Assignment 1

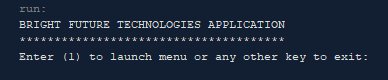
PROG6112

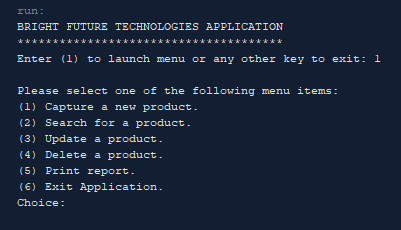
Miguel Almeida

ST10025374

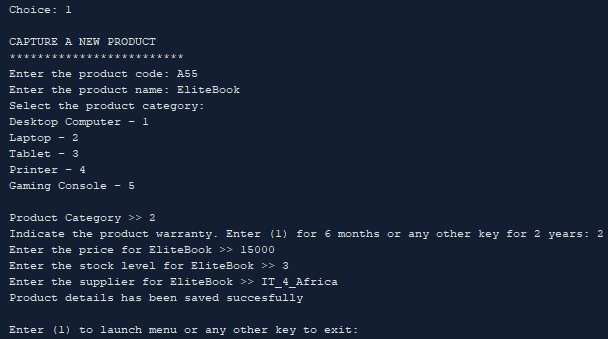
# Question 1

## Q.1.1

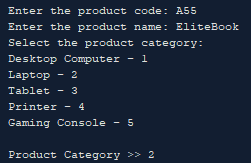




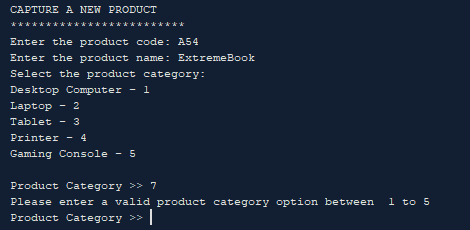
## Q.1.2

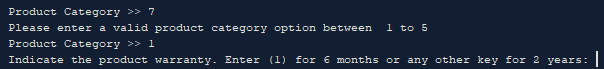


## Q.1.4



## Q.1.5

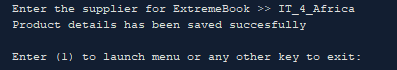




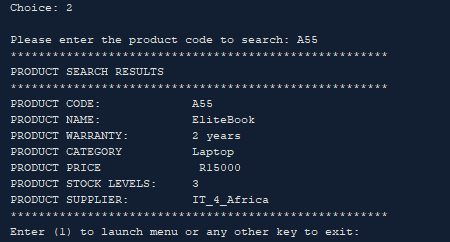
## Q.1.6

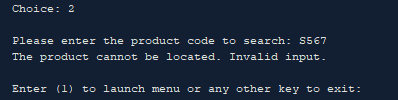


## Q.1.7

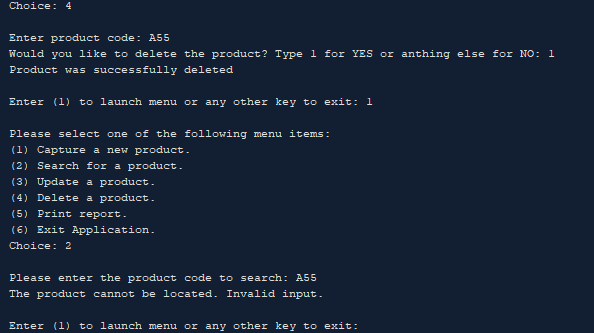


## Q.1.8

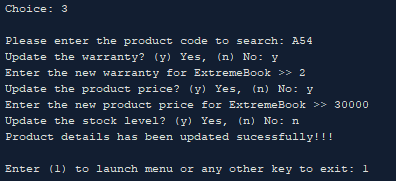




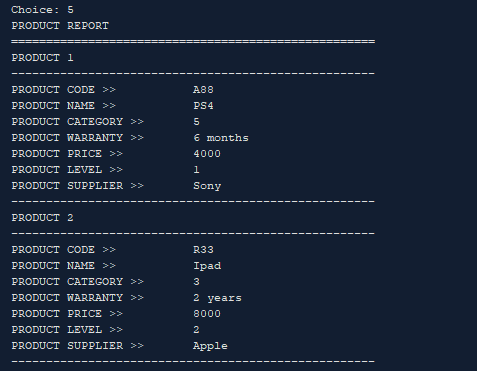
## Q.1.9

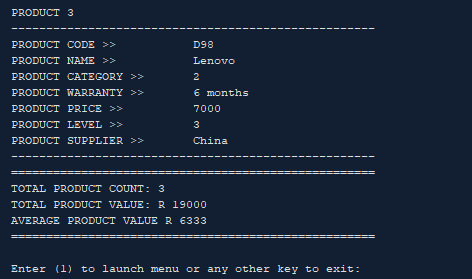


## Q.1.10

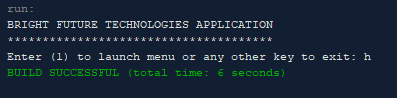


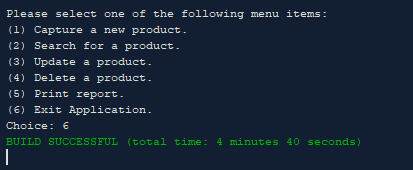
## Q.1.11





## Q.1.12





Code

/\*

Program Description:

Java Application to help manage Hardware Devices Supplier Business

\*/

package assigment\_question\_1;

import java.util.Scanner;

public class Assigment\_Question\_1 {

public static void main(String[] args) {

Products methods = new Products();

Scanner input = new Scanner(System.in);

//Print Program launcher

System.out.print("BRIGHT FUTURE TECHNOLOGIES APPLICATION\n"

+ "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

+ "Enter (1) to launch menu or any other key to exit: ");

//Get user Input

String userInput = input.next();

//If user selects 1 call menu

if (userInput.matches("1"))

{

do

{//Display menu for user to choose an option

methods.DisplayMenu();

userInput = input.next();//Get user choice

switch(userInput){

case "1"://(1) Capture a new product

methods.captureProduct();

break;

case "2"://(2) Search for a product

methods.searchProduct();

break;

case "3"://(3) Update Product

methods.updateProduct();

break;

case "4"://(4) Delete a product

methods.deleteProduct();

break;

case "5"://(5) Print Report

methods.printReport();

break;

case "6"://(6) Exit Application

methods.exitApplication();

break;

}

//Ask user to menu or leave program

System.out.print("\nEnter (1) to launch menu"

+ " or any other key to exit: ");

//Get user choice if 1 keep loop running

userInput = input.next();

} while(userInput.matches("1"));

}

else //If user selects anything beside 1 call exit program method

{

methods.exitApplication();

}

}

}//==========================================================================END

/\*

Class contains products info, all gets and sets

\*/

package assigment\_question\_1;

public class ReportData {

private String productCode;

private String productName;

private int productCategory;

private String productWarranty;

private int productPrice;

private int productStockLevel;

private String productSupplier;

public ReportData() {

}

/\*\*

\* @return the productCode

\*/

public String getProductCode() {

return productCode;

}

/\*\*

\* @param productCode the productCode to set

\*/

public void setProductCode(String productCode) {

this.productCode = productCode;

}

/\*\*

\* @return the productName

\*/

public String getProductName() {

return productName;

}

/\*\*

\* @param productName the productName to set

\*/

public void setProductName(String productName) {

this.productName = productName;

}

/\*\*

\* @return product category matched from int to string

\*/

public String getProductCategoryString() {

String productCategoryString = "";//Store Product Category

//Match int with correct product category

switch (productCategory) {

case 1:

productCategoryString = "Desktop Computer";

break;

case 2:

productCategoryString = "Laptop";

break;

case 3:

productCategoryString = "Tablet";

break;

case 4:

productCategoryString = "Printer";

break;

case 5:

productCategoryString = "Gaming Console";

break;

default:

break;

}

return productCategoryString;

}

/\*\*

\* @return the productCategory

\*/

public int getProductCategory() {

return productCategory;

}

/\*\*

\* @param productCategory the productCategory to set

\*/

public void setProductCategory(int productCategory) {

this.productCategory = productCategory;

}

/\*\*

\* @return determine the productWarranty time based

\* on userInput and return it

\*/

public String getProductWarranty() {

String warranty = "";

//If user inputs 1 return 6 months ortherwise return 2 years

if (productWarranty.equals("1"))

{

warranty = "6 months";

} else {

warranty = "2 years";

}

return warranty;

}

/\*\*

\* @param productWarranty the productWarranty to set

\*/

public void setProductWarranty(String productWarranty) {

this.productWarranty = productWarranty;

}

/\*\*

\* @return the productPrice

\*/

public int getProductPrice() {

return productPrice;

}

/\*\*

\* @param productPrice the productPrice to set

\*/

public void setProductPrice(int productPrice) {

this.productPrice = productPrice;

}

/\*\*

\* @return the productStockLevel

\*/

public int getProductStockLevel() {

return productStockLevel;

}

/\*\*

\* @param productStockLevel the productStockLevel to set

\*/

public void setProductStockLevel(int productStockLevel) {

this.productStockLevel = productStockLevel;

}

/\*\*

\* @return the productSupplier

\*/

public String getProductSupplier() {

return productSupplier;

}

/\*\*

\* @param productSupplier the productSupplier to set

\*/

public void setProductSupplier(String productSupplier) {

this.productSupplier = productSupplier;

}

}//==========================================================================END

/\*

Class contains all your

working methods

\*/

package assigment\_question\_1;

import static java.lang.Integer.parseInt;

import java.util.ArrayList;

import java.util.Scanner;

public class Products {

Scanner input = new Scanner(System.in);

private ReportData productInfo = new ReportData();

private ArrayList<ReportData> MyList = new ArrayList<ReportData>();

//--------------------------------------------------------------------------

//Method to Display Menu

public static void DisplayMenu() {

//Print Menu

System.out.print("\nPlease select one of the following menu items:\n"

+ "(1) Capture a new product.\n"

+ "(2) Search for a product.\n"

+ "(3) Update a product.\n"

+ "(4) Delete a product.\n"

+ "(5) Print report.\n"

+ "(6) Exit Application.\n"

+ "Choice: ");

}//end of Method------------------------------------------------------------

//Method to capture product details

public void captureProduct() {

this.productInfo = new ReportData();//Allocating memory

System.out.println("\nCAPTURE A NEW PRODUCT\n"

+ "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

//PRODUCT CODE

System.out.print("Enter the product code: ");

this.productInfo.setProductCode(input.next());

//PRODUCT NAME

System.out.print("Enter the product name: ");

this.productInfo.setProductName(input.next());

//PRODUCT CATEGORY

System.out.print("Select the product category: "

+ "\nDesktop Computer - 1"

+ "\nLaptop - 2"

+ "\nTablet - 3"

+ "\nPrinter - 4"

+ "\nGaming Console - 5\n\n");

boolean valid = false;

do{

System.out.print("Product Category >> ");

this.productInfo.setProductCategory(input.nextInt());

//Check if product is between 1 and 5

//and if not make it invalid and inform user

//and ask for a valid product category

if((this.productInfo.getProductCategory() < 1) || (this.productInfo.getProductCategory() > 5))

{

System.out.println("Please enter a valid product category option between 1 to 5");

}

else {//If selection is between 1 and 5 make variable true to exit the loop

valid = true;

}

} while(!valid);//If valid becomes true loop will be EXIT

//PRODUCT WARRANTY

System.out.print("Indicate the product warranty. Enter (1) "

+ "for 6 months or any other key for 2 years: ");

this.productInfo.setProductWarranty(input.next());

//PRODUCT PRICE

System.out.print("Enter the price for " + this.productInfo.getProductName() +" >> ");

this.productInfo.setProductPrice(input.nextInt());

//PRODUCT STOCK LEVEL

System.out.print("Enter the stock level for " + this.productInfo.getProductName() + " >> ");

this.productInfo.setProductStockLevel(input.nextInt());

//PRODUCT SUPPLIER

System.out.print("Enter the supplier for " + this.productInfo.getProductName() + " >> ");

this.productInfo.setProductSupplier(input.next());

//Info sucessfully captured message

System.out.println("Product details has been saved succesfully");

//Add Object to array list

this.MyList.add(productInfo);

}//end of method------------------------------------------------------------

//Method to search for a product using product code

public void searchProduct() {

//Ask User for Input product code

System.out.print("\nPlease enter the product code to search: ");

String userInput = input.next();//Get Product Code input

boolean found = false;

//Loop for the length of array

for (int j = 0 ; j < this.MyList.size(); j++)

{//Check if userInput matches with array product code

if (userInput.equals(this.MyList.get(j).getProductCode()))

{

found = true;//If product is found change to true

System.out.print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.print("\nPRODUCT SEARCH RESULTS");

System.out.print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

System.out.print("\nPRODUCT CODE: " + this.MyList.get(j).getProductCode());

System.out.print("\nPRODUCT NAME: " + this.MyList.get(j).getProductName());

System.out.print("\nPRODUCT WARRANTY: " + this.MyList.get(j).getProductWarranty());

System.out.print("\nPRODUCT CATEGORY " + this.MyList.get(j).getProductCategoryString());

System.out.print("\nPRODUCT PRICE R" + this.MyList.get(j).getProductPrice());

System.out.print("\nPRODUCT STOCK LEVELS: " + this.MyList.get(j).getProductStockLevel());

System.out.print("\nPRODUCT SUPPLIER: " + this.MyList.get(j).getProductSupplier());

System.out.print("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

}

}

//If product is not found inform user

if (found == false)

{

System.out.println("The product cannot be located. Invalid input.");

}

}//end of method------------------------------------------------------------

//Method to delete a product

public void deleteProduct() {

System.out.print("\nEnter product code: ");

String userInput = input.next();

int i = 0;

boolean control = false;

//Find product code

for (i = 0; i < this.MyList.size(); i++)

{//If Product is found

if (this.MyList.get(i).getProductCode().equals(userInput))

{//Ask user confirmation to delete

control = true;

System.out.print("Would you like to delete the product? Type 1 for YES or anthing else for NO: ");

userInput = input.next();

//If user confirms to delete

if ("1".equals(userInput))

{//deletes object

//Inform user that product was deleted

System.out.println("Product was successfully deleted");

this.MyList.remove(i);//Delete object

break;//exit

}

else//If user does not confirm delete

{

break;//exit

}

}

}

//Inform user in case product code is not found

if (control == false)

{

System.out.println("Product was not found");

}

}//end of method------------------------------------------------------------

//Method to update details of product

public void updateProduct () {

System.out.print("\nPlease enter the product code to search: ");

String userInput = input.next();

boolean control = false;

for (int i = 0; i < this.MyList.size(); i++)

{//If Product is found

if (this.MyList.get(i).getProductCode().equals(userInput))

{

control = true;

//WARRANTY======================================================

System.out.print("Update the warranty? (y) Yes, (n) No: ");

userInput = input.next();

//If answer is yes update

if ("y".equals(userInput))

{

System.out.print("Enter the new warranty for " + this.MyList.get(i).getProductName() + " >> ");

userInput = input.next();

this.MyList.get(i).setProductWarranty(userInput);

}

//PRICE=========================================================

System.out.print("Update the product price? (y) Yes, (n) No: ");

userInput = input.next();

//If answer is yes update

if ("y".equals(userInput))

{

System.out.print("Enter the new product price for " + this.MyList.get(i).getProductName() + " >> ");

userInput = input.next();

this.MyList.get(i).setProductPrice(parseInt(userInput));

}

//STOCK LEVEL===================================================

System.out.print("Update the stock level? (y) Yes, (n) No: ");

userInput = input.next();

//If answer is yes update

if ("y".equals(userInput))

{

System.out.print("Enter the new product stock level for " + this.MyList.get(i).getProductName() + " >> ");

userInput = input.next();

this.MyList.get(i).setProductStockLevel(parseInt(userInput));

}

//Inform user that product was updated sucessfully

System.out.println("Product details has been updated sucessfully!!!");

}

}

//If user does not type valid product code

if (control == false)

{

System.out.println("Product was not found.");

}

}//end of method------------------------------------------------------------

//Method to Display report

public void printReport() {

int totalProductCount = 0;

int totalProductValue = 0;

int averageProductValue = 0;

System.out.println("PRODUCT REPORT");

System.out.println("====================================================");

//Print each object gets and sets

for (int i = 0; i < this.MyList.size(); i++)

{

System.out.println("PRODUCT " + (i + 1));

System.out.println("----------------------------------------------------");

System.out.println("PRODUCT CODE >> " + this.MyList.get(i).getProductCode());

System.out.println("PRODUCT NAME >> " + this.MyList.get(i).getProductName());

System.out.println("PRODUCT CATEGORY >> " + this.MyList.get(i).getProductCategory());

System.out.println("PRODUCT WARRANTY >> " + this.MyList.get(i).getProductWarranty());

System.out.println("PRODUCT PRICE >> " + this.MyList.get(i).getProductPrice());

System.out.println("PRODUCT LEVEL >> " + this.MyList.get(i).getProductStockLevel());

System.out.println("PRODUCT SUPPLIER >> " + this.MyList.get(i).getProductSupplier());

System.out.println("----------------------------------------------------");

totalProductCount = totalProductCount + 1;

totalProductValue = totalProductValue + this.MyList.get(i).getProductPrice();

}

averageProductValue = totalProductValue / totalProductCount;

System.out.println("====================================================");

System.out.println("TOTAL PRODUCT COUNT: " + totalProductCount);

System.out.println("TOTAL PRODUCT VALUE: R " + totalProductValue);

System.out.println("AVERAGE PRODUCT VALUE R " + averageProductValue);

System.out.println("====================================================");

}//end of method------------------------------------------------------------

//Method to terminate program

public void exitApplication() {

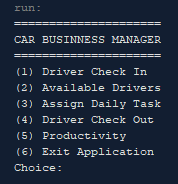
System.exit(0);//terminate

}//end of method------------------------------------------------------------

}//==========================================================================END

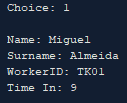
# Question 2

Menu

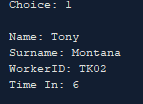


Option 1

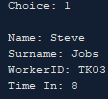
Save driver 1



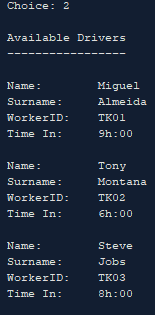
Save driver 2



Save driver 3

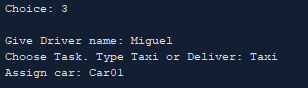


Option 2

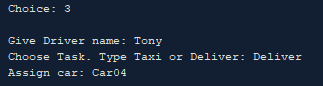


Option 3

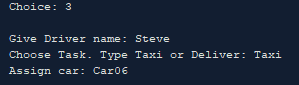
Driver 1



Driver 2

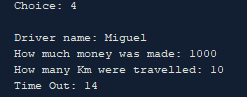


Driver 3

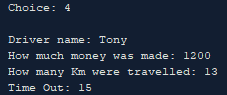


Option 4

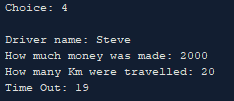
Driver 1



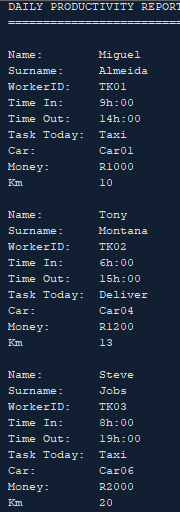
Driver 2



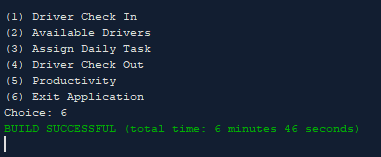
Driver 3



Option 5



Option 6



Code

/\*

Program description: This program will help an entrepeneur

manage his taxi and delivery business daily and keep track of drivers

The company consists of 10 cars and each car has a specific fuel consumption

Each driver has a daily task either taxi or delivery

Productivity is important so perfomance is taken into account

\*/

package assignment\_question\_2;

import java.util.Scanner;

public class Assignment\_Question\_2 {

public static void main(String[] args) {

//arrays, loops, inheritance, constructors and information hiding

//Learning Unit 1: Advanced arrays.

//Learning Unit 2: Introduction to inheritance.

Scanner input = new Scanner(System.in);

Manager methods = new Manager();

//Header

System.out.print("=====================\nCAR BUSINNESS MANAGER"

+ "\n=====================");

boolean valid = false;

do

{

//Display Menu

methods.menu();

//Get user choice

String userInput = input.next();

switch(userInput)

{

case "1":

methods.checkInDriver();

break;

case "2":

methods.availableDrivers();

break;

case "3":

methods.assignTask();

break;

case "4":

methods.checkOutDriver();

break;

case "5":

methods.printReport();

break;

case "6":

valid = true;//Exit

break;

}

} while (!valid);//Keep loop till valid is true

}

}//==========================================================================END

package assignment\_question\_2;

public class Driver {

private String name;

private String Surname;

private int timeIn;

private int timeOut;

private String workerID;

private String dailyTask;

private String Car;

private double fuel;

private int money;

private int km;

public Driver() {

}

/\*\*

\* @return the timeOut

\*/

public int getTimeOut() {

return timeOut;

}

/\*\*

\* @param timeOut the timeOut to set

\*/

public void setTimeOut(int timeOut) {

this.timeOut = timeOut;

}

/\*\*

\* @return the km

\*/

public int getKm() {

return km;

}

/\*\*

\* @param km the km to set

\*/

public void setKm(int km) {

this.km = km;

}

/\*\*

\* @return the dailyTask

\*/

public String getDailyTask() {

return dailyTask;

}

/\*\*

\* @param dailyTask the dailyTask to set

\*/

public void setDailyTask(String dailyTask) {

this.dailyTask = dailyTask;

}

/\*\*

\* @return the Car

\*/

public String getCar() {

return Car;

}

/\*\*

\* @param Car the Car to set

\*/

public void setCar(String Car) {

this.Car = Car;

}

/\*\*

\* @return the money

\*/

public int getMoney() {

return money;

}

/\*\*

\* @param money the money to set

\*/

public void setMoney(int money) {

this.money = money;

}

/\*\*

\* @return the name

\*/

public String getName() {

return name;

}

/\*\*

\* @param name the name to set

\*/

public void setName(String name) {

this.name = name;

}

/\*\*

\* @return the Surname

\*/

public String getSurname() {

return Surname;

}

/\*\*

\* @param Surname the Surname to set

\*/

public void setSurname(String Surname) {

this.Surname = Surname;

}

/\*\*

\* @return the timeIn

\*/

public int getTimeIn() {

return timeIn;

}

/\*\*

\* @param timeIn the timeIn to set

\*/

public void setTimeIn(int timeIn) {

this.timeIn = timeIn;

}

/\*\*

\* @return the workerID

\*/

public String getWorkerID() {

return workerID;

}

/\*\*

\* @param workerID the workerID to set

\*/

public void setWorkerID(String workerID) {

this.workerID = workerID;

}

@Override

public String toString()

{

return "\nName: " + getName()

+ "\nSurname: " + getSurname()

+ "\nWorkerID: " + getWorkerID()

+ "\nTime In: " + getTimeIn() + "h:00";

}

}//==========================================================================END

package assignment\_question\_2;

import java.util.ArrayList;

import java.util.Scanner;

public class Manager extends Driver {

Scanner input = new Scanner(System.in);

public Driver worker = new Driver();

public ArrayList<Driver> MyList = new ArrayList<Driver>();

public Manager() {

}

//Method to display menu

public void menu() {

System.out.println("\n(1) Driver Check In");

System.out.println("(2) Available Drivers");

System.out.println("(3) Assign Daily Task");

System.out.println("(4) Driver Check Out");

System.out.println("(5) Productivity");

System.out.println("(6) Exit Application");

System.out.print("Choice: ");

}//end of method------------------------------------------------------------

//Method to Check In driver when they arrive to work

public void checkInDriver() {

//Ask User for Inputs

this.worker = new Driver();//Allocating memory

System.out.print("\nName: ");

this.worker.setName(input.next());

System.out.print("Surname: ");

this.worker.setSurname(input.next());

System.out.print("WorkerID: ");

this.worker.setWorkerID(input.next());

System.out.print("Time In: ");

this.worker.setTimeIn(input.nextInt());

this.MyList.add(worker);//Save Object in array list

}//end of method------------------------------------------------------------

//Method to display available drivers

public void availableDrivers() {

System.out.println("\nAvailable Drivers");

System.out.println("-----------------");

//Loop thru ist and print all drivers that checked In

for (int i = 0; i < this.MyList.size(); i++)

{

System.out.print(this.MyList.get(i).toString() + "\n");

}

}//end of method------------------------------------------------------------

//Method to Distribute Task to Drivers

public void assignTask() {

System.out.print("\nGive Driver name: ");

String userInput = input.next();

for (int i = 0; i < this.MyList.size(); i ++)

{//Search for Driver with driver name

if (this.MyList.get(i).getName().equals(userInput))

{//Ask user for inputs

System.out.print("Choose Task. Type Taxi or Deliver: ");

this.MyList.get(i).setDailyTask(input.next());

System.out.print("Assign car: ");

this.MyList.get(i).setCar(input.next());

break;//Exit Loop

}

}

}//end of method------------------------------------------------------------

//Method to check out Driver

public void checkOutDriver() {

System.out.print("\nDriver name: ");

String userInput = input.next();

for (int i = 0; i < this.MyList.size(); i ++)

{//Search for Driver with driver name

if (this.MyList.get(i).getName().equals(userInput))

{//Ask user for inputs

System.out.print("How much money was made: ");

this.MyList.get(i).setMoney(input.nextInt());

System.out.print("How many Km were travelled: ");

this.MyList.get(i).setKm(input.nextInt());

System.out.print("Time Out: ");

this.MyList.get(i).setTimeOut(input.nextInt());

break;//exit loop

}

}

}//end of method------------------------------------------------------------

//Method Print Productivity report

public void printReport() {

//Header

System.out.println("\nDAILY PRODUCTIVITY REPORT");

System.out.println("=========================");

toString();

}//end of method------------------------------------------------------------

@Override

public String toString()

{//Print each driver daily performance

for(int i = 0; i < this.MyList.size(); i++)

{

System.out.println( "\nName: " + this.MyList.get(i).getName()

+ "\nSurname: " + this.MyList.get(i).getSurname()

+ "\nWorkerID: " + this.MyList.get(i).getWorkerID()

+ "\nTime In: " + this.MyList.get(i).getTimeIn() + "h:00"

+ "\nTime Out: " + this.MyList.get(i).getTimeOut() + "h:00"

+ "\nTask Today: " + this.MyList.get(i).getDailyTask()

+ "\nCar: " + this.MyList.get(i).getCar()

+ "\nMoney: R" + this.MyList.get(i).getMoney()

+ "\nKm " + this.MyList.get(i).getKm());

}

return "";

}

}//==========================================================================END