

Mini Project

Task 1

Write a console base shopping cart application to allow a user to add, remove, list, save and load their shopping cart.

The following is an example session of the shopping cart application in action

```
java -cp shoppingcart.jar vttp2022.cart.Main shoppingcart
Using shoppingcart directory for persistence
There are 10 carts in shoppingcart directory
> load fred
fred shopping cart loaded
> list
1. apple
> add orange grapes
orange, grapes added to the cart
> list
1. apple
2. orange
3. grapes
> delete 1
apple removed from cart
list
1. orange
2. grapes
> save
cart contents saved to fred
> exit
```

Implement the following commands

- `list` - list the contents of the cart. If the cart is empty print an appropriate message
- `add` - add one or more items to the cart. Multiple items are separated by a single white space
- `delete` - delete an item using the item's index (from the list command). If the index is incorrect, print an appropriate message

- `load` - load a user's cart from a file. The file name is the user's name; for example `load fred` will load the contents of the shopping cart from the file `fred.cart` from the shopping cart (see below) directory. If the file does not exist then the initial shopping cart will be empty.
- `save` - save the contents of the shopping cart to a file in the shopping cart directory appending the file name with the suffix `.cart`; eg. if the user's name is fred, then the file name should be `fred.cart`. If a cart file with the same name exists in the shopping cart directory, overwrite the file
- `exit` - exit the shopping cart application

The shopping cart files are saved in a shopping cart directory. This directory is passed as a parameter to the shopping cart application when the application is started

```
java -cp shoppingcart.jar vtttp2022.cart.Main shoppingcart
```

where `shoppingcart` is the shopping cart directory. Shopping carts should be loaded and saved from and to this directory.

See Java documentation for `java.io.File` class for reading directories.

Task 2

Modify Task 1 to a client/server shopping cart application. The shopping cart application should behave the same as specified in Task 1.

The following is an example session of the client/server shopping cart application in action

Starting the server

```
java -cp ./shoppingcart.jar vtttp2022.cart.server.Main
shoppingcart 3000
Starting shopping cart server on port 3000
Using shoppingcart directory for persistence
Connection received...
```

The server takes 2 parameters:

1. shopping cart directory
2. port the server listens to

Starting the client

```
java -cp ./shoppingcart.jar vtftp2022.cart.client.Main fred@localhost:3000
```

```
Connected to shopping cart server at localhost on fred port 3000
```

```
fred shopping cart loaded
```

```
> list
```

```
1. apple
```

```
> add orange grapes
```

```
orange, grapes added to the cart
```

```
> list
```

```
1. apple
```

```
2. orange
```

```
3. grapes
```

```
> delete 1
```

```
apple removed from cart
```

```
list
```

```
1. orange
```

```
2. grapes
```

```
> save
```

```
cart contents saved to fred
```

```
> exit
```

The client application takes a single parameter which is the connection string. The connection string has the following format

`<user>@<host>:<port>`

For example `user@localhost:3000` means connect to the shopping cart server a localhost listening on port 3000. Load fred's shopping cart.

Task 3

Convert the shopping cart server in Task 2 to a multi threaded server. You can ignore race conditions eg. saving to the same shopping cart file by 2 or more clients.