

## 課題20(総合課題)

メールアドレスを含む連絡先を管理するアドレス帳プログラムのため、以下の2つのクラスを作成しましょう。

- **AddressCard**  
名前, 年齢, メールアドレスを含むアドレス帳のデータ  
(アドレスカードと呼称する)を表すクラス
- **AddressBook**  
アドレスカードをまとめたアドレス帳を表すクラス

# 準備

ホームページから `AddressCard.java`,  
`AddressBook.java`, `Exam20.java`をダウンロードします

`AddressCard.java`, `AddressBook.java`は今回作成する  
クラスの未完成バージョンです. 以降の説明に従ってプログラムを  
追記してください.

`Exam20.java`は今回作成したクラスの正しさを検証するための  
テスターです. 2つのクラスが未完成でも実行はできます.

`Exam20.java`をコンパイル, 実行し, 次ページに示す実行結果  
になることを確認してください.

# ここまでの実行結果

さんは0歳でメールアドレスはです。

===== (1) =====

未完成です

===== (2) =====

addressInfoメソッドの動作確認:

addressInfoメソッドの動作確認:

===== (3) =====

山田花子さんのメールアドレスは該当者なしです。

佐藤次郎さんのメールアドレスは該当者なしです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

===== (5) =====

山田を含むアドレスリスト

未完成です

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

Exam20 を実行すると上記の結果となります。各課題作成後、

===== (n) =====

までが一致することを確認してください(nは課題番号)。

# AddressCardクラスのドキュメント

## フィールドの詳細

### name

```
private String name
```

アドレスカードに載っている人の名前

### mailAddr

```
private String mailAddr
```

アドレスカードに載っている人のメールアドレス

### age

```
private int age
```

アドレスカードに載っている人の年齢

## コンストラクタの詳細

### AddressCard

```
public AddressCard(String name,  
                    String mailAddr,  
                    int age)
```

引数で受け取った名前, メールアドレス, 年齢をフィールドにセットする

#### パラメータ:

name - 名前

mailAddr - メールアドレス

age - 年齢

## メソッドの詳細

### getName

```
public String getName()
```

アドレスカードに載っている人の名前を戻すゲッター

#### 戻り値:

アドレスカードに載っている人の名前

### getMailAddr

```
public String getMailAddr()
```

アドレスカードに載っている人のメールアドレスを戻すゲッター

#### 戻り値:

アドレスカードに載っている人のメールアドレス

### getAge

```
public int getAge()
```

アドレスカードに載っている人の年齢を戻すゲッター

#### 戻り値:

アドレスカードに載っている人の年齢

## addressInfo

```
public String addressInfo()
```

このアドレスカードの情報を文字列で戻す。例えば名前が「田中一郎」、アドレスが「tanaka@hoge.com」、年齢が「21」ならば「田中一郎,tanaka@hoge.com,21」という文字列を戻す。なおコンマ(,)は半角文字であり、英字モードで入力されたものである。注意:必ずゲッターを用いて実装すること。ヒント:"abc"という文字列と"def"という文字列を連結したい場合は "abc" + "def"とすればよい(教科書p.92)。例: `String text = "abc"+"def";`

### 戻り値:

アドレスカードの情報を表す文字列

## contains

```
public boolean contains(String name)
```

引数で与えられた文字列が、アドレスカードに載っている人の名前に含まれているかを判定する。含まれていたらtrueを、含まれていなければfalseを戻す。

### パラメータ:

name - アドレスカードに載っている人の名前に含まれているか判定する文字列

### 戻り値:

引数で与えられた文字列が含まれているかの判定結果 (true:含まれる, false:含まれない)

# AddressBookクラスのドキュメント

## フィールドの詳細

### listOfAddressCards

```
private ArrayList<AddressCard> listOfAddressCards
```

このアドレス帳で管理されているアドレスカードのリスト。[AddressCard](#)クラスの[ArrayList](#)で管理する。

## コンストラクタの詳細

### AddressBook

```
public AddressBook()
```

[AddressCard](#)クラスの[ArrayList](#)のフィールド [listOfAddressCards](#)を初期化する。



## メソッドの詳細

### registerCard

```
public void registerCard(AddressCard card)
```

新たなアドレスカードを登録する。すなわち、**AddressCard**クラスの**ArrayList**のフィールド**listOfAddressCards**に、**AddressCard**クラスのオブジェクトを追加する。

#### パラメータ:

card - 新たに登録するアドレスカード

### getList

```
public ArrayList<AddressCard> getList()
```

アドレスカードのリストを戻すゲッター

#### 戻り値:

アドレスカードのリスト

### askMailAddr

```
public String askMailAddr(String name)
```

引数`name`と同じ名前の人のメールアドレスを戻す。もし同姓同名の人がいる場合は、最初に名前が合致した人のメールアドレスを戻す。該当者がいなければ「該当者なし」と戻す。

#### パラメータ:

`name` - メールアドレスを検索したい人の名前

#### 戻り値:

検索結果のメールアドレス

### printAllAddresses

```
public void printAllAddresses()
```

フィールド`listOfAddressCards`に含まれるアドレスカードの情報すべてを表示する。情報とは、アドレスカードを表す`AddressCard`クラスオブジェクトの`addressInfo`メソッドにより作られるものを指す。

## serchByName

```
public ArrayList<String> serchByName(String name)
```

引数`name`が名前に含まれるアドレスカードの情報を探索する。該当するアドレスカードがあれば、各アドレスカードの情報をそれぞれ`String`にして`String型ArrayList`で返す。各アドレスカードの情報とは、各アドレスカードを表す`AddressCard`クラスオブジェクトの`addressInfo`メソッドにより作られるものを指す。該当するアドレスカードがなければ「該当者なし」という`String`1つだけを要素に持つ`String型ArrayList`を返す。

### パラメータ:

`name` - 探索する名前

### 戻り値:

該当するアドレスカードの情報を`String型ArrayList`で返す

## checkSameAddress

```
public boolean checkSameAddress(AddressCard card1,  
                                AddressCard card2)
```

引数で与えられた2つのAddressCardクラスのオブジェクトが、同じ情報を持っているかを判定する。情報とは、アドレスカードを表すAddressCardクラスオブジェクトのaddressInfoメソッドにより作られるものを指す。同じ情報を持っていたらtrueを、そうでなければfalseを返す。

### パラメータ:

card1 - 判定対象のAddressCardクラスのオブジェクト1

card2 - 判定対象のAddressCardクラスのオブジェクト2

### 戻り値:

同じ情報を持っているかの判定結果 (true:持っている, false:持っていない)

## checkDuplicate

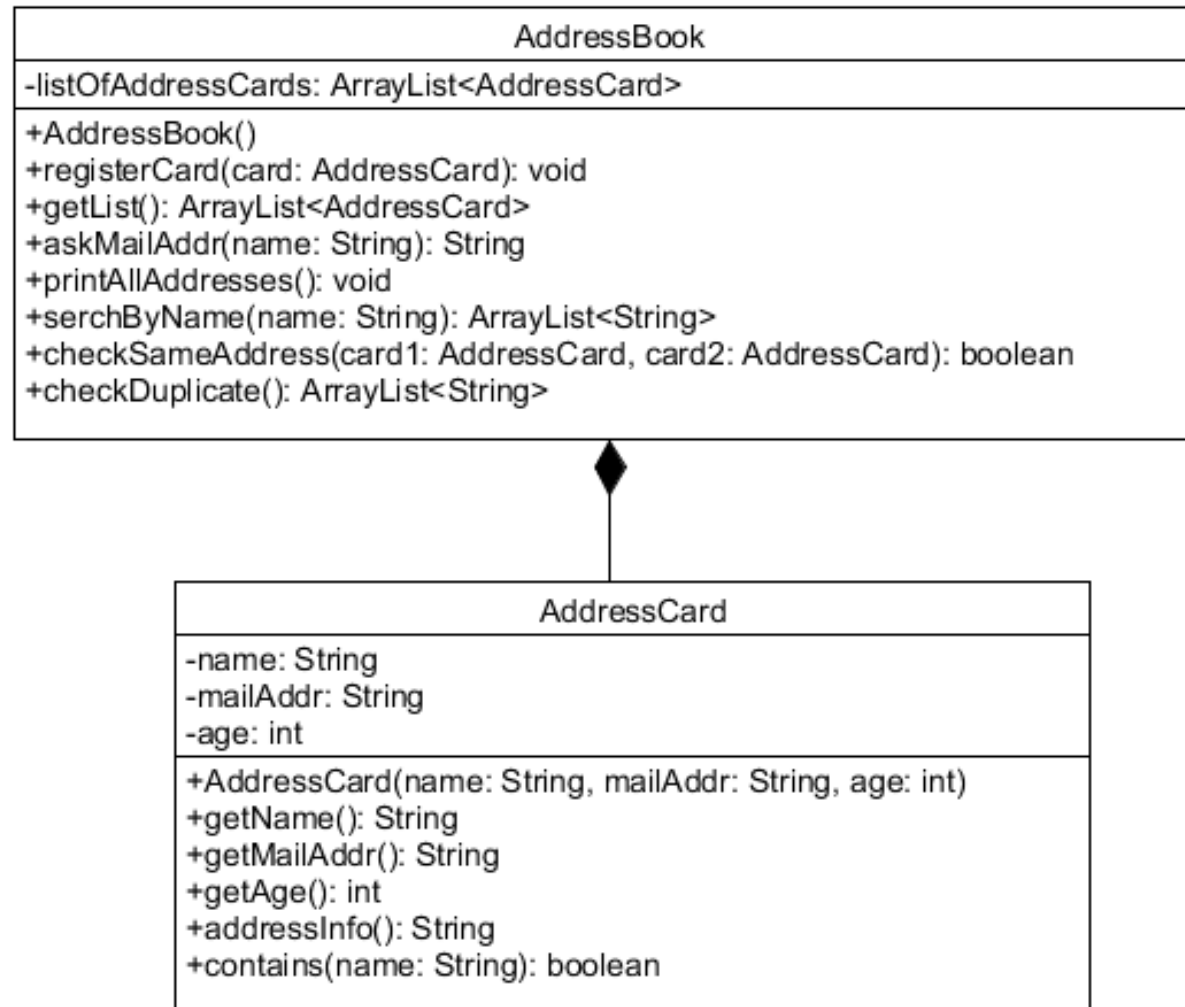
```
public ArrayList<String> checkDuplicate()
```

フィールドlistOfAddressCardsに、同じ情報を持っているアドレスカードがないかを判定する。情報とは、アドレスカードを表すAddressCardクラスオブジェクトのaddressInfoメソッドにより作られるものを指す。同じ情報を持っているアドレスカードがあれば、その情報をStringにしてString型ArrayListで返し(同じ情報をもつカードの情報は複数ArrayListに格納する必要はない)、そうでなければ「重複なし」というString1つだけを要素に持つString型ArrayListを返す。

### 戻り値:

同じ情報を持っているかの結果を示すString型ArrayList

# クラス図



## 課題20-1

AddressCardクラスの説明に従ってコンストラクタと以下の3つのメソッドを完成させましょう。

- `getName()`
- `getMailAddr()`
- `getAge()`

完成したらExam20.javaにより動作確認しましょう。

# ここまでの実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

未完成です

===== (2) =====

addressInfoメソッドの動作確認:

addressInfoメソッドの動作確認:

===== (3) =====

山田花子さんのメールアドレスは該当者なしです。

佐藤次郎さんのメールアドレスは該当者なしです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

===== (5) =====

山田を含むアドレスリスト

未完成です

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

## 課題20-2

AddressBookクラスの説明に従って、以下の2つのメソッドを完成させましょう。

- `registerCard(AddressCard card)`
- `getList()`



# ここまでの実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

山田太郎さんは24歳でメールアドレスはtaro@bar.comです。

===== (2) =====

addressInfoメソッドの動作確認:

addressInfoメソッドの動作確認:

===== (3) =====

山田花子さんのメールアドレスは該当者なしです。

佐藤次郎さんのメールアドレスは該当者なしです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

===== (5) =====

山田を含むアドレスリスト

未完成です

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

## 課題20-3

AddressCardクラスの説明に従って、以下のメソッドを完成させましょう。

- `addressInfo()`

注意：必ずAddressCardクラスのゲッターを用いて実装してください。

# ここまでの実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

山田太郎さんは24歳でメールアドレスはtaro@bar.comです。

===== (2) =====

addressInfoメソッドの動作確認:田中一郎,tanaka@hoge.com,21

addressInfoメソッドの動作確認:山田太郎,taro@bar.com,24

===== (3) =====

山田花子さんのメールアドレスは該当者なしです。

佐藤次郎さんのメールアドレスは該当者なしです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

===== (5) =====

山田を含むアドレスリスト

未完成です

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

## 課題20-4

AddressBookクラスの説明に従って、以下のメソッドを完成させましょう.

- `askMailAddr(String name)`

# ここまでの実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

山田太郎さんは24歳でメールアドレスはtaro@bar.comです。

===== (2) =====

addressInfoメソッドの動作確認:田中一郎,tanaka@hoge.com,21

addressInfoメソッドの動作確認:山田太郎,taro@bar.com,24

===== (3) =====

山田花子さんのメールアドレスはhanako@foo.comです。

佐藤次郎さんのメールアドレスはsato999@kindai.ac.jpです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

===== (5) =====

山田を含むアドレスリスト

未完成です

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

## 課題20-5

AddressBookクラスの説明に従って、以下のメソッドを完成させましょう。

- `printAllAddresses()`

# ここまでの実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

山田太郎さんは24歳でメールアドレスはtaro@bar.comです。

===== (2) =====

addressInfoメソッドの動作確認:田中一郎,tanaka@hoge.com,21

addressInfoメソッドの動作確認:山田太郎,taro@bar.com,24

===== (3) =====

山田花子さんのメールアドレスはhanako@foo.comです。

佐藤次郎さんのメールアドレスはsato999@kindai.ac.jpです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

田中一郎,tanaka@hoge.com,21

山田花子,hanako@foo.com,20

山田太郎,taro@bar.com,24

佐藤次郎,sato999@kindai.ac.jp,30

田中一郎,tanaka@hoge.com,21

===== (5) =====

山田を含むアドレスリスト

未完成です

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

## 課題20-6

AddressCardクラスの説明に従って、以下のメソッドを完成させましょう。

- `contains(String name)`

また、AddressBookクラスの説明に従って、以下のメソッドを完成させましょう。

- `serchByName(String name)`



# ここまでの実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

山田太郎さんは24歳でメールアドレスはtaro@bar.comです。

===== (2) =====

addressInfoメソッドの動作確認:田中一郎,tanaka@hoge.com,21

addressInfoメソッドの動作確認:山田太郎,taro@bar.com,24

===== (3) =====

山田花子さんのメールアドレスはhanako@foo.comです。

佐藤次郎さんのメールアドレスはsato999@kindai.ac.jpです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

田中一郎,tanaka@hoge.com,21

山田花子,hanako@foo.com,20

山田太郎,taro@bar.com,24

佐藤次郎,sato999@kindai.ac.jp,30

田中一郎,tanaka@hoge.com,21

===== (5) =====

山田を含むアドレスリスト

山田花子,hanako@foo.com,20

山田太郎,taro@bar.com,24

中川を含むアドレスリスト

該当者なし

===== (6) =====

重複チェック結果:重複なし

===== (7) =====

## 課題20-7

AddressBookクラスの説明に従って、以下のメソッドを完成させましょう。

- `checkSameAddress(AddressCard card1, AddressCard card2)`
- `checkDuplicate()`

# 最終的な実行結果

田中一郎さんは21歳でメールアドレスはtanaka@hoge.comです。

===== (1) =====

山田太郎さんは24歳でメールアドレスはtaro@bar.comです。

===== (2) =====

addressInfoメソッドの動作確認:田中一郎,tanaka@hoge.com,21

addressInfoメソッドの動作確認:山田太郎,taro@bar.com,24

===== (3) =====

山田花子さんのメールアドレスはhanako@foo.comです。

佐藤次郎さんのメールアドレスはsato999@kindai.ac.jpです。

奈々氏権兵衛さんのメールアドレスは該当者なしです。

===== (4) =====

田中一郎,tanaka@hoge.com,21

山田花子,hanako@foo.com,20

山田太郎,taro@bar.com,24

佐藤次郎,sato999@kindai.ac.jp,30

田中一郎,tanaka@hoge.com,21

===== (5) =====

山田を含むアドレスリスト

山田花子,hanako@foo.com,20

山田太郎,taro@bar.com,24

中川を含むアドレスリスト

該当者なし

===== (6) =====

重複チェック結果:田中一郎,tanaka@hoge.com,21

===== (7) =====