GARBAGE

& RECYCLİNG

PROJECT

Team Members

Dorukhan YILDIZ   202011003

Hikmet Berkin BULUT   202111057

Ahmet Eren YAĞLI   202011038

A-Introduction of Program

Program Language: C++

Project Name: Recycling Vending Machines

Compiler: GCC

B-Design of Program

1)The Purpose of the Program

Main purpose is encouraging the people to recycling by exchange the waste with money by the vending machines.

Goal: Our aim is creating an opportunity for the users to exchange their wastes with money that calculated by default price which is set by admin and wastes amount/weight and attributes which are given by user. There is also an opportunity for Admin to create vending machines with its location (city), storage limit, deposit money and accepted waste type. At this point admin can also set default prices of accepted wastes and after all can see the user's information.

We used the C++ language to write this program.

After admin login, if there is a published machine, users can access this machine by registering or logging the user menu. User menu works as:

- Asks to user to register or login the published machine properly. Requested data from the user are username and password.

- Users must re-login after registering themselves in the system. After this entry, they will be able to access 3 basic functions in the program

- If admin wants to reach the admin menu, he/she can choose the login and enters 000 to username and 000 to password so that the machine system will transfers to admin menu.
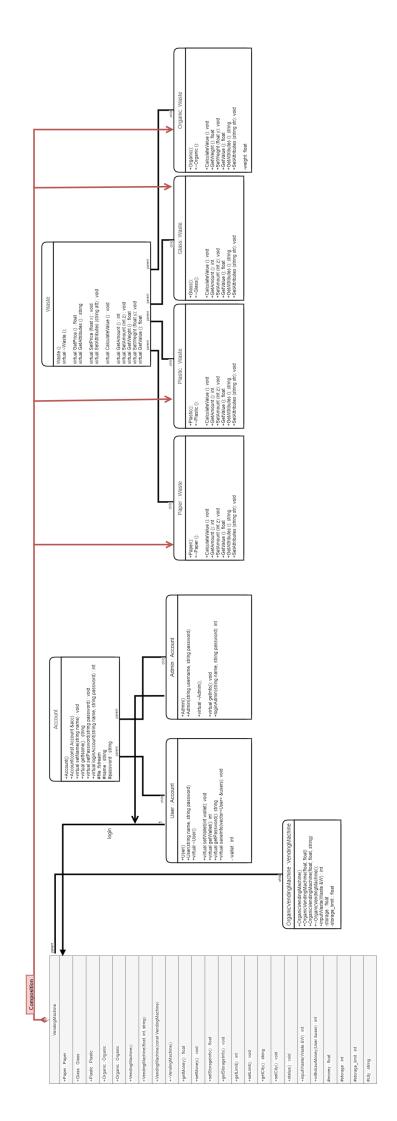
2)Solution Design

- When the program starts, it asks for admin login to create vending machines. Accepted waste type determines the machines type. Ex: If

accepted waste type is Organic so the vending machine created as organic vending machine and default waste created as Organic. Next, storage limit and deposit money of vending machine must be set then default price of waste have to be set.

- If there is any vending machine created, admin can choose it to set to the system. After that, admin can see and change the deposit money of current vending machine that set by admin also can publish it to the user's access.

- After publishing a vending machine, system asks for register or login to users. After a user register and login to published machine, he/she can input waste which has a dependent type to type of the vending machine. User must enter amount/weight and attributes of the wastes then program check if machine has enough storage. If it so User gets the money that depends on value of the waste to their accounts, then user can withdraw the money from current machine if there are enough money.

- If admin wants to reach admin menu again, he/she can use 000 for username and 000 for password. If any user Registered to system admin can see its information. Also create new vending machines, set a new current machine, change its deposit money, and publish the current machine.

Note: Every vending machine has its own default wastes and whenever a user wants to input waste, system creates a unique waste by copying the current machines default waste.

Composition

VendingMachine

```
+Paper : Paper
+Glass: Glass
+Plastic : Plastic
+Organic: Organic
+Organic: Organic
+VendingMachine()
+VendingMachine(float, int, string)
+VendingMachine(const VendingMachine)
+~VendingMachine()
+getMoney() : float
+setMoney() : void
+setStorage(int) : float
+getWeight() : int
+getStorage() : void
+setLimit() : int
+setLimit() : void
+getCity() : string
+setCity() : void
+status() : void
+inputWaste(Waste &W) : int
+withdrawMoney(User &user) : int
#money : float
#storage : int
#storage_limit : int
#city : string
```

Account

```
+Account()
+Account(const Account &acc)
+virtual setName(string name) : void
+virtual getName() : string
+virtual setPassword(string password) : void
+virtual loginAccount(string name, string password) : int
#file : fstream
#name : string
#password : string
```

User : Account

```
+User()
+User(string name, string password)
+virtual ~User()
+virtual setWallet(int wallet) : void
+virtual getWallet() : int
+virtual getPassword() : string
+virtual saveInfo(vector<User> &users) : void
- wallet : int
```

login

Admin : Account

```
+Admin()
+Admin(string username, string password)
+virtual ~Admin();
+virtual getInfo() : void
+virtual getInfo() : int
+loginAdmin(string name, string password) : int
```

OrganicVendingMachine : VendingMachine

```
+OrganicVendingMachine()
+OrganicVendingMachine(float, float)
+OrganicVendingMachine(float, float, string)
+~OrganicVendingMachine();
+inputWaste(Waste &W) : int
-storage : float
-storage_limit : float
```

Waste

```
Waste ()
virtual ~Waste ();
virtual GetPrice () : float
virtual GetAttributes () : string
virtual SetPrice (float x) : void
virtual SetAttributes (string att) : void
virtual CalculateValue () : void
virtual GetAmount () : int
virtual SetAmount (int z) : void
virtual GetWeight () : float
virtual SetWeight (float y) : void
virtual GetValue () : float
```

Paper : Waste

```
+Paper();
+~Paper ();
+CalculateValue () : void
+GetAmount () : int
+SetAmount (int z) : void
+GetValue () : float
+GetAttributes () : string
+SetAttributes (string str) : void
```

Plastic : Waste

```
+Plastic();
+~Plastic ();
+CalculateValue () : void
+GetAmount () : int
+SetAmount (int z) : void
+GetValue () : float
+GetAttributes () : string
+SetAttributes (string str) : void
```

Glass: Waste

```
+Glass();
+~Glass();
+CalculateValue () : void
+GetAmount () : int
+SetAmount (int z) : void
+GetValue () : float
+GetAttributes () : string
+SetAttributes (string str) : void
```

Organic: Waste

```
+Organic();
+~Organic();
+CalculateValue () : void
+GetWeight () : float
+SetWeight (float x) : void
+SetValue () : float
+GetAttributes () : string
+SetAttributes (string str) : void
-weight : float
```

4)Testing Results

```
Register or Login: Login
Enter your name: 000
Enter your password: 000
[1] Get Info
[2] Create a Vending Machine
[3] Check Current Balance
[4] Set This Vending Machine
[5] Publish Machine
[6] Turn Off
Enter Choice : █
```

1-) If the username and password are equal to 000, we will have an admin login.

```
Waiting for admin log in.
Username: admin
Password: root█
```

2-) When setting up the vending machine for the first time, we need to enter the username and password as in the picture.

```
Register or Login: Register
Enter a username: Kullanici
Enter a password: 1234
Enter a password again: 1234
You have successfully registered.
Register or Login: Login
Enter your name: Kullanici
Enter your password: 1234
Login successful.

===== Welcome Kullanici =====

Current Balance : 0
[1] Input Waste
[2] Withdraw Money
[3] Log Out
Enter Choice: █
```

3-) Anyone who uses the vending machine for the first time should register first and then enter the username and password from the login screen and can log in to the vending machine.

```
[1] Get Info
[2] Create a Vending Machine
[3] Check Current Balance
[4] Set This Vending Machine
[5] Publish Machine
[6] Turn Off
Enter Choice : 1
Username   Password     Budget
eren          123       300

doruk         123       4000
```

4-) If we choose 1 from the admin menu, it prints the usernames, passwords and balances of the users logged into the system.

```
[1] Get Info
[2] Create a Vending Machine
[3] Check Current Balance
[4] Set This Vending Machine
[5] Publish Machine
[6] Turn Off
Enter Choice : 2
Welcome to Vengding Machine Setter!
Accepted waste type:
1-Organic | 2-Anorganic ==> 1
City Name: Ankara
Deposit Money: 10000
Input the storage limit: 400
Set the price of Organic waste: 1
```

5-) If we choose 2 from the admin menu, we can create a new vending machine.

```
[1] Get Info
[2] Create a Vending Machine
[3] Check Current Balance
[4] Set This Vending Machine
[5] Publish Machine
[6] Turn Off
Enter Choice : 3
[1] View Current Balance
[2] Deposit Money
Enter Choice : 1
Vending Machine Balance: 10000
```

6-) If we choose 3 from the admin menu, we can view the money in the vending machine or deposit money in the vending machine.

```
[1] Get Info
[2] Create a Vending Machine
[3] Check Current Balance
[4] Set This Vending Machine
[5] Publish Machine
[6] Turn Off
Enter Choice : 4
Anorganic Machines:
Organic Machines:
[1] Ankara

Select number of city: █
```

7-) By selecting 4 from the admin menu, we can set which vending machine will run as.

```
[1] Get Info
[2] Create a Vending Machine
[3] Check Current Balance
[4] Set This Vending Machine
[5] Publish Machine
[6] Turn Off
Enter Choice : 5
```

8-) We can publish the vending machine we have created by selecting 5 from the admin menu.

```
Current Balance : 0
[1] Input Waste
[2] Withdraw Money
[3] Log Out
Enter Choice: 1
1-Paper | 2-Plastic | 3-Glass ==> 2
Enter amount of plastic: 100
Completed Successfully

You succesfully added waste to machine. Your balance updated!
```

9-) The user can throw a garbage to the vending machine by selecting 1 from the user menu.

```
Current Balance : 300
[1] Input Waste
[2] Withdraw Money
[3] Log Out
Enter Choice: 2
Don't forget your money!!!
```

10-) The user can withdraw all the money accumulated in his / her virtual wallet in the vending machine by selecting 2 from the user menu.

```
===== Welcome afu =====
Current Balance : 0
[1] Input Waste
[2] Withdraw Money
[3] Log Out
Enter Choice : 3
===== Goodbye afu =====
```

11-) The user can logout by selecting 3 from the user menu.

5)Errors and Shortcomings

- Admin can always create new vending machines but cannot perform the creating new waste types. Also, users cannot reach all created vending machines except published one. That causes some problems such as if user has inorganic wastes and published machine is organic vending machine, it is useful for the user.

5.1How much it serves the purpose

- Program works well within the scope of project because it performs most of the process, which is vital to purpose like, admin can determine the vending machine attributes, see the information of the users and user can exchange the waste with money.

5.2) Result / Comment / Suggestion
  Finally, we achieved what we wanted at the beginning. We used Polymorphism by creating wastes with different forms and Abstraction by declaring virtual functions of parent classes and making the parent Waste class pure abstract in our project.