Final Project presentation

Here is where my presentation begins!
https://www.youtube.com/watch?v=AYPLLLF26_A&t=9s

20XX

Theme of my project is financial digitization. And the project can perform many financials simulation.

The project consists of 6 different classes: Account, DailyAccount, SavingsAccount, Warehouse, User, and Main class

Content:

```java
abstract class Account {
    protected double balance;

    // constructor
    public Account(double balance) {
        this.balance = balance;
    }

    //  method
    public void deposit(double amount) {
        balance += amount;
    }

    // abstract method
    public abstract void withdraw(double amount);
}
```

01

Abstraction

Account class is an abstract class and withdraw is abstract method

02 Inheritance

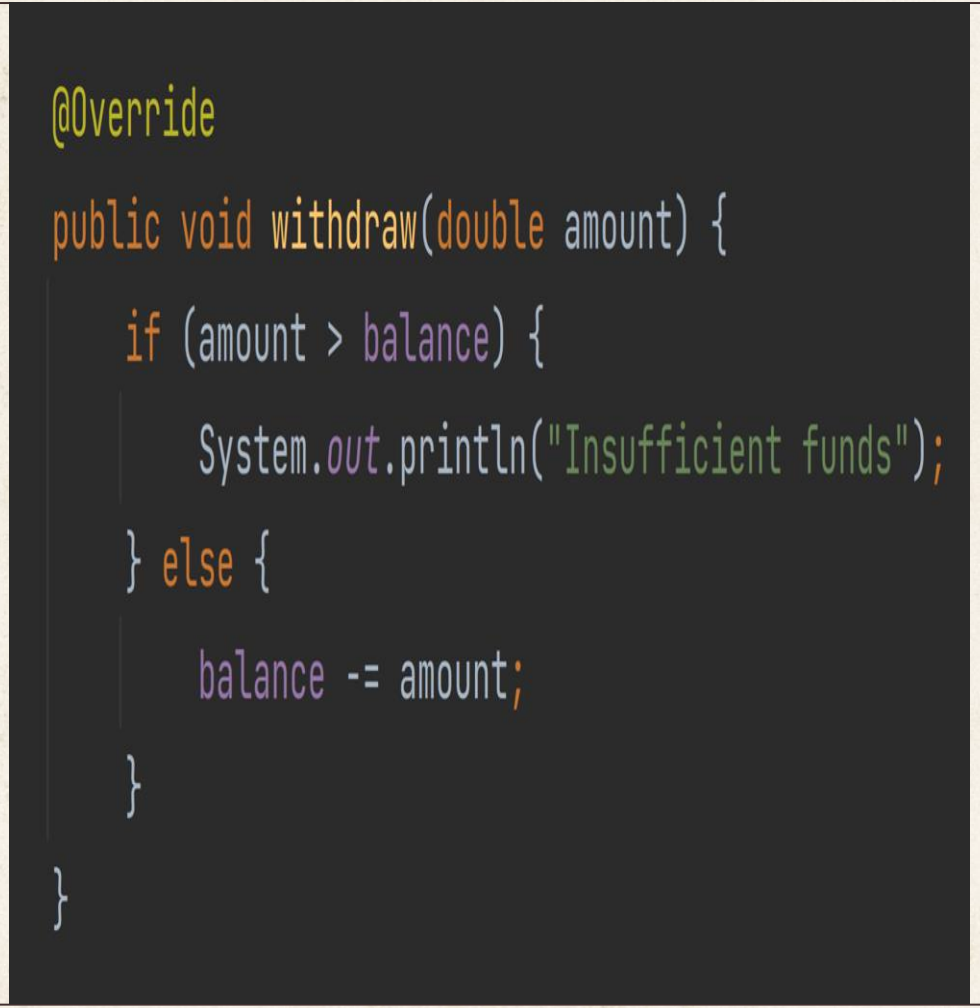
```
class DailyAccount extends Account {
```

```
class SavingsAccount extends Account {
```

Inheritance was implemented by both Savings and Daily Account class by using the keyword extends.

03 Polymorphism

Polymorphism with abstract class implementation. Because we always need to override the abstract method of the parent class on the each child class



```java
@Override
public void withdraw(double amount) {
    if (amount > balance) {
        System.out.println("Insufficient funds");
    } else {
        balance -= amount;
    }
}
```

Also with getAccount method. Because they are in the same class and have different functionality and parameter then it’s method overloading

04 Generics

Generic class

```
class Warehouse < A extends Account > {
```

Implemented by warehouse class and warehouse takes a type of parameter "A" extends Account.

That means it will get the subclass "Account"

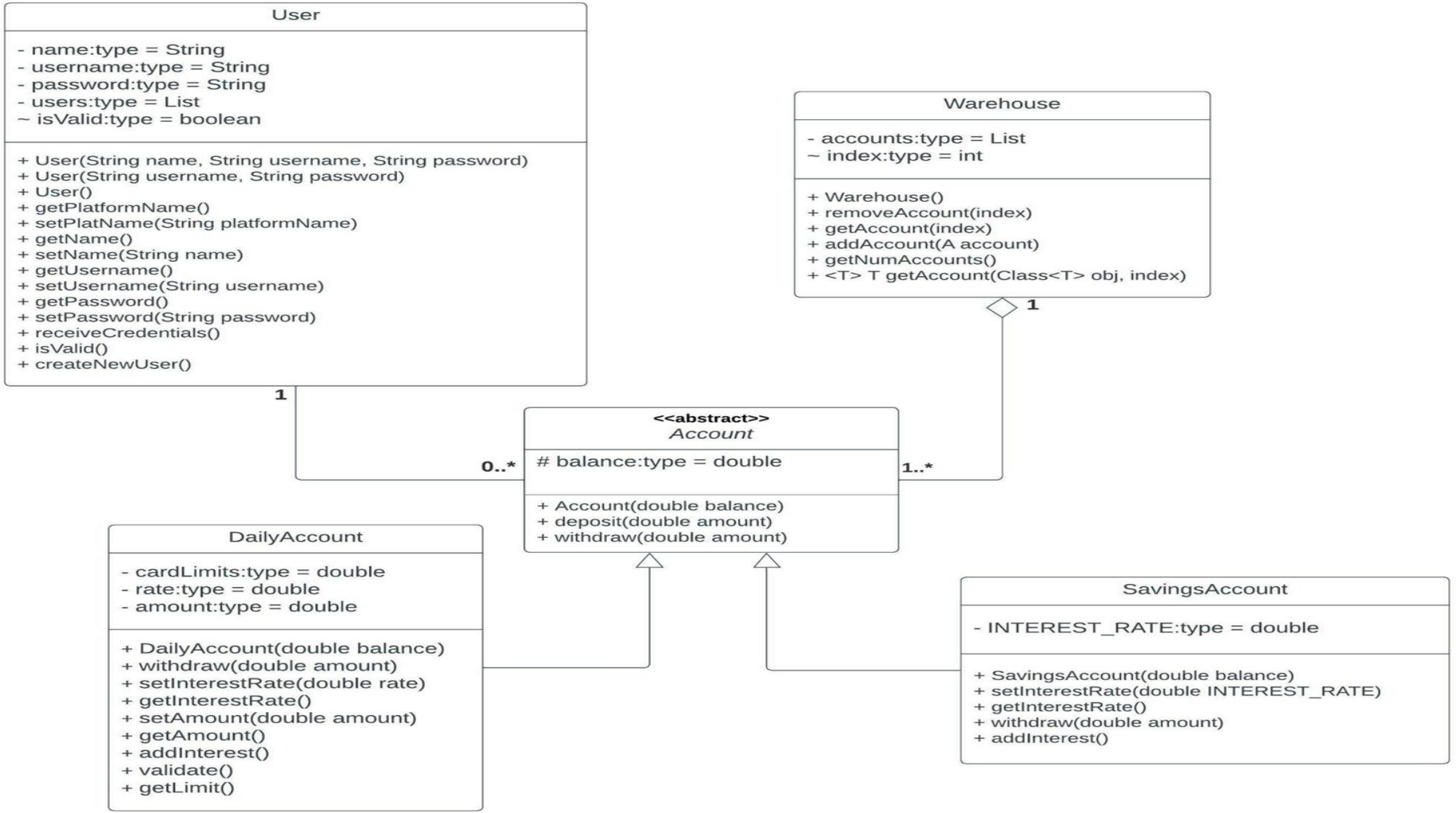
```java
public <T> T getAccount(Class<T> obj, int index) {
    Account account = accounts.get(index);
    if (obj.isInstance(account)) {
        return obj.cast(account);
    } else {
        return null;
    }
}
```

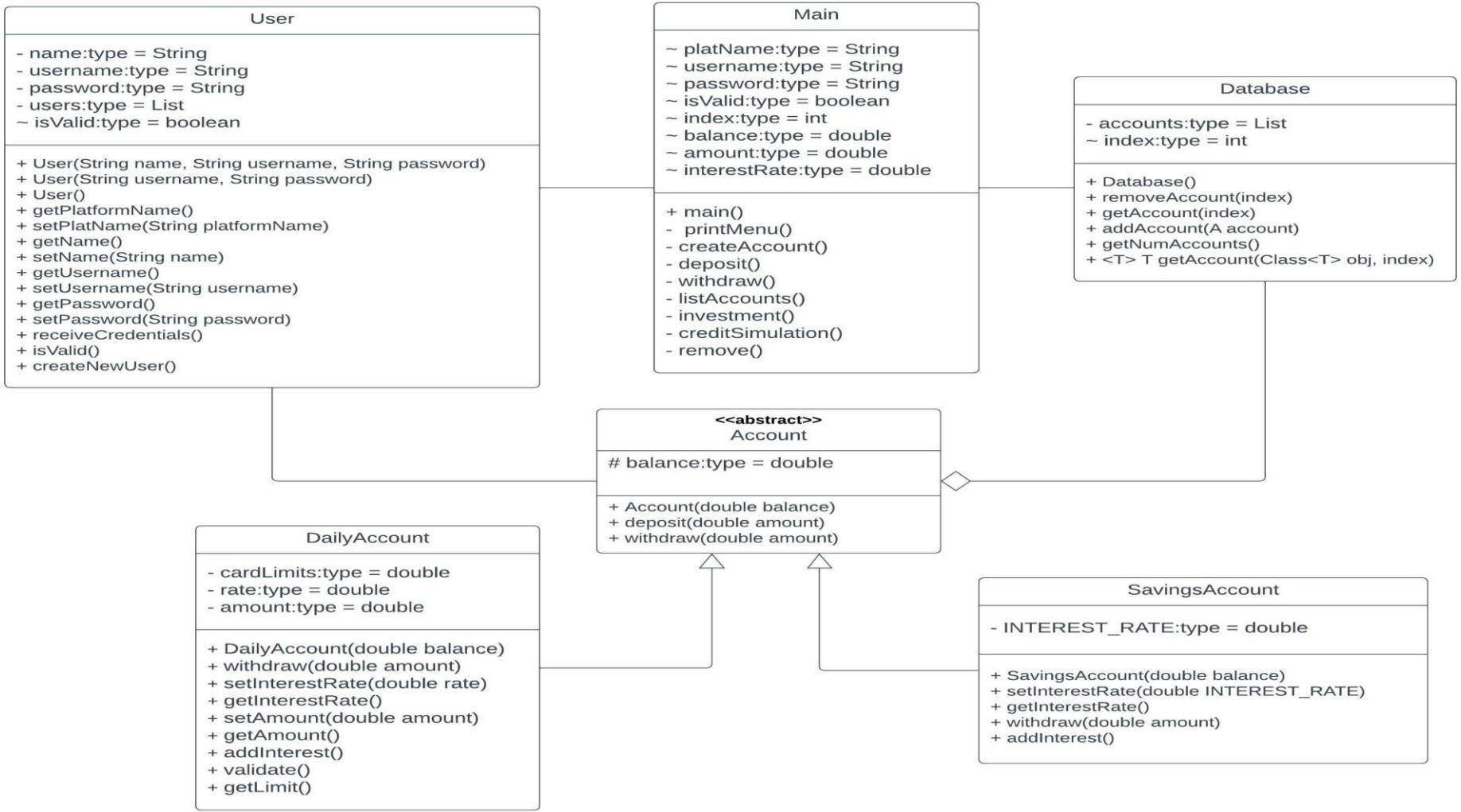
Generic method

takes a class<T> object "obj" and an index as parameters. That mean the method can return an object of any type.

05

Class diagram


User
- name:type = String
- username:type = String
- password:type = String
- users:type = List
~ isValid:type = boolean
+ User(String name, String username, String password)
+ User(String username, String password)
+ User()
+ getPlatformName()
+ setPlatName(String platformName)
+ getName()
+ setName(String name)
+ getUsername()
+ setUsername(String username)
+ getPassword()
+ setPassword(String password)
+ receiveCredentials()
+ isValid()
+ createNewUser()
Warehouse
- accounts:type = List
~ index:type = int
+ Warehouse()
+ removeAccount(index)
+ getAccount(index)
+ addAccount(A account)
+ getNumAccounts()
+ <T> T getAccount(Class<T> obj, index)
1
1
<<abstract>>
Account
balance:type = double
+ Account(double balance)
+ deposit(double amount)
+ withdraw(double amount)
0..*
1..*
DailyAccount
- cardLimits:type = double
- rate:type = double
- amount:type = double
+ DailyAccount(double balance)
+ withdraw(double amount)
+ setInterestRate(double rate)
+ getInterestRate()
+ setAmount(double amount)
+ getAmount()
+ addInterest()
+ validate()
+ getLimit()
SavingsAccount
- INTEREST_RATE:type = double
+ SavingsAccount(double balance)
+ setInterestRate(double INTEREST_RATE)
+ getInterestRate()
+ withdraw(double amount)
+ addInterest()


User
- name:type = String
- username:type = String
- password:type = String
- users:type = List
~ isValid:type = boolean
+ User(String name, String username, String password)
+ User(String username, String password)
+ User()
+ getPlatformName()
+ setPlatName(String platformName)
+ getName()
+ setName(String name)
+ getUsername()
+ setUsername(String username)
+ getPassword()
+ setPassword(String password)
+ receiveCredentials()
+ isValid()
+ createNewUser()
Main
~ platName:type = String
~ username:type = String
~ password:type = String
~ isValid:type = boolean
~ index:type = int
~ balance:type = double
~ amount:type = double
~ interestRate:type = double
+ main()
- printMenu()
- createAccount()
- deposit()
- withdraw()
- listAccounts()
- investment()
- creditSimulation()
- remove()
Database
- accounts:type = List
~ index:type = int
+ Database()
+ removeAccount(index)
+ getAccount(index)
+ addAccount(A account)
+ getNumAccounts()
+ <T> T getAccount(Class<T> obj, index)
<<abstract>>
Account
balance:type = double
+ Account(double balance)
+ deposit(double amount)
+ withdraw(double amount)
DailyAccount
- cardLimits:type = double
- rate:type = double
- amount:type = double
+ DailyAccount(double balance)
+ withdraw(double amount)
+ setInterestRate(double rate)
+ getInterestRate()
+ setAmount(double amount)
+ getAmount()
+ addInterest()
+ validate()
+ getLimit()
SavingsAccount
- INTEREST_RATE:type = double
+ SavingsAccount(double balance)
+ setInterestRate(double INTEREST_RATE)
+ getInterestRate()
+ withdraw(double amount)
+ addInterest()

//Consulting/Toolkit/

Program Demo

20xx

Starting menu

```
Create your partners name: TEST

Create Account
--------------------------
Enter name: TEST
Enter new username: TEST
Enter new password: TEST
Retype new password: TEST
User Successfully Created.

TEST partners Login
----------
Enter username ('0' to CLOSE PROGRAM): TEST
Enter password: TEST
```

Main menu

```
1. Create account
2. Add Money          --- (Income)
3. Take Money         --- (Outcome)
4. Passive Income     --- (Investment)
5. Passive Outcome    --- (Credit/Paylater)
6. List accounts
7. Remove account
8. Quit

Enter your choice:
```

Create account

```
Enter your choice: 1
Creating Account:
'1' Daily account
'2' Savings account
Input: 1
Enter initial balance: 100
```

```
Enter your choice: 1
Creating Account:
'1' Daily account
'2' Savings account
Input: 2
Enter initial balance: 200
```

After create account

```
Account number 1:
  Type       :  Daily account
  Balance    :  100.0

Account number 2:
  Type       :  Savings account
  Balance    :  200.0
```

Add money

```
Enter your choice: 2
Enter account number: 1
Enter amount to deposit: 10
```

After

```
Account number 1:
  Type       :  Daily account
  Balance    :  110.0

Account number 2:
  Type       :  Savings account
  Balance    :  200.0
```

Take money

```
Enter your choice: 3
Enter account number: 1
Enter amount to withdraw: 20
```

After

```
Account number 1:
  Type       :  Daily account
  Balance    :  90.0

Account number 2:
  Type       :  Savings account
  Balance    :  200.0
```

Passive income

```
Enter your choice: 4
Enter account number: 2
Enter interest rate: 1
With Interest Rate 1.0
New Balance    :  400.00
Profit         :  200.00
Monthly profit :  16.66667
Try another invest? (Y/N)
n
```

Passive outcome

```
Enter your choice: 5
Enter account number: 1
Card Limit   = 100000.0
User balance = 90.0
Enter amount to pay: 30
Enter the interest rate: 0.1
With Interest Rate 0.1
New Balance    :  57.00
loss           :  3.00
Monthly loss   :  0.250
```

Remove account

```
Enter your choice: 7
Account number 1:
  Type       :  Daily account
  Balance    :  57.0

Account number 2:
  Type       :  Savings account
  Balance    :  400.0

Enter account number to be removed: 2
```

After remove

```
Enter your choice: 6
Account number 1:
  Type      :  Daily account
  Balance   :  57.0
```

20XX

//Consulting/*Toolkit*/

Thanks!

Do you have any questions?

youremail@freepik.com | +91 620 421 838 yourwebsite.com