Commencé le	jeudi 7 décembre 2017, 11:11
État	Terminé
Terminé le	mardi 12 décembre 2017, 22:22
Temps mis	5 jours 11 heures
Points	5,00/6,00
Note	**16,67** sur 20,00 (**83**%)

Description

The job is to modify an existing project. The project works, but it can be made better in two ways:

1. make the execution more robust by handling error conditions using exceptions;
2. make the project easier to maintain by eliminating redundancies using enums.

Download the existing code from addressbook_v1t.jar and incorporate it into your favourite development tool.

You'll first have to write an executable class with an appropriate `main` method to run the application. Run the code and look at the source to understand what it's doing.

Note: No for loops have been used in this code; instead it uses Java8 methods, eg, `stream` and `forEach.`

Question 1

Correct

Non noté

This test case indicates the problem with the code.

Enter

```
// KaBoom!
```

into the Answer. You don't have to enter any real code as v1t `AddressBook` and `ContactDetails` are already baked-in and it's those classes which are being tested.

Once the test has failed, as it should, look at the failing method to determine what the problem is. Do not try to fix the problem here, that's the next question...

For example:

Test	Result
`AddressBook book = new AddressBook();` `book.addDetails(null);`	

Réponse:

```
// KaBoom!
```

Vérifier

	Test	
✓	`AddressBook book = new AddressBook();` `book.addDetails(null);`	✓

Passed all tests! ✓

Question 2

Correct

Note de 1,00 sur 1,00

The previous `NullPointerException` at least informed us where there was a problem. But we can do better.

Modify your `AddressBook#addDetails` method to throw an `IllegalArgumentException` with the message "`Null details passed to addDetails.`" whenever a null argument is passed.

Try it out by pasting your `AddressBook` into the Answer. You don't need to paste in `ContactDetails` yet, we're good with the one that's supplied.

Ok, it just replaces one exception with another, but at least here we're starting to control what's going on. Don't try to fix the exception just yet.

For example:

Test	**Result**
`AddressBook book = new AddressBook();` `book.addDetails(null);`	`Null details passed to addDetails.`

Réponse:

```
package addressbook.v1t;

import java.util.LinkedList;
import java.util.List;
import java.util.Set;
import java.util.SortedMap;
import java.util.TreeMap;
import java.util.TreeSet;

/**
 * A class to maintain an arbitrary number of contact details. Details are
 * indexed by both name and phone number.
 *
 * @author David J. Barnes and Michael Kölling.
 * @version 2016.02.29
 */
public class AddressBook {
    // Storage for an arbitrary number of details.
```

Vérifier

	Test	**Expected**	**Got**
✓	`AddressBook book = new AddressBook();` `book.addDetails(null);`	`Null details passed to addDetails.`	`Null details passe`

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 3
Correct
Note de 1,00 sur 1,00

An exception that gets raised is useful for signaling a problem. But it's even better if the exception can be handled somewhere so that the application can keep running.

So let's handle the IllegalArgumentException. But where? Someplace on the call stack...

Veuillez choisir une réponse :

- a. AddressBook#addDetails itself
- b. AddressBookTextInterface#add
- [✓ c. AddressBookTextInterface#run] ✓ Handle the exception here and the application can keep on rolling
- d. AddressBookDemo#showInterface
- e. main

Vérifier

Your answer is correct.

Correct
Note pour cet envoi : 1,00/1,00.

Question 4
Correct
Note de 1,00 sur 1,00

Now you know where to handle it (hint - `AddressBookTextInterface#run`) . Let's do it.

You'll have to paste both `AddressBook` and `AddressBookTextInterface` into the Answer. `ContactDetails, Parser` and `CommandWords` are supplied.

For example:

Test	Input	Resu

`AddressBook book = new AddressBook();`
`System.out.println(book.getNumberOfEntries());`
`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
`interaction.run();`
`System.out.println(book.getNumberOfEntries());`
	help
	add
	Fred Foobar
	555-1234
	There
	quit
		0
		Addı
		Type
		> ac
		> Na
		1

`AddressBook book = new AddressBook();`
`System.out.println(book.getNumberOfEntries());`
`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
`interaction.run();`
`System.out.println(book.getNumberOfEntries());`
	help
	add
	Fred Foobar
	555-1234
	There
	add
	
	Everywhere
	quit
		0
		Addı
		Type
		> ac
		> Na
		> Go
		1

Réponse:

```
package addressbook.v1t;

import java.util.Arrays;

/**
 * Provide a textual interface to an AddressBook. Different commands provide
 * access to the data in the address book.
 *
 * One to search the address book.
 *
 * One to allow a set of contact details to be entered.
 *
 * One to show all the entries in the book.
 *
 * @author David J. Barnes and Michael Kölling.
 * @version 2016.02.29
 */
public class AddressBookTextInterface {
```

Vérifier

	Test	Input	Expe

✓	`AddressBook book = new AddressBook();`
	`System.out.println(book.getNumberOfEntries());`
	`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
	`interaction.run();`
	`System.out.println(book.getNumberOfEntries());`
		help
		add
		Fred Foobar
		555-1234
		There
		quit
			0
			Addr
			Type
			> ad
			> Na
			1

✓	`AddressBook book = new AddressBook();`
	`System.out.println(book.getNumberOfEntries());`
	`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
	`interaction.run();`
	`System.out.println(book.getNumberOfEntries());`
		help
		add
		Fred Foobar
		555-1234
		There
		add
		
		Everywhere
		quit
			0
			Addr
			Type
			> ad
			> Na
			> Go
			1

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Description

Congratulations - you've navigated your way around the `IllegalArgumentException`! That exception, along with a ton of others, are part of the Java API. Unfortunately, there may be situations where the built-in exceptions aren't good enough, and this is one. We need our own exception to describe the situation where you're trying to add a new contact, but you're duplicating an existing key.

Question 5

Correct

Note de 1,00 sur 1,00

Define your `DuplicateKeyException` class based on

```
package addressbook.v1t;

/**
 * Thrown if a key is already in use.
 * @author Peter Sander
 */
@SuppressWarnings("serial")
public class DuplicateKeyException extends ???? {

    /**
     * Constructor. Stores message with superclass.
     * @param messge
     */
        public DuplicateKeyException(String message)

        @Override
        public String toString() {
                return "Key already in use: " + getMessage() + ".";
        }
}
```

The interesting question is of course, "What is the superclass?" Java ortodoxy says `Exception;` experience says `RuntimeException.` Your choice of course.

Paste just your `DuplicateKeyException` into Answer.

Réponse:

```
package addressbook.v1t;

/**
 * Thrown if a key is already in use.
 * @author Peter Sander
 * @author F
 */
@SuppressWarnings("serial")
public class DuplicateKeyException extends RuntimeException {

    /**
     * Constructor. Stores message with superclass.
     * @param messge
     */
    public DuplicateKeyException(String message){

    }

```

Vérifier

	Test	
✓	`Pssst - go with RuntimeException`	✓

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 6

Correct

Note de 1,00 sur 1,00

More work to be done on `AddressBook#addDetails` I'm afraid. The `ContactDetails` passed as argument might already exist in the address book. Verify and raise the appropriate exception if necessary. Oh, and handle any exception so that your application can keep on running.

Submit your `AddressBook`, `AddressBookTextInterface` and `DuplicateKeyException` into Answer.

For example:

Test	Input	Resu

`AddressBook book = new AddressBook();`
`System.out.println(book.getNumberOfEntries());`
`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
`interaction.run();`
`System.out.println(book.getNumberOfEntries());`
	help
	add
	Fred Foobar
	555-1234
	There
	quit
		0
		Addı
		Type
		> ac
		> Na
		1

Réponse:

```
package addressbook.v1t;

import java.util.LinkedList;
import java.util.List;
import java.util.Set;
import java.util.SortedMap;
import java.util.TreeMap;
import java.util.TreeSet;

/**
 * A class to maintain an arbitrary number of contact details. Details are
 * indexed by both name and phone number.
 *
 * @author David J. Barnes and Michael Kölling.
 * @version 2016.02.29
 */
public class AddressBook {
    // Storage for an arbitrary number of details.
```

Vérifier

	Test	Input	Expe

✓	`AddressBook book = new AddressBook();`
	`System.out.println(book.getNumberOfEntries());`
	`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
	`interaction.run();`
	`System.out.println(book.getNumberOfEntries());`
		help
		add
		Fred Foobar
		555-1234
		There
		quit
			0
			Addr
			Type
			> ad
			> Na
			1

✓	`AddressBook book = new AddressBook();`
	`System.out.println(book.getNumberOfEntries());`
	`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
	`interaction.run();`
	`System.out.println(book.getNumberOfEntries());`
		help
		add
		Fred Foobar
		555-1234
		There
		add
		Fred Foobar
		555-2345
		Everywhere
		quit
			0
			Addr
			Type
			> ad
			> Na
			> Go
			1

Passed all tests! ✓

Correct

Note pour cet envoi : 1,00/1,00.

Question 7

Incorrect

Note de 0,00 sur 1,00

One more exception class to define - `NoMatchingDetailsException`. Sort of like `DuplicateKeysException`, but this time make it

```
public class NoMatchingDetailsException extends Exception {
    // stuff omitted

    @Override
    public String toString() {
        return "No details matching: " + getMessage() + " were found.";
    }
}
```

After this, you'll appreciate `RuntimeException`!

Modify your `AddressBook` class as appropriate to use `NoMatchingDetailsException` whenever a method argument might cause difficulty.

Oh, and while you're at it - add two new methods to `AddressBook`: `get` and `remove`

```
    /**
     * Find an entry matching a key.
     */
    private void get() {
        System.out.println("Type the key of the entry.");
        // code to supply
        System.out.println(result);
    }

    /**
     * Remove an entry matching a key.
     */
    private void remove() {
        System.out.println("Type the key of the entry.");
        // code to supply
    }
```

You'll need to modify AddressBookTextInterface and CommandWords. The change to the latter class would be

```
private static final String validCommands[] = {"add", "get", "search",
            "list", "remove", "help", "quit",};
```

Submit everything except `ContactDetails` to Answer.

For example:

Test	Input	Resu

AddressBook book = new AddressBook();
System.out.println(book.getNumberOfEntries());
AddressBookTextInterface interaction = new AddressBookTextInterface(book);
interaction.run();
System.out.println(book.getNumberOfEntries());
	help
	add
	Fred Foobar
	555-1234
	There
	quit
		0
		Addı
		Type
		> ac
		> Na
		1

Réponse:

```
package addressbook.v1t;

import java.util.LinkedList;
import java.util.List;
import java.util.Set;
import java.util.SortedMap;
import java.util.TreeMap;
import java.util.TreeSet;

/**
 * A class to maintain an arbitrary number of contact details. Details are
 * indexed by both name and phone number.
 *
 * @author David J. Barnes and Michael Kölling.
 * @version 2016.02.29
 */
public class AddressBook {
    // Storage for an arbitrary number of details.
```

Vérifier

	Test	Input

✓	`AddressBook book = new AddressBook();`
	`System.out.println(book.getNumberOfEntries());`
	`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
	`interaction.run();`
	`System.out.println(book.getNumberOfEntries());`
		`help`
		`add`
		`Fred Foobar`
		`555-1234`
		`There`
		`quit`

✗	`AddressBook book = new AddressBook();`
	`System.out.println(book.getNumberOfEntries());`
	`AddressBookTextInterface interaction = new AddressBookTextInterface(book);`
	`interaction.run();`
	`System.out.println(book.getNumberOfEntries());`
		`help`
		`add`
		`Fred Foobar`
		`555-1234`
		`There`
		`get`
		`555-1234`
		`get`
		`J. Random Luser`
		`get`
		`555-0000`
		`quit`

Your code must pass all tests to earn any marks. Try again.

Incorrect

Note pour cet envoi : 0,00/1,00.