

UKM Parking Program

By Shameer Ali – A173586

Group 13

\*\*use the pdf version if the word file seem not organized\*\*

Introduction

Since UKM have many parking spaces and also each year new student brings in new vehicle which makes the parking in the whole UKM not organised. So students and staffs don’t get to park their vehicle in some places due to this. And the staffs can’t keep track of parking’s in UKM.

This program is intended to keep track of vehicle and parking in the UKM. Students and staff can use this to select and reserve parking. They can also see if a parking is full or not so that they can quickly reserve a spot in the parking before going there or go to another parking if it’s full.

How To Use the Software,  
  
GUI Design,  
&  
  
File I/O Format

Mohamed Shameer Ali bin A.M Abdul Rahman

A173586

-run the UKMParking.Java file

\*\*The GUI you see in here won’t be 100% the same as the final program. It will have slight colour change and buttons design that’s all\*\*

Contents

[**Files to Store Data** 4](#_Toc39012576)

[The person file need to be written in : 4](#_Toc39012577)

[The parking file need to be written in : 4](#_Toc39012578)

[**Starting The Program** 5](#_Toc39012579)

[**Select From File** 5](#_Toc39012580)

[**Start New** 5](#_Toc39012581)

[**Parking Lots** 6](#_Toc39012582)

[**Parked List** 7](#_Toc39012583)

[**Create New** 7](#_Toc39012584)

[**Change Job** 8](#_Toc39012585)

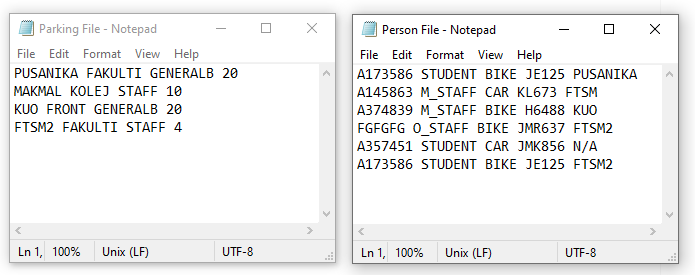
[**Change Vehicle** 9](#_Toc39012586)

[**Features of the program** 9](#_Toc39012587)

[**Flaw and Constraints** 10](#_Toc39012588)

[**Advantage and Feature** 10](#_Toc39012589)

# **Files to Store Data**



1. If you choose to start from a list of data that you have you can follow this step or else you can start a new file from page 4 at “Start New” section.
2. The image above is how the data need to be stored in the files.

## The person file need to be written in :

**Person\_ID < > Person\_Job < > Vehicle\_Type < > Vehicle\_Plat\_No < > Parked\_At**

**Person\_ID :** ID of the person.

**Person\_Job:** Either “STUDENT”, “0\_STAFF” (stands for ordinary staff), “M\_STAFF” **(**stand for management staff).

**Vehicle\_Type:** Either “Car”, “BIKE”, “LORRY”, “BUS”.

**Vehicle\_Plat\_No:** The Plat Numbur of that vehicle.

**Parked\_At:** The name of the parking area the vehicle is parked or else write “N/A” if none.

## The parking file need to be written in :

**Parking\_name < > Parking\_Area < > Parking\_Type < > Parking\_space**

**Parking\_name** **:** The name of the parking. Can be what ever you want. Person file will refer this. You don’t need to make parkings for all the person in person file. You can do that in the program later if you want.

**Parking\_Area :** The area at UKM. Eg: KOLEJ, FAKULTI, PUBLIC.

**Parking Type:** Either “SATFF”, “GENERALB”, “GENERALC”, “MANAGEMENT”.

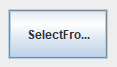
**Parking\_Space:** The amount of space in that parking lot.

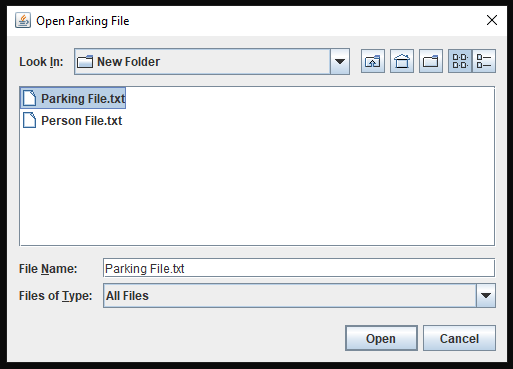
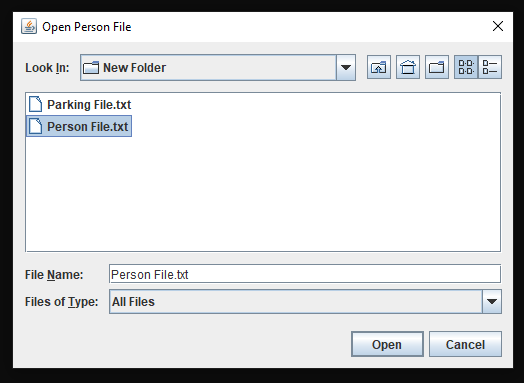
Each word is separated by a <space>.

1. Name the file anything you want and just store the file anywhere in you computer.

# **Starting The Program**

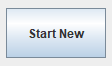
## **Select From File**

1. Start the program and click “Select From File” Button if you have a file to open.  
    
2. See the tittle on the File Chooser pop up.
3. Based on the image below select the file stored the person information on the “Open Person File” File\_Chooser and click “Open”. Select the file stored parking information on the “Open Parking File” File\_Chooser and click “Open”.

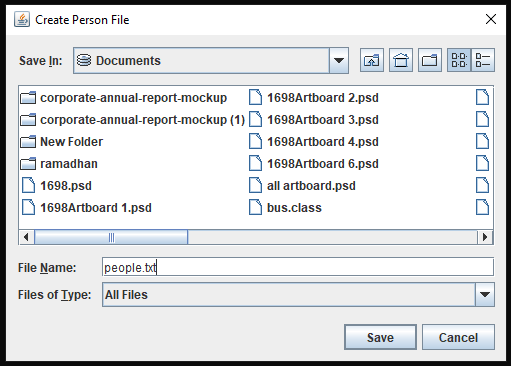
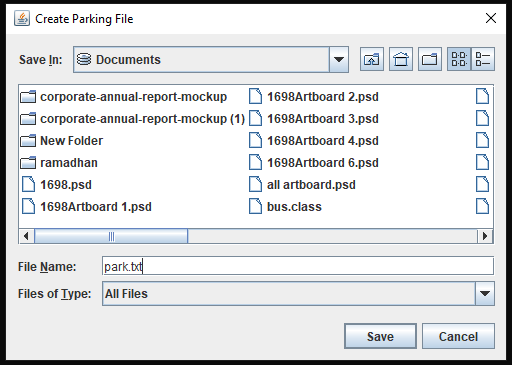


## **Start New**

1. If you don’t have any file to choose from, then juts click the “Start New” Button.



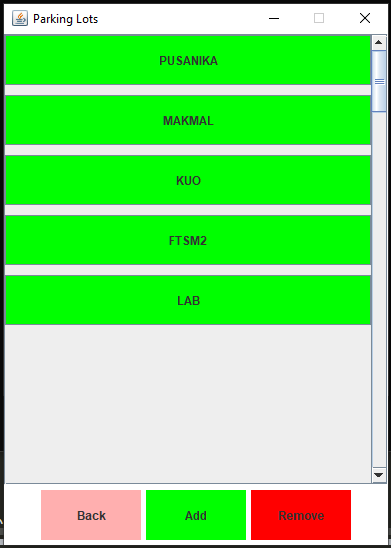
1. See the tittle on the pop up.
2. Based on the image below, type in the name of the person file you want to make at the “Create Person File” pop up and click “Save” to make a new file to store person informations. Then Type in the name of the parking file you want to make at the “Create Parking File pop up and click “Save” to make a new file to store parking informations.

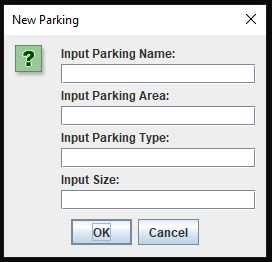


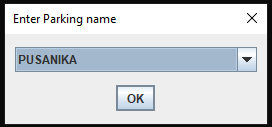
# **Parking Lots**



The “Parking Lots” Button will bring you to the list of parking names. The parking names is in button so you can click it and see who’s in that parking lot.

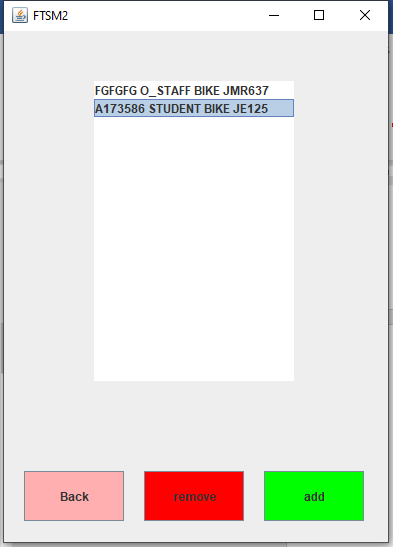
Based on the image on the left, if you use “Select From File” you Gui will look something like that. All the parking you created is a button that you can click. But if you use “Start New” you will still have the same Gui but without the parking names buttons.

If you click “add” a pop up will appear where you can make a new parking space. Refer to page 2 at the “Parking files need to be written in:“ section for guidance to fill these up. The parking space you created will search the person file to see if there is anyone at the new parking you have created. That’s why you don’t need to make all the parking in the parking file and can just do it in the Gui.

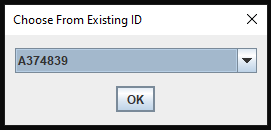


By clicking “remove” you can select the parking space you want to remove from your data from the drop down menu.

# **Parked List**



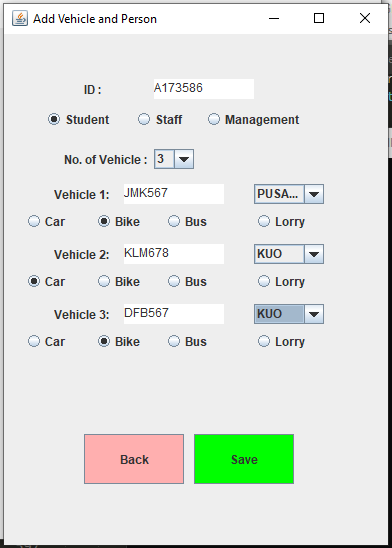
By clicking one of the parking spaces, it will open a frame containing the list of people in that parking lot. To edit the data, you can click on the data in the list like I clicked the A173586 person. Then you can choose “remove” to clear that person from the parking lot. Note that once you click remove the person Parked\_At data will be set to “N/A” and be removed from that current list.

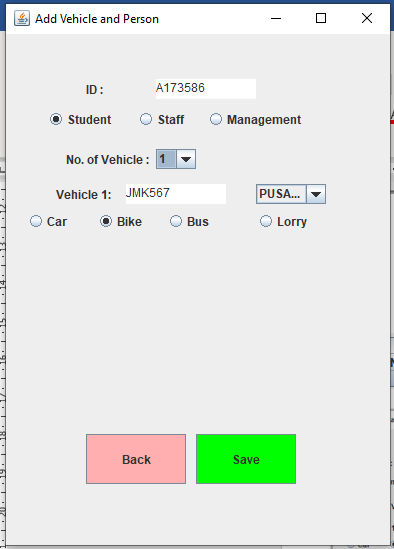


If you choose add, a new pop up will appear where you can choose a person to add to that parking lot. The pop up have a drop down menu containing the ID of person that have been created or from the list.

# **Create New**

1. This is where you can create new id, vehicle and assign parking space to that person.

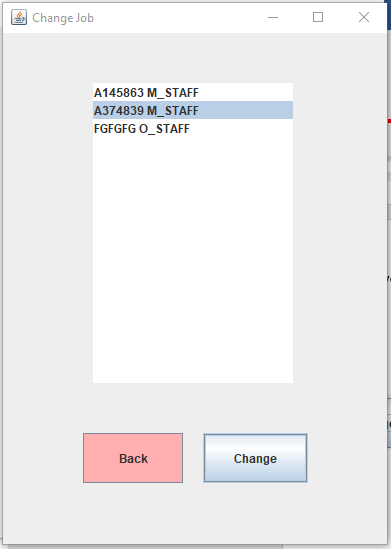
2. You need to enter the ID and select the job so that the parking spaces available for that job can be selected.

3. You can choose the amount of vehicle that the ID want to register.  


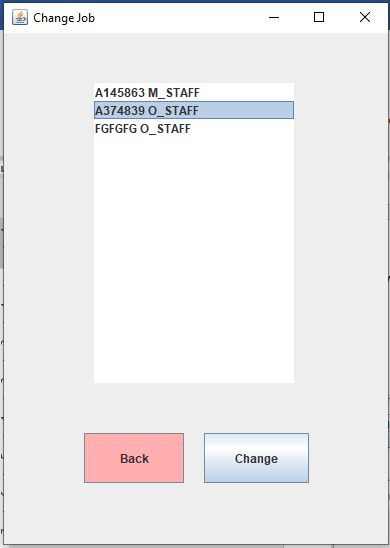
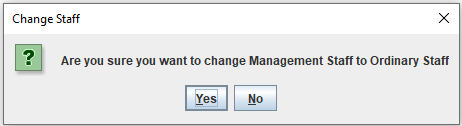
4. After you have done. Just click save and your new data will be saved.

# **Change Job**

Here you can change appoint management staff or end management staff.

  
1. Just click the person you want to change their job status and then click change.

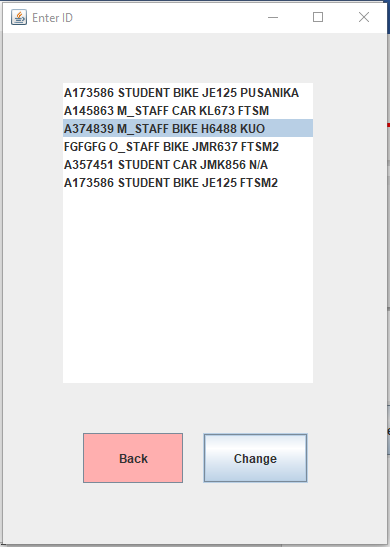
2. A pop up will appear asking you if you wish to change their job from “…” to “…”.

3. Click “Yes” if you wish to continue.  
3. It will refresh and show you the change.

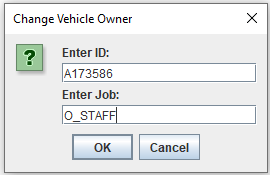
# **Change Vehicle**



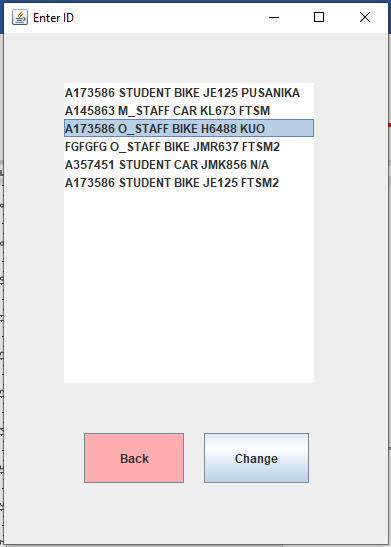
This is where you can change the vehicle owner that has been assigned before to a new owner.

1. Just click the vehicle you want to change the owner of and then click change.

2. A pop up will appear asking you the new owner ID and their job. The job should only be either “STUDENT”, “O\_STAFF” or “M\_STAFF”.

3. Click ok after you have filled up.

4. The list will refresh showing the new assigned owner.



# **Features of the program**

1. The program will save and write the data in the file anytime the user alter or add any data.
2. The program will automatically turn on the user “Caps Lock” on their keyboard when they run the program. They can see the “Caps lock” light light-up. The user can just close the caps lock if they want and the program will work just fine.

# **Flaw and Constraints**

1. Maybe I could have made the coding shorter by using the full feature of all the classes.
2. The textfield in the optionpane could be turned into radiobutton to make it easier for users to input data.
3. The frame cannot be resized due to the null layout of the frame that will make the component messed up when resized.
4. Forgot to add JScrollPane in the JList to make the JList scroll able. Now the Jlist can only display data inside it’s size.

# **Advantage and Feature**

1. Users can create as many parking spaces they want.
2. Any alteration of data will write the file automatically.

~If there’s any problem, please contact me at “01140448922” or email me at [a173586@siswa.ukm.edu.my](mailto:a173586@siswa.ukm.edu.my) . As far as I code & tested, this program is complete and has all the functions needed and also efficient~



List of Class

&

Explaination

Mohamed Shameer Ali Bin A.M Abdul Rahman

A173586

`Run the “UKMParking.java” to use the program`

\*\*use the pdf version if the word file seem not organized\*\*

Contents

[Class: person 13](#_Toc38563074)

[Class: staff 13](#_Toc38563075)

[Class: students 13](#_Toc38563076)

[Class: vehicle 14](#_Toc38563077)

[Class: car 14](#_Toc38563078)

[Class: motorcycle 14](#_Toc38563079)

[Class: bus 14](#_Toc38563080)

[Class: lorry 14](#_Toc38563081)

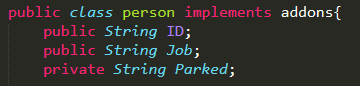
[Class: parking 15](#_Toc38563082)

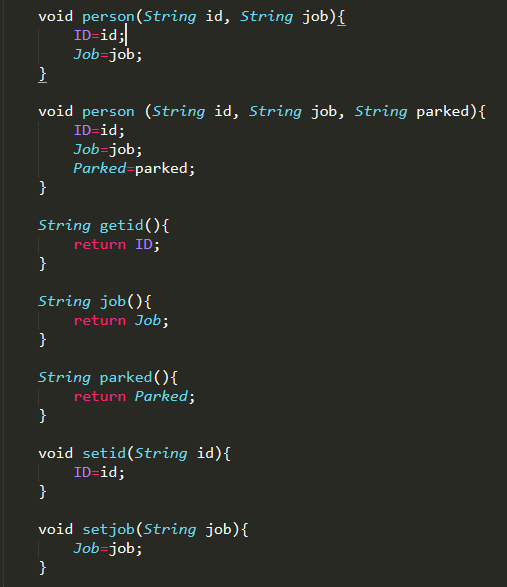
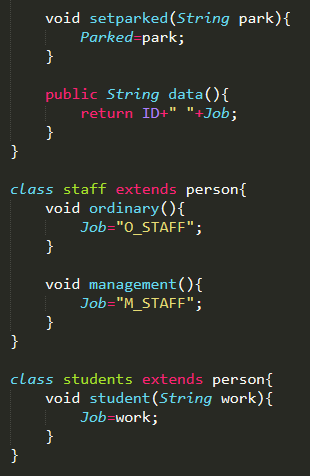
[Class : UKMParking 16](#_Toc38563083)

[Interface: Addons 19](#_Toc38563084)

Class: person,

Class: staff, Class: students

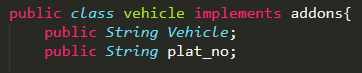


This is the person class where it contains the information of every person who has their data. It implements addons interface.

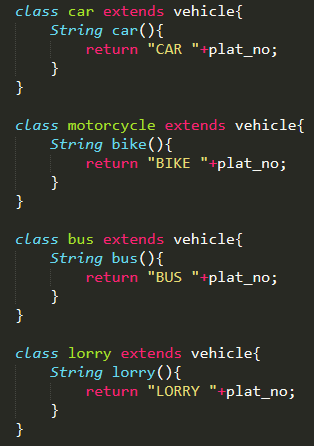
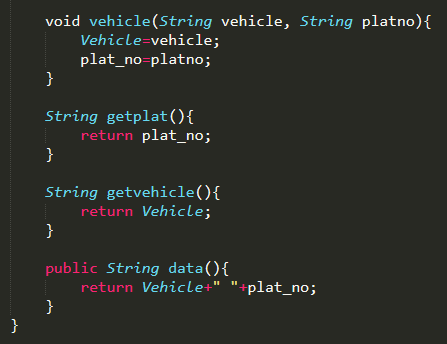
This person has 2 constructors, person(id, job) & person(id, job, parked) with different parameters each. And it also contains set setid(), setjob(), setparked() and get getid(), job(), parked() methods if the data needed to be altered or transferred.   
  
The staff and students class is a child class of person class. It inherits all attributed from person class. Staff class store the information for staffs and students class is the same for student information. The ordinary(), management() & student() methods in the two classes are used to change their job when needed.

Class: vehicle,

Class: car, Class: motorcycle, Class: bus, Class: lorry



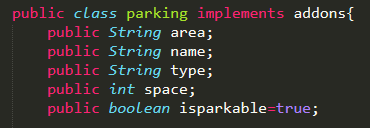
This is the vehicle class where it contains the information of every vehicle and it’s data. It implements addons interface.



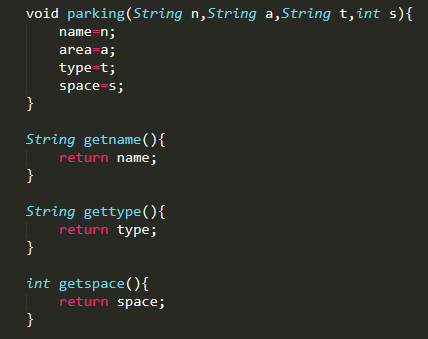
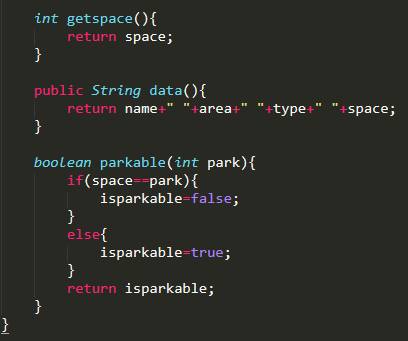
The vehicle class has 1 constructor vehicle() with 2 parameter. The class has its get getplat(), getvehicle(), data() methods and if any data need to be altered it can be done through the constructor.

The other four classes car, motorcycle, bus and lorry is a child class of vehicle and is used to stored their own vehicle if any user owns it. Each classes have their own method car(), bike(), bus(), lorry() that is used to get data.

# Class: parking



This class handles all the information regarding parking area. It stores the data of parking spaces. It implements addons interface.

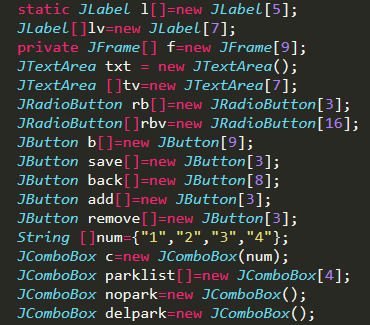
The parking class has one constructor parking() with four parameter. It has its own get getname(), gettype(), getspace(), data() methods to transfer data.

It also has a parkable parkable() methods that keeps record if a specific parking is full or not and if it can be parked.

# Class : UKMParking



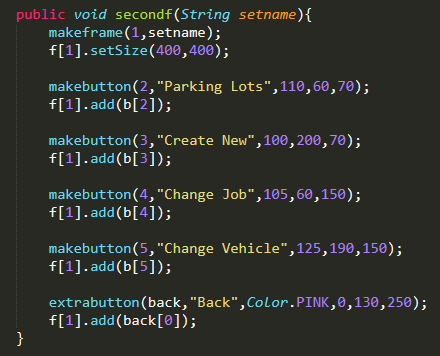
This class is the main class that contains the GUI. This is the class the user should run to use the program. It extends JFrame and implements ActionListener



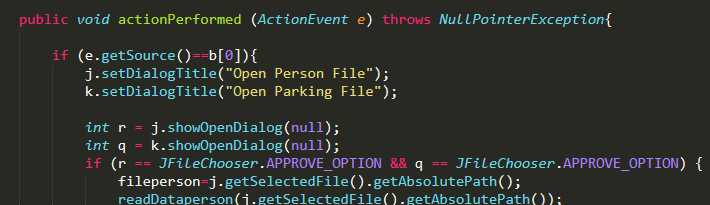
The buttons, frame, radiobuttons, textarea, lables and other components are created in array to minimize codes.



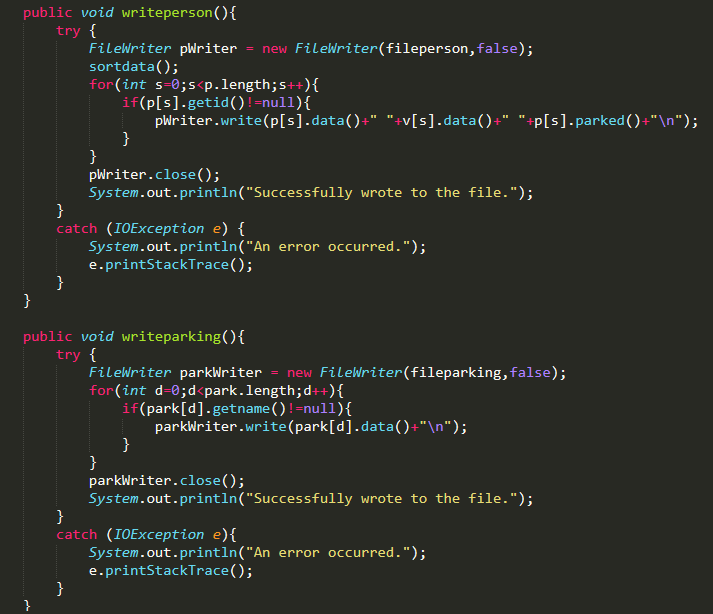
The makeframe(), makebutton() maketext() are methods to create any JFrame, JButtons, JTextArea and it can just be called with the parameter needed to make those components.



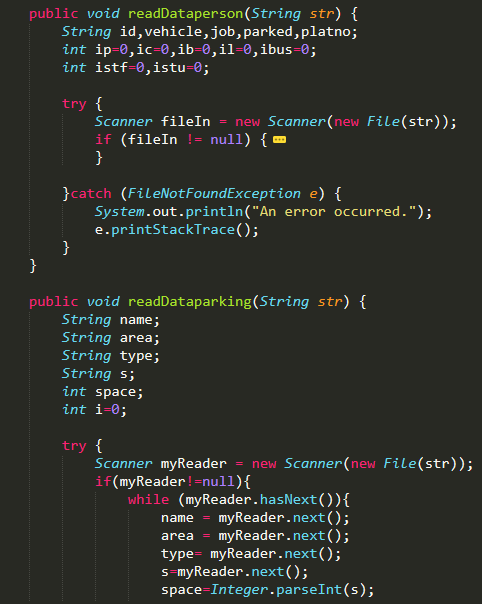
The firstf(), secondf(), thirdf(), forthf(), fifthf(), sixthf(), seventh(), all the method that end with f are methods that makes the frame. For example, based on the picture o the left, it’s a method to make the second frame containing all it’s components inside the frame.



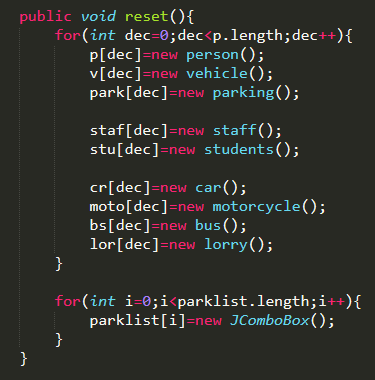
This is the actionPerformed method. This is the method where all the JButtons, JRadiobutton, JList actions take place. It’s what makes the GUI work completely.



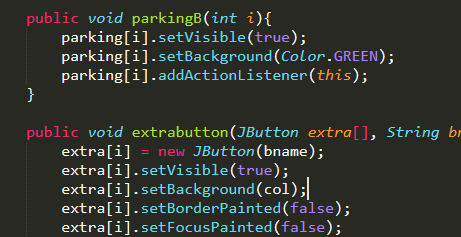
The writeperson() and writeparking() methods are used to write data into the person and parking file. Each time a data is added or altered, this method is called.



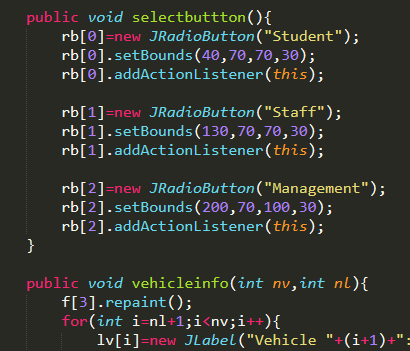
The readDataperson() and readDataparking() method is called when a file is opened at the starting of the program. It reads the file n the directory the file has been chosen.

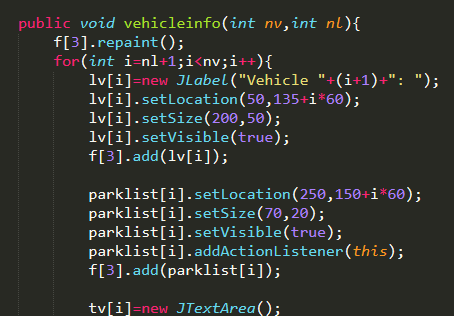


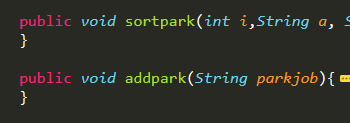
This is the method that creates or make the array ‘null’ so that if the user wishes to start new all the classes array will be null but it won’t effect any files.

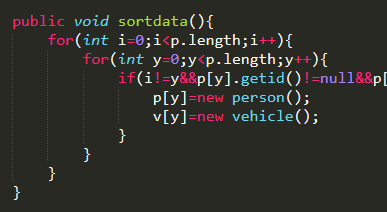
parkingB() method is used to make new parking buttons. Where when user add a new parking the button will be made using the parking. 

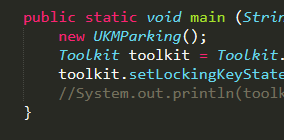
The extrabutton() method is used to make back, add, remove button. So if one of those button is created and added into frame, it will use this method.

selectbuttons() method creates the radiobutton for the create new frame.

The vehicleinfo() method handles the addition and removal of textarea, radiobutton and combobox according to the numbers of vehicle selected and the combobox.

sortpark() and addpark() method handles the addition and removal of strings from comboboxes.

sortdata() method look for any repetitions in data inside the person and vehicle class array and then clears the data to prevent duplication of data when writing.

The last one is the main() method. It starts the program and also turn on the capslock button on the users keyboard.

# Interface: Addons

This is the interface file used in person, vehicle, and parking java file.

That’s all the classes and methods explained. If you want to run the program, you can run the UKMParking.java file.