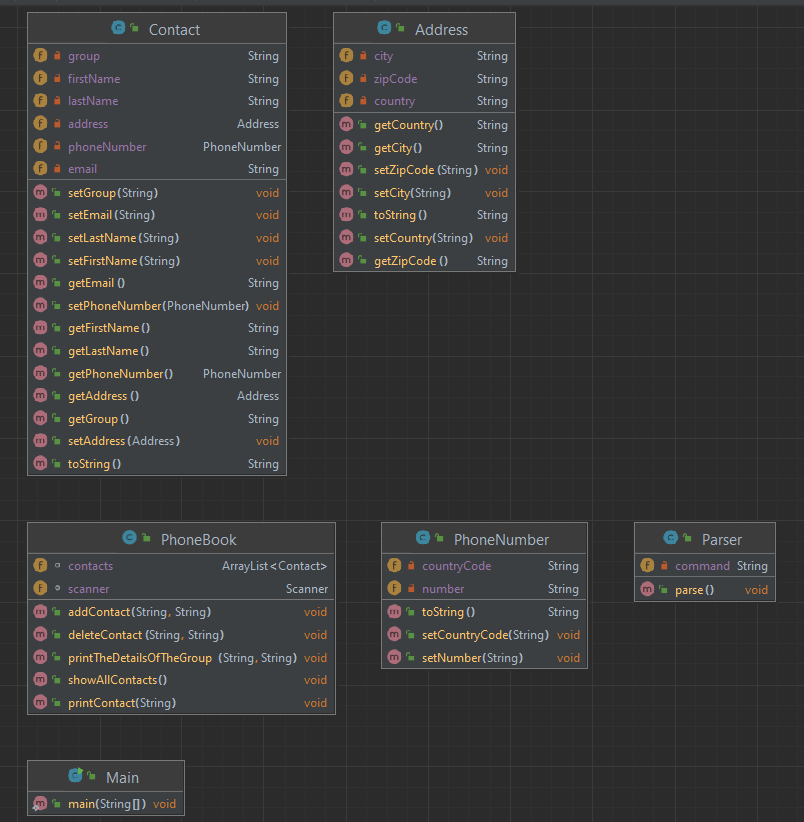


(نمونه‌ای از خروجی دستور اضافه کردن مخاطب)

**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**

