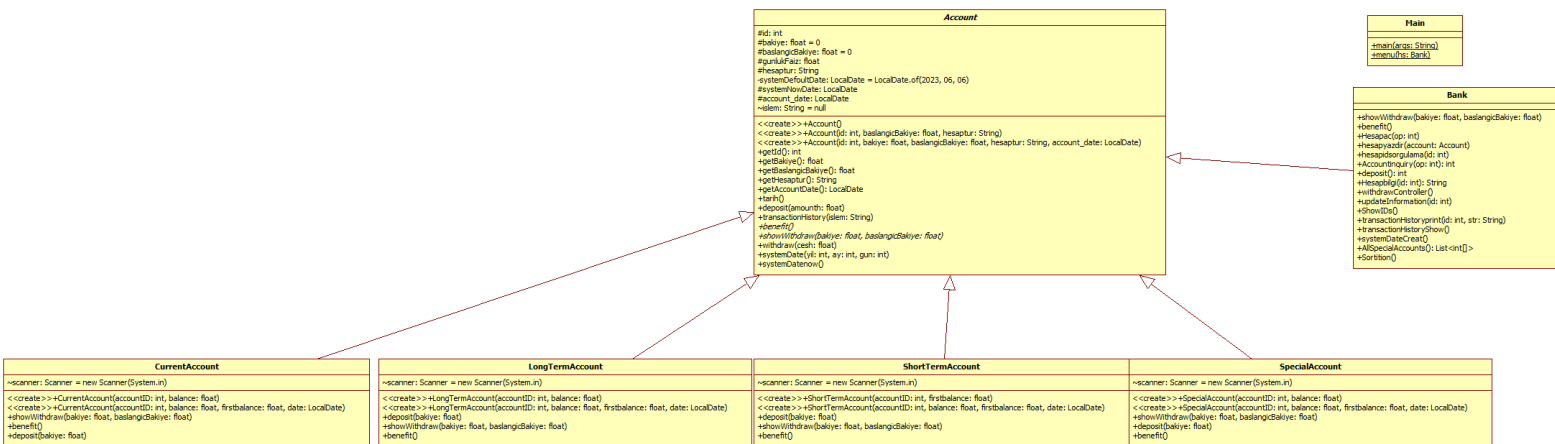




NESNEYE DAYALI PROGRAMLAMA
FİNAL
PROJE ÖDEVİ

KUBİLAY DİCLE
202013172019

PROJENİN UML DİYAGRAMI



Bu linkten görselin asıl boyutuna ulaşabilirsiniz ayrıca size orijinal resmini atacağım hocam

→ <https://l24.im/pgDw>

PROJENİN İŞLEYİŞ MANTIĞI;

Bu kod, bir banka hesabının temel işlevlerini gerçekleştiren bir sınıf olan "Hesap" sınıfını tanımlar. Bu sınıf, bir hesap kimliği (id) ve hesap bakiyesi (bakiye) gibi özelliklere sahiptir. Ayrıca, hesabın başlangıç bakiyesini (baslangicBakiye) izlemek için bir özellik daha bulunmaktadır. Başlangıç bakiyesi, hesap açıldığında belirlenen ilk bakiyedir.

Hesap sınıfı, para yatırma ve para çekme işlemlerini gerçekleştirmek için iki adet metoda sahiptir: paraYatir ve paraCek.

- paraYatir metodu, hesaba belirli bir miktarda para yatırmak için kullanılır. Bu metod, mevcut bakiyeye yatırılan miktarı ekler.
- paraCek metodu, hesaptan belirli bir miktarı çekmek için kullanılır. Bu metod, mevcut bakiyeden çekilmek istenen miktarı düşer. Ancak, çekilmek istenen miktarın mevcut bakiyeden büyük olmaması gerektiğini kontrol eder. Aksi takdirde, işlem gerçekleştirilmez.

Bu Hesap sınıfı soyut bir sınıftır, yani doğrudan bir örneği oluşturulamaz. Bu sınıfı genişleten alt sınıflar, Hesap sınıfının özelliklerini ve metodlarını kullanabilir ve özelleştirebilir. Örneğin, farklı türde hesaplar için (örneğin, bireysel hesap veya işletme hesabı) ayrı alt sınıflar oluşturulabilir.

PROJENİN NESNE YAPILARI, METHODLARI VE DEĞİŞKENLERİ;

Sınıflar:

Proje toplam 7 adet sınıftan oluşmaktadır bunlar;

- Account
- ShortTermAccount
- LongTermAccount
- SpecialAccount
- CurrentAccount
- Bank
- Main

Account;

Account sınıfı, bir banka hesabını temsil etmek için kullanılan soyut bir sınıftır. Banka hesabının genel özelliklerini ve işlemlerini içerir.

Method Tanımları

- `getId()`: Hesabın benzersiz kimlik numarasını döndürür.
- `getBakiye()`: Hesabın mevcut bakiyesini döndürür.
- `getBaslangicBakiye()`: Hesabın başlangıç bakiyesini döndürür.
- `getHesaptur()`: Hesabın türünü döndürür.
- `getAccountDate()`: Hesabın oluşturulma tarihini döndürür.
- `tarih()`: Sistem başlangıç tarihini ekrana yazdırır.
- `deposit(float amounth)`: Belirtilen miktarı hesaba yatırır.
- `transactionHistory(String islem)`: Son işlemi kaydeder.
- `benefit()`: Hesaba faiz ekler (alt sınıflar tarafından uygulanmalıdır).
- `showWithdraw(float bakiye, float baslangicBakiye)`: Para çekme işlemi gerçekleştirir (alt sınıflar tarafından uygulanmalıdır).
- `withdraw(float cesh)`: Belirtilen miktarı hesaptan çeker.
- `systemDate(int yıl, int ay, int gün)`: Sistemin geçerli tarihini belirler.
- `systemDatenow()`: Sistemin mevcut tarihini belirler.

ShortTermAccount;

Sınıfını genişleten ve kısa vadeli hesapları temsil etmek için kullanılan bir alt sınıftır. Kısa vadeli hesapların özelliklerini ve işlemlerini içerir

Method tanımları

- deposit(float bakiye): Belirtilen miktarda bakiyeyi hesaba yatırır ve işlem geçmişine kaydeder.
- showWithdraw(float bakiye, float baslangicBakiye): Para çekme işlemini gerçekleştirir. Minimum 1000 TL bakiye şartına uyarak çekilebilir miktarı kontrol eder. İşlem geçmişine kaydeder.
- benefit(): Hesaba faiz ekler. Kısa vadeli hesaplar için günlük faiz hesaplamasını gerçekleştirir.

LongTermAccount;

Sınıfını genişleten ve uzun vadeli hesapları temsil etmek için kullanılan bir alt sınıftır. Uzun vadeli hesapların özelliklerini ve işlemlerini içerir.

Method Tanımları

- deposit(float bakiye): Belirtilen miktarda bakiyeyi hesaba yatırır ve işlem geçmişine kaydeder.
- showWithdraw(float bakiye, float baslangicBakiye): Para çekme işlemini gerçekleştirir. Minimum 1500 TL bakiye şartına uyarak çekilebilir miktarı kontrol eder. İşlem geçmişine kaydeder.
- benefit(): Hesaba faiz ekler. Uzun vadeli hesaplar için günlük faiz hesaplamasını gerçekleştirir.

Special Account;

Sınıfını genişleten ve özel hesapları temsil etmek için kullanılan bir alt sınıftır. Özel hesapların özelliklerini ve işlemlerini içerir

Method tanımları

- showWithdraw(float bakiye, float baslangicBakiye): Para çekme işlemini gerçekleştirir. Minimum hesap açılış bakiyesi şartına uyarak çekilebilir miktarı kontrol eder. İşlem geçmişine kaydeder.

- `deposit(float bakiye)`: Belirtilen miktarda bakiyeyi hesaba yatırır.
- `benefit()`: Hesaba faiz ekler. Özel hesaplar için günlük faiz hesaplamasını gerçekleştirir.

CurrentAccount;

Sınıfını genişleten ve cari hesapları temsil etmek için kullanılan bir alt sınıftır. Cari hesapların özelliklerini ve işlemlerini içerir.

Method Tanımları

- `showWithdraw(float bakiye, float baslangicBakiye)`: Para çekme işlemini gerçekleştirir. Hesaptaki bakiyeyi kontrol ederek çekilebilir miktarı kontrol eder. İşlem geçmişine kaydeder.
- `deposit(float bakiye)`: Belirtilen miktarda bakiyeyi hesaba yatırır. İşlem geçmişine kaydeder.
- `benefit()`: Hesaba faiz eklemeyerek günlük faizi sıfırlar.

Bank;

Sınıfın genel işlevi, hesap açma, hesaba para yatırma, hesaptan para çekme, hesap bilgilerini güncelleme, işlem geçmişini görüntüleme gibi banka işlemlerini gerçekleştirmektedir.

Method Tanımları

- `-showWithdraw(float bakiye, float baslangicBakiye)`: Bu metod, çekme işlemi gerçekleştirildiğinde bakiye ve başlangıç bakiyesini göstermek için kullanılır. Tanımı sınıf içinde bulunmakla birlikte, metod gövdesi boştur ve alt sınıflar tarafından uygulanmalıdır.
- `benefit()`: Bu metod, hesaba ait faiz miktarını hesaplamak için kullanılır. Tanımı sınıf içinde bulunmakla birlikte, metod gövdesi boştur ve alt sınıflar tarafından uygulanmalıdır.
- `hesapac(int op) throws IOException`: Bu metod, kullanıcıya hesap türü seçme imkanı sunar ve kullanıcının seçimine göre ilgili hesap türünü oluşturur. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- `hesapyazdir(Account account) throws IOException`: Bu metod, oluşturulan bir hesap nesnesini kullanarak hesap bilgilerini bir dosyaya yazar. Metod, giriş/çıkış işlemleri

gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.

- hesapidcreator(int id) throws IOException: Bu metod, hesap ID'sini kullanarak bir dosya oluşturur. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- hesapidsorgulama(int op) throws IOException: Bu metod, kullanıcıdan bir hesap ID'si alır ve var olan bir hesap ID'si olup olmadığını kontrol eder. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- deposit() throws IOException: Bu metod, kullanıcıdan bir hesap ID'si alır ve belirtilen hesaba para yatırma işlemini gerçekleştirir. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- hesapbilgi(int id) throws IOException: Bu metod, bir hesap ID'sine göre hesap bilgilerini bir dosyadan okur. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- withdrawcontrol() throws IOException: Bu metod, kullanıcıdan bir hesap ID'si alır ve belirtilen hesaptan para çekme işlemini gerçekleştirir. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- bilgiguncelle(int id) throws IOException: Bu metod, bir hesap ID'sine göre hesap bilgilerini günceller. Metod, giriş/çıkış işlemleri gerçekleştirir ve IOException fırlatma potansiyeline sahiptir.
- tarih(): Bu metod, sistemin geçerli tarihini döndürür.

Main;

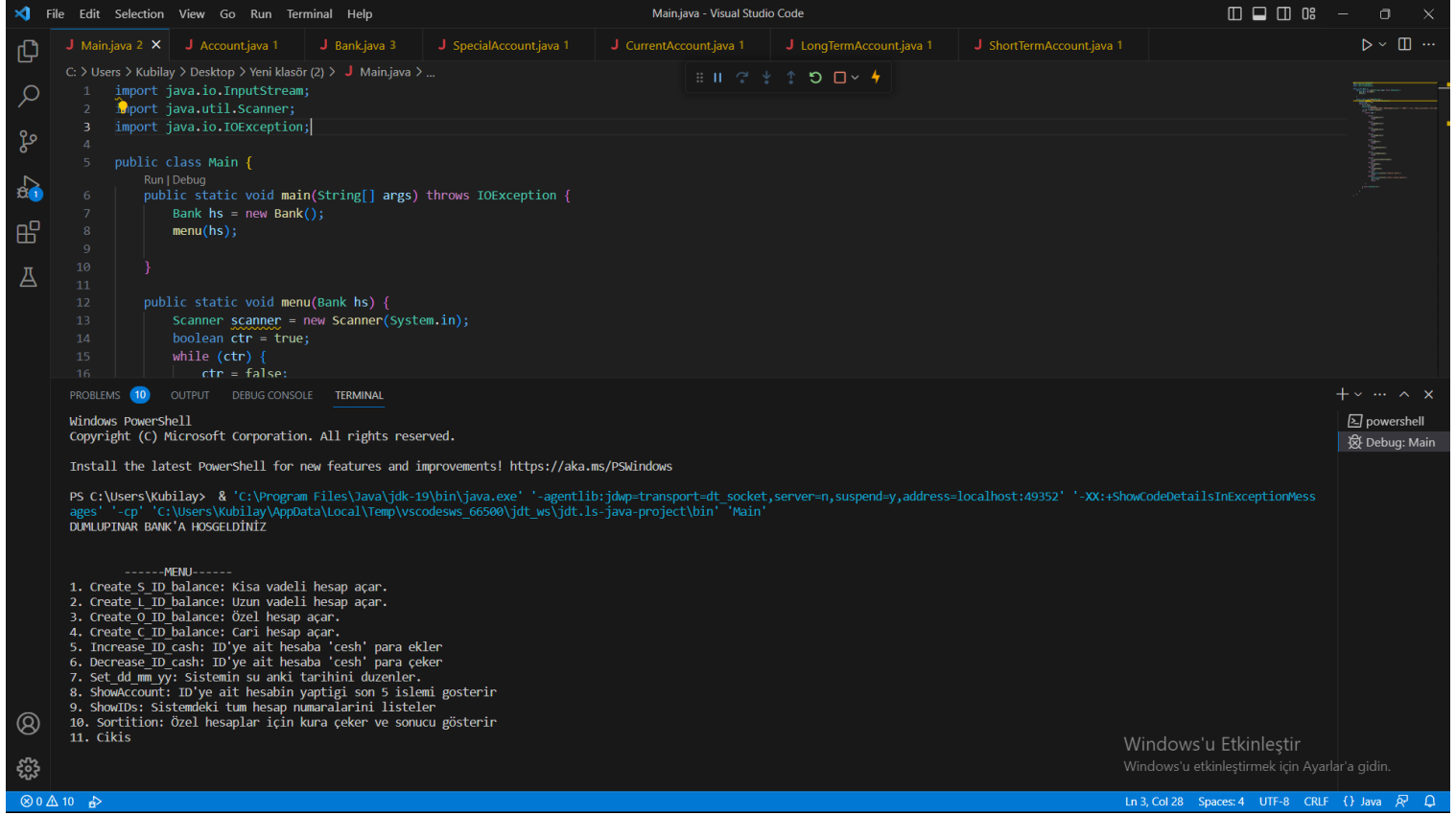
Bu sınıf sadece Bank türünden bir nesne kullanıcıya işlemlerini yapmasını sağlar. Kullanıcıya arayüz sağlar

Method Tanımları

- main(String[] args): Programın başlangıç noktasıdır. `Bank` sınıfından bir örnek oluşturur ve menu() metodunu çağırarak ana menüyü başlatır.
- menu(Bank hs): Kullanıcı arayüzünü sunan ana menüyü oluşturur. Kullanıcıdan bir seçenek girmesini ister ve seçeneğe göre ilgili `Bank` sınıfı metodunu çağırır. Kullanıcı çıkış yapana kadar menüyü tekrar tekrar gösterir.

KOD İLE İLGİLİ GÖRSELLER;

MENÜ



```
File Edit Selection View Go Run Terminal Help
Main.java - Visual Studio Code

J Main.java 2 X J Account.java 1 J Bank.java 3 J SpecialAccount.java 1 J CurrentAccount.java 1 J LongTermAccount.java 1 J ShortTermAccount.java 1

C:\Users\Kubilay\Desktop> Yeni klasör (2) > J Main.java > ...
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     Run | Debug
7     public static void main(String[] args) throws IOException {
8         Bank hs = new Bank();
9         menu(hs);
10    }
11
12    public static void menu(Bank hs) {
13        Scanner scanner = new Scanner(System.in);
14        boolean ctr = true;
15        while (ctr) {
16            ctr = false;
17        }
18    }
19 }

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL
Windows PowerShell
Copyright (c) Microsoft Corporation. All rights reserved.

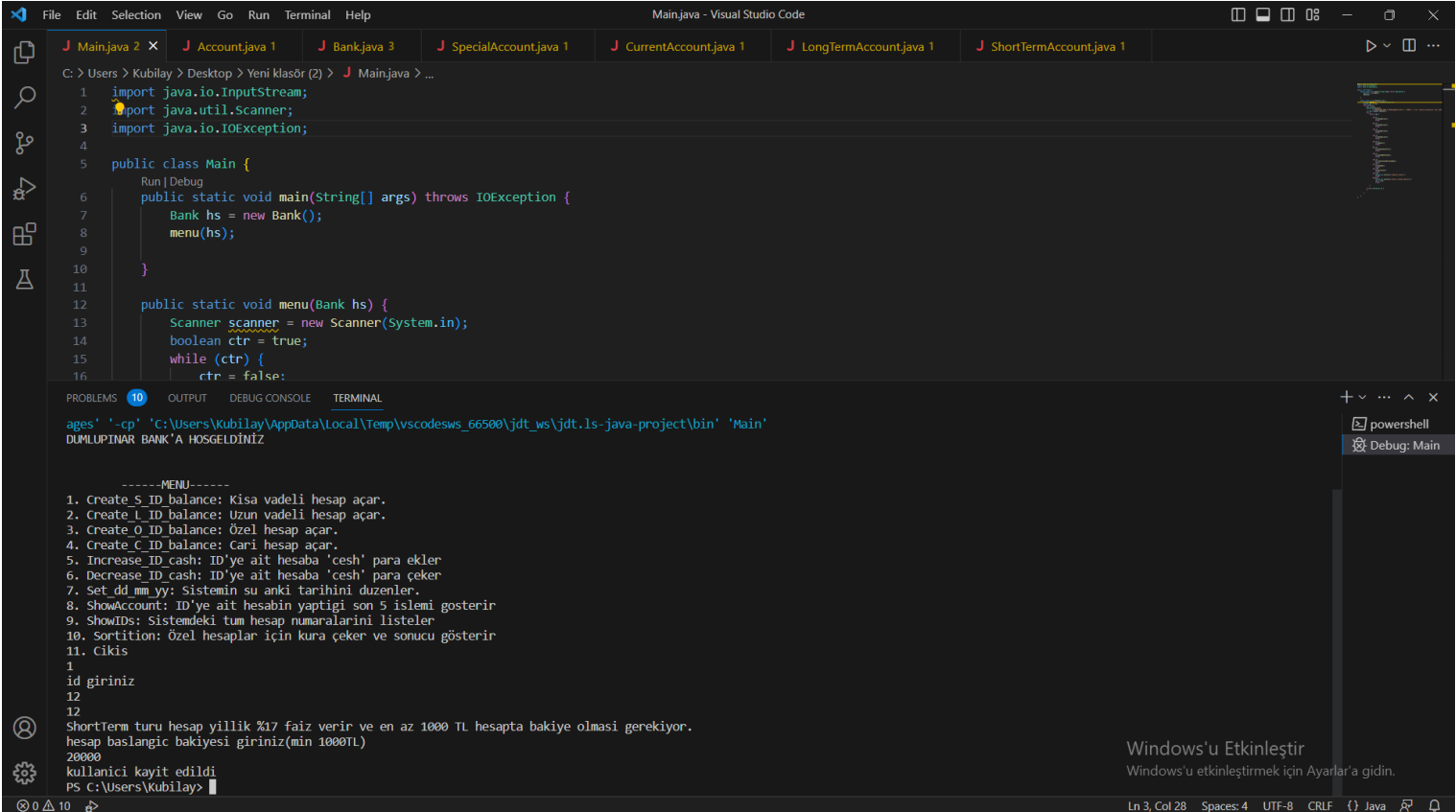
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:49352' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin\' 'Main'
DÜMLÜPİNAR BANK'A HOŞGELDİNİZ

-----MENU-----
1. Create_S_ID balance: Kisa vadeli hesap açar.
2. Create_L_ID balance: Uzun vadeli hesap açar.
3. Create_O_ID balance: Özel hesap açar.
4. Create_C_ID balance: Cari hesap açar.
5. Increase_ID_cash: ID'ye ait hesaba 'cesh' para ekler
6. Decrease_ID_cash: ID'ye ait hesaba 'cesh' para çeker
7. Set_dd_mm_yy: Sistemin su anki tarihini duzenler.
8. ShowAccount: ID'ye ait hesabin yaptigi son 5 islemi gosterir
9. ShowIDs: Sistemdeki tum hesap numaralarini listeler
10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
11. cikis

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.
```

KISA VADELİ HESAP OLUŞTURMA



```
File Edit Selection View Go Run Terminal Help
Main.java - Visual Studio Code

J Main.java 2 X J Account.java 1 J Bank.java 3 J SpecialAccount.java 1 J CurrentAccount.java 1 J LongTermAccount.java 1 J ShortTermAccount.java 1

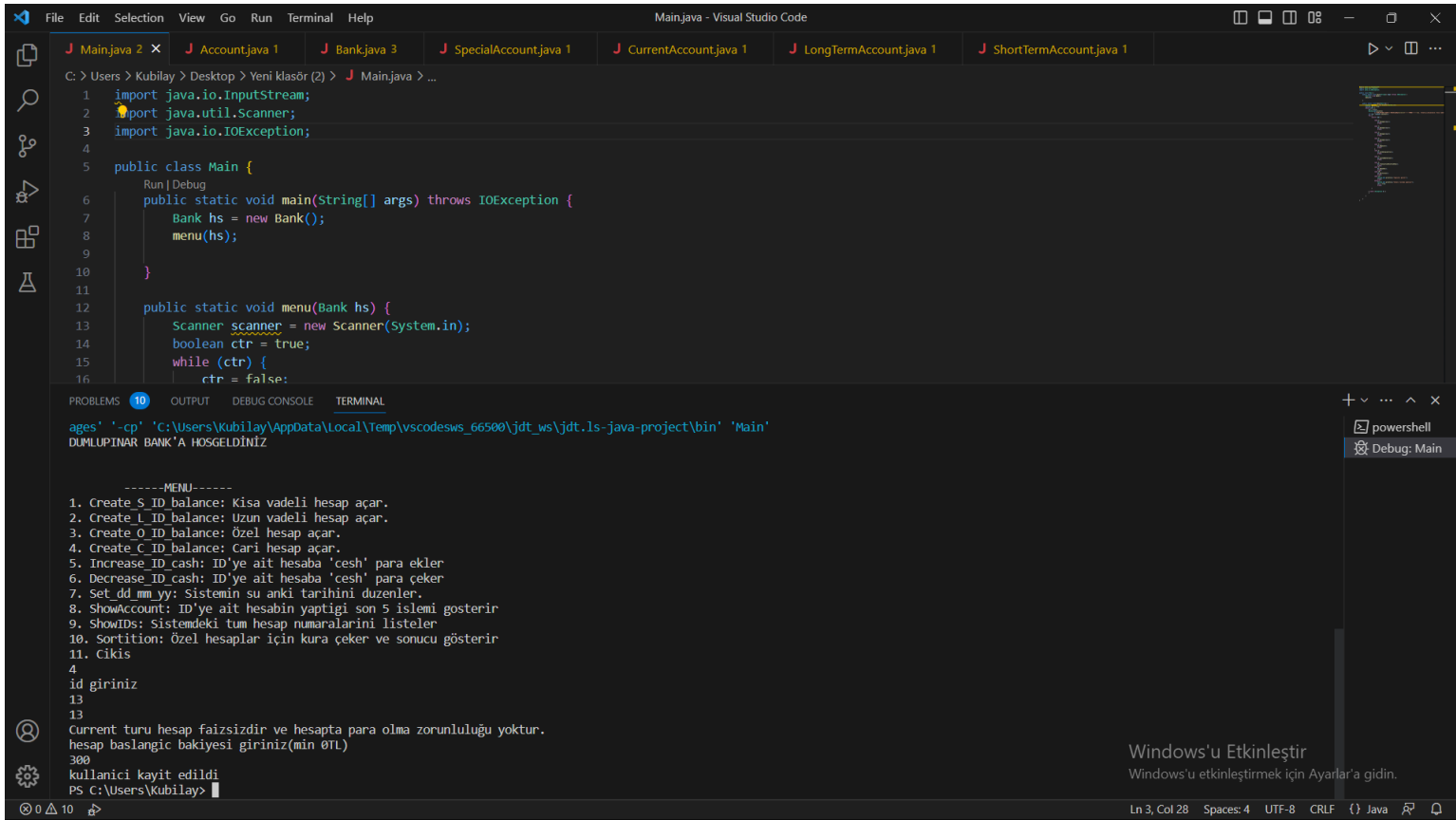
C:\Users\Kubilay\Desktop> Yeni klasör (2) > J Main.java > ...
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     Run | Debug
7     public static void main(String[] args) throws IOException {
8         Bank hs = new Bank();
9         menu(hs);
10    }
11
12    public static void menu(Bank hs) {
13        Scanner scanner = new Scanner(System.in);
14        boolean ctr = true;
15        while (ctr) {
16            ctr = false;
17        }
18    }
19 }

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL
ages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin\' 'Main'
DÜMLÜPİNAR BANK'A HOŞGELDİNİZ

-----MENU-----
1. Create_S_ID balance: Kisa vadeli hesap açar.
2. Create_L_ID balance: Uzun vadeli hesap açar.
3. Create_O_ID balance: Özel hesap açar.
4. Create_C_ID balance: Cari hesap açar.
5. Increase_ID_cash: ID'ye ait hesaba 'cesh' para ekler
6. Decrease_ID_cash: ID'ye ait hesaba 'cesh' para çeker
7. Set_dd_mm_yy: Sistemin su anki tarihini duzenler.
8. ShowAccount: ID'ye ait hesabin yaptigi son 5 islemi gosterir
9. ShowIDs: Sistemdeki tum hesap numaralarini listeler
10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
11. cikis
1
id giriniz
12
12
ShortTerm turu hesap yıllık %17 faiz verir ve en az 1000 TL hesapta bakiye olması gerekiyor.
hesap baslangic bakiyesi giriniz(min 1000TL)
20000
kullanici kayit edildi
PS C:\Users\Kubilay>

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.
```

CARİ HESAP OLUŞTURMA

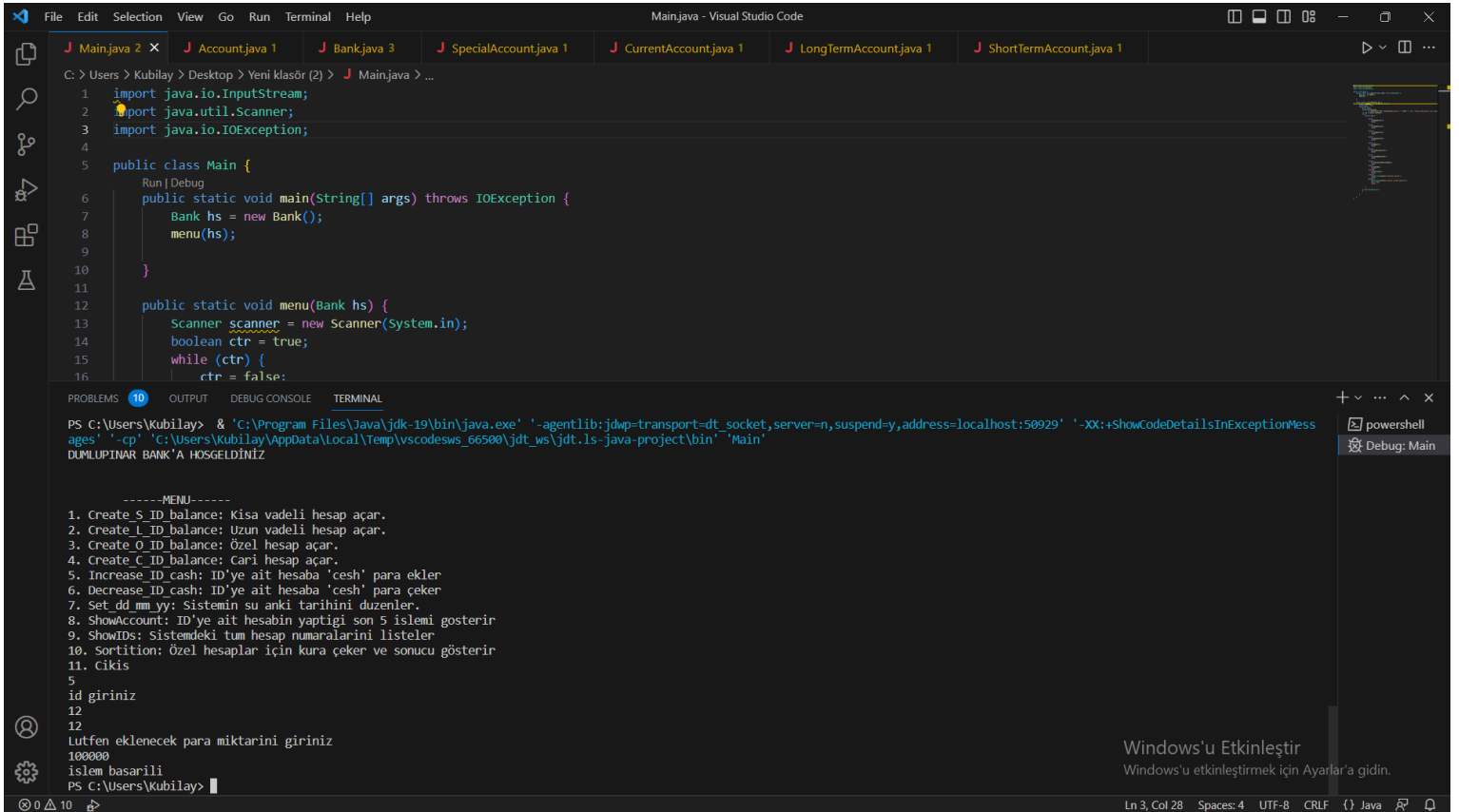


```
File Edit Selection View Go Run Terminal Help
Main.java - Visual Studio Code

J Main.java 2 X J Account.java 1 J Bank.java 3 J SpecialAccount.java 1 J CurrentAccount.java 1 J LongTermAccount.java 1 J ShortTermAccount.java 1

C:\Users\Kubilay> Desktop > Yeni klasör (2) > J Main.java > ...
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     Run | Debug
7     public static void main(String[] args) throws IOException {
8         Bank hs = new Bank();
9         menu(hs);
10    }
11
12    public static void menu(Bank hs) {
13        Scanner scanner = new Scanner(System.in);
14        boolean ctr = true;
15        while (ctr) {
16            ctr = false;
17
18            ages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin' 'Main'
19            DÜMLUPINAR BANK'A HOŞGELDİNİZ
20
21            -----MENU-----
22            1. Create S ID balance: Kisa vadeli hesap açar.
23            2. Create L ID balance: Uzun vadeli hesap açar.
24            3. Create O ID balance: Özel hesap açar.
25            4. Create C ID balance: Cari hesap açar.
26            5. Increase ID cash: ID'ye ait hesaba 'cesh' para ekler
27            6. Decrease ID cash: ID'ye ait hesaba 'cesh' para çeker
28            7. Set dd mm yy: Sistemin su anki tarihini duzenler.
29            8. ShowAccount: ID'ye ait hesabın yaptigi son 5 islemi gosterir
30            9. ShowIDs: Sistemdeki tum hesap numaralarını listeler
31            10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
32            11. Çikis
33            4
34            id giriniz
35            13
36            Current turu hesap faizsizdir ve hesapta para olma zorunluluğu yoktur.
37            hesap baslangic bakiyesi giriniz(min 0TL)
38            300
39            kullanıcı kayıt edildi
40            PS C:\Users\Kubilay> █
```

HESABA PARA YÜKLEME



```
File Edit Selection View Go Run Terminal Help
Main.java - Visual Studio Code

J Main.java 2 X J Account.java 1 J Bank.java 3 J SpecialAccount.java 1 J CurrentAccount.java 1 J LongTermAccount.java 1 J ShortTermAccount.java 1

C:\Users\Kubilay> Desktop > Yeni klasör (2) > J Main.java > ...
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     Run | Debug
7     public static void main(String[] args) throws IOException {
8         Bank hs = new Bank();
9         menu(hs);
10    }
11
12    public static void menu(Bank hs) {
13        Scanner scanner = new Scanner(System.in);
14        boolean ctr = true;
15        while (ctr) {
16            ctr = false;
17
18            PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:50929' '-XX:+ShowCodeDetailsInExceptionMess
19            ages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin' 'Main'
20            DÜMLUPINAR BANK'A HOŞGELDİNİZ
21
22            -----MENU-----
23            1. Create S ID balance: Kisa vadeli hesap açar.
24            2. Create L ID balance: Uzun vadeli hesap açar.
25            3. Create O ID balance: Özel hesap açar.
26            4. Create C ID balance: Cari hesap açar.
27            5. Increase ID cash: ID'ye ait hesaba 'cesh' para ekler
28            6. Decrease ID cash: ID'ye ait hesaba 'cesh' para çeker
29            7. Set dd mm yy: Sistemin su anki tarihini duzenler.
30            8. ShowAccount: ID'ye ait hesabın yaptigi son 5 islemi gosterir
31            9. ShowIDs: Sistemdeki tum hesap numaralarını listeler
32            10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
33            11. Çikis
34            5
35            id giriniz
36            12
37            Lutfen eklenecek para miktarini giriniz
38            100000
39            işlem basarili
40            PS C:\Users\Kubilay> █
```


HESABIN SON 5 İŞLEMİNİ GÖSTERME

The screenshot shows the Visual Studio Code editor with a Java project named 'Main.java'. The code implements a menu-driven application for a bank system. The 'menu' method calls 'showAccount' to display the last 5 transactions. The terminal output shows the program running and displaying the menu options. The 'showAccount' method is implemented to show the last 5 transactions for a given account ID.

```
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     public static void main(String[] args) throws IOException {
7         Bank hs = new Bank();
8         menu(hs);
9     }
10
11     public static void menu(Bank hs) {
12         Scanner scanner = new Scanner(System.in);
13         boolean ctr = true;
14         while (ctr) {
15             // ... menu options ...
16         }
17     }
18
19     public static void showAccount(Bank hs, int id) {
20         // ... implementation ...
21     }
22 }
```

Terminal Output:

```
islem basarili
PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:51137' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin\' 'Main'
DÜMLUPINAR BANK'A HOŞGELDİNİZ

-----MENU-----
1. Create_S_ID balance: Kisa vadeli hesap açar.
2. Create_L_ID balance: Uzun vadeli hesap açar.
3. Create_O_ID balance: Özel hesap açar.
4. Create_C_ID balance: Cari hesap açar.
5. Increase_ID_cash: ID'ye ait hesaba 'cesh' para ekler
6. Decrease_ID_cash: ID'ye ait hesaba 'cesh' para çeker
7. Set_dd_mm_yy: Sistemin su anki tarihini duzenler.
8. ShowAccount: ID'ye ait hesabin yaptigi son 5 islemi gosterir
9. ShowIDs: Sistemdeki tum hesap numaralarini listeler
10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
11. Cikis
id giriniz
12
12
Bakiye: +300.0 2023-06-12
Bakiye: +100000.0 2023-06-12
PS C:\Users\Kubilay>
```

TÜM HESAPLARIN ID'SİNİ GÖSTERME

The screenshot shows the Visual Studio Code editor with the same Java project. The 'showIDs' method is implemented to list all account IDs. The terminal output shows the program running and displaying the menu options. The 'showIDs' method is implemented to list all account IDs.

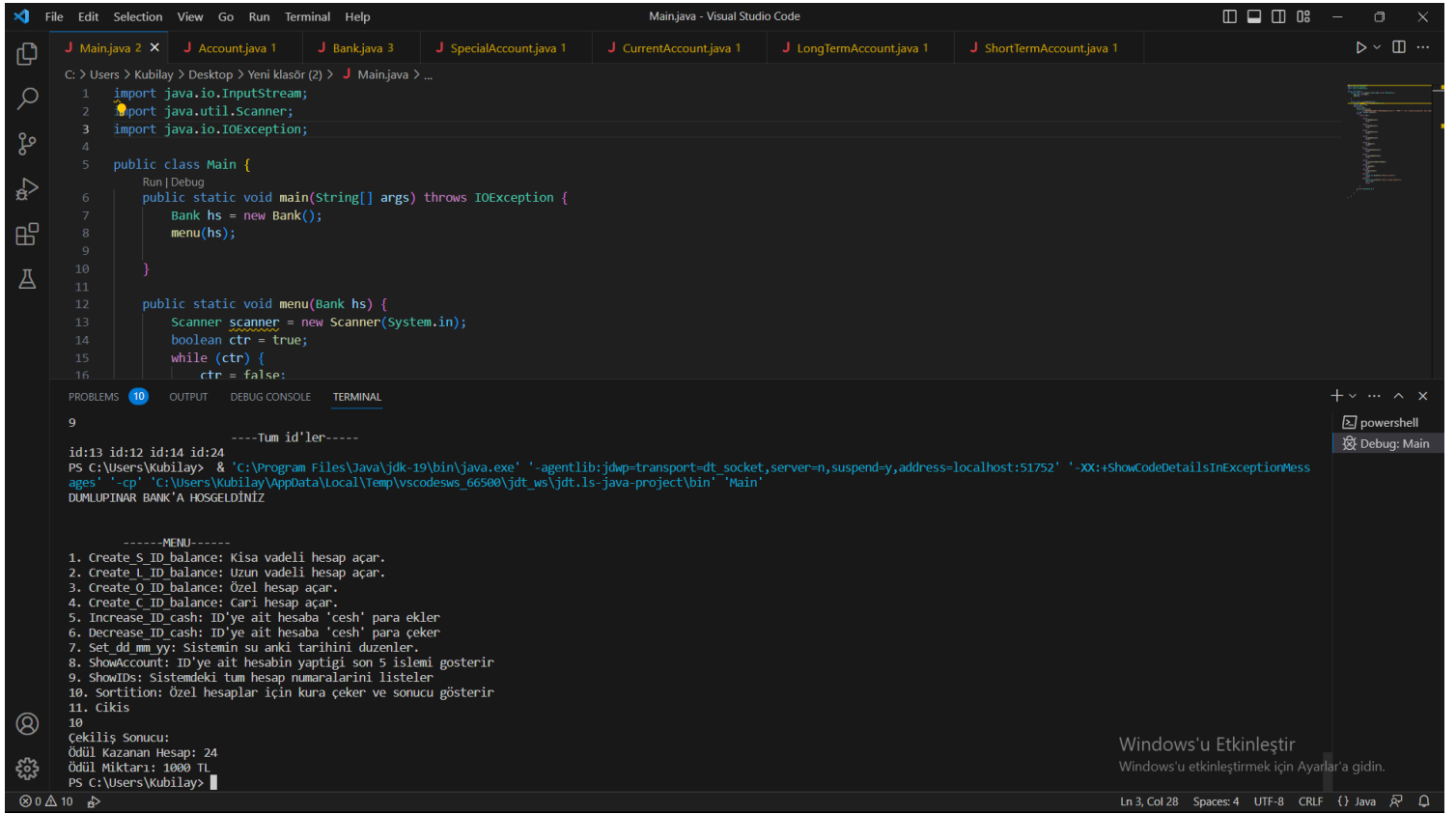
```
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     public static void main(String[] args) throws IOException {
7         Bank hs = new Bank();
8         menu(hs);
9     }
10
11     public static void menu(Bank hs) {
12         Scanner scanner = new Scanner(System.in);
13         boolean ctr = true;
14         while (ctr) {
15             // ... menu options ...
16         }
17     }
18
19     public static void showIDs(Bank hs) {
20         // ... implementation ...
21     }
22 }
```

Terminal Output:

```
Special turu hesap yillik %12 faiz verir ve en az hesap açtigindaki kadar para hesaptabakiye olması gerekiyor.
hesap baslangic bakiyesi giriniz(min 1TL)
400
kullanici kayıt edildi
PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:51543' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin\' 'Main'
DÜMLUPINAR BANK'A HOŞGELDİNİZ

-----MENU-----
1. Create_S_ID balance: Kisa vadeli hesap açar.
2. Create_L_ID balance: Uzun vadeli hesap açar.
3. Create_O_ID balance: Özel hesap açar.
4. Create_C_ID balance: Cari hesap açar.
5. Increase_ID_cash: ID'ye ait hesaba 'cesh' para ekler
6. Decrease_ID_cash: ID'ye ait hesaba 'cesh' para çeker
7. Set_dd_mm_yy: Sistemin su anki tarihini duzenler.
8. ShowAccount: ID'ye ait hesabin yaptigi son 5 islemi gosterir
9. ShowIDs: Sistemdeki tum hesap numaralarini listeler
10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
11. Cikis
9
----Tum id'ler-----
id:13 id:12 id:14 id:24
PS C:\Users\Kubilay>
```

ÇEKİLİŞ YAPMA



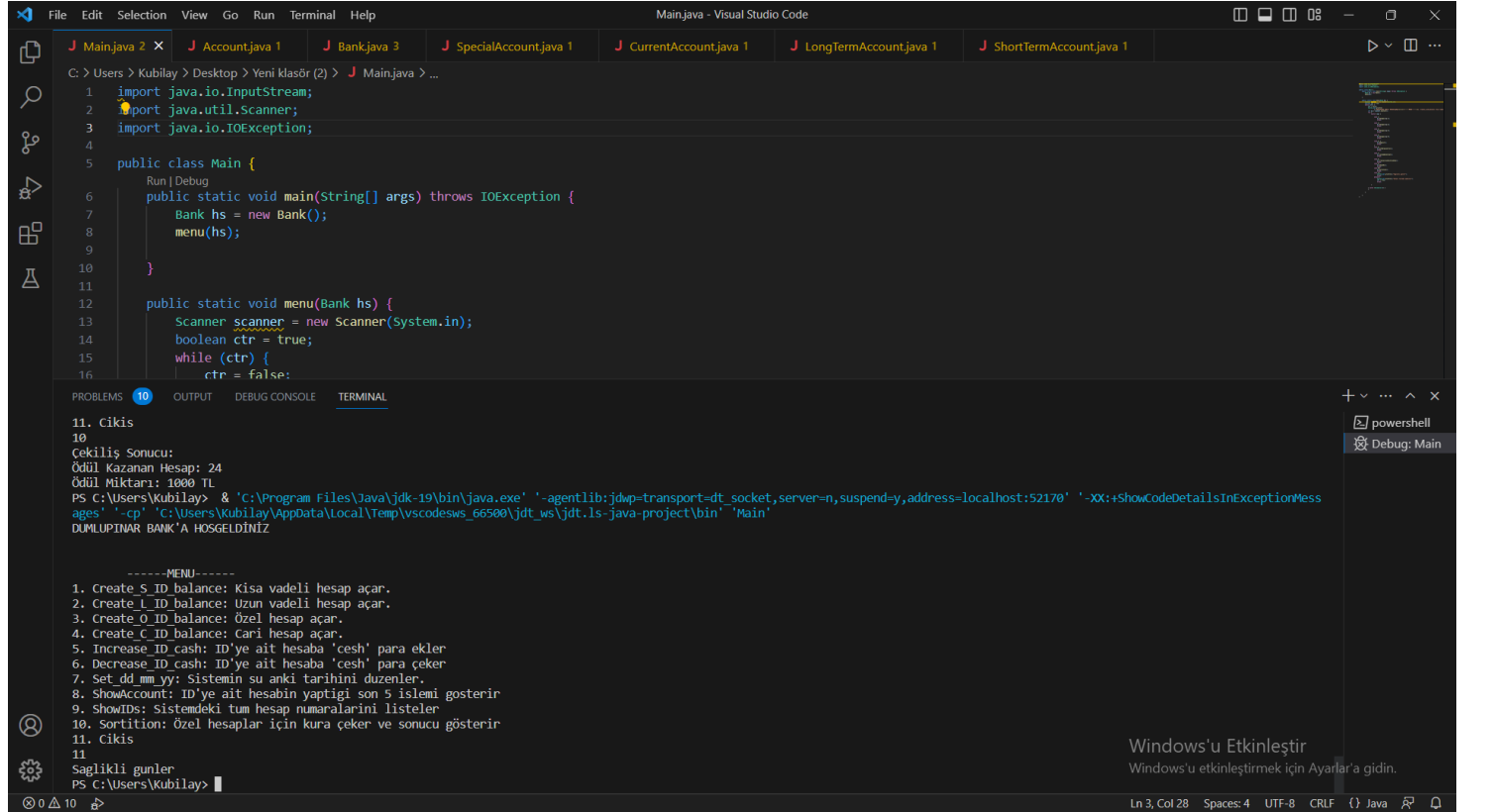
```
File Edit Selection View Go Run Terminal Help
Main.java - Visual Studio Code

C: > Users > Kubilay > Desktop > Yeni klasör (2) > J Main.java > ...
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     public static void main(String[] args) throws IOException {
7         Bank hs = new Bank();
8         menu(hs);
9     }
10
11     public static void menu(Bank hs) {
12         Scanner scanner = new Scanner(System.in);
13         boolean ctr = true;
14         while (ctr) {
15             ctr = false;
16         }
17     }
18 }

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL
9
id:13 id:12 id:14 id:24
PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:51752' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin' 'Main'
DUMLUPTINAR BANK'A HOSGELDİNİZ

-----MENU-----
1. Create S ID balance: Kisa vadeli hesap açar.
2. Create L ID balance: Uzun vadeli hesap açar.
3. Create O ID balance: Özel hesap açar.
4. Create C ID balance: Cari hesap açar.
5. Increase ID cash: ID'ye ait hesaba 'cesh' para ekler
6. Decrease ID cash: ID'ye ait hesaba 'cesh' para çeker
7. Set dd mm yy: Sistemin su anki tarihini duzenler.
8. ShowAccount: ID'ye ait hesabin yaptigi son 5 islemi gosterir
9. ShowIDs: Sistemdeki tum hesap numaralarini listeler
10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
11. Cikis
10
Çekiliş Sonucu:
Ödül Kazanan Hesap: 24
Ödül Miktarı: 1000 TL
PS C:\Users\Kubilay>
```

SİSTEMDEN ÇIKMA



```
File Edit Selection View Go Run Terminal Help
Main.java - Visual Studio Code

C: > Users > Kubilay > Desktop > Yeni klasör (2) > J Main.java > ...
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     public static void main(String[] args) throws IOException {
7         Bank hs = new Bank();
8         menu(hs);
9     }
10
11     public static void menu(Bank hs) {
12         Scanner scanner = new Scanner(System.in);
13         boolean ctr = true;
14         while (ctr) {
15             ctr = false;
16         }
17     }
18 }

PROBLEMS 10 OUTPUT DEBUG CONSOLE TERMINAL
11. Cikis
10
Çekiliş Sonucu:
Ödül Kazanan Hesap: 24
Ödül Miktarı: 1000 TL
PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:52170' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin' 'Main'
DUMLUPTINAR BANK'A HOSGELDİNİZ

-----MENU-----
1. Create S ID balance: Kisa vadeli hesap açar.
2. Create L ID balance: Uzun vadeli hesap açar.
3. Create O ID balance: Özel hesap açar.
4. Create C ID balance: Cari hesap açar.
5. Increase ID cash: ID'ye ait hesaba 'cesh' para ekler
6. Decrease ID cash: ID'ye ait hesaba 'cesh' para çeker
7. Set dd mm yy: Sistemin su anki tarihini duzenler.
8. ShowAccount: ID'ye ait hesabin yaptigi son 5 islemi gosterir
9. ShowIDs: Sistemdeki tum hesap numaralarini listeler
10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
11. Cikis
11
Saglikli gunler
PS C:\Users\Kubilay>
```

ID YANLIŞ GİRİLDİĞİNDE UYARI VERMESİ

The screenshot shows a Visual Studio Code editor with a Java project. The main file is `Main.java`, which contains a `Main` class with a `main` method and a `menu` method. The `main` method creates a `Bank` object and calls the `menu` method. The `menu` method uses a `Scanner` to read user input and performs various actions based on the input. The terminal output shows the execution of the program, displaying a menu with 11 options and the user's input.

```
1 import java.io.InputStream;
2 import java.util.Scanner;
3 import java.io.IOException;
4
5 public class Main {
6     public static void main(String[] args) throws IOException {
7         Bank hs = new Bank();
8         menu(hs);
9     }
10
11     public static void menu(Bank hs) {
12         Scanner scanner = new Scanner(System.in);
13         boolean ctr = true;
14         while (ctr) {
15             ctr = false;
16
17             11
18             Saglikli gunler
19             PS C:\Users\Kubilay> & 'C:\Program Files\Java\jdk-19\bin\java.exe' '-agentlib:jdwp=transport=dt_socket,server=n,suspend=y,address=localhost:52570' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\Kubilay\AppData\Local\Temp\vscodesws_66500\jdt_ws\jdt.ls-java-project\bin\' 'Main'
20             DUMLUPINAR BANK'A HOSGELDİNİZ
21
22             -----MENU-----
23             1. Create_S_ID balance: Kısa vadeli hesap açar.
24             2. Create_L_ID balance: Uzun vadeli hesap açar.
25             3. Create_O_ID balance: Özel hesap açar.
26             4. Create_C_ID balance: Cari hesap açar.
27             5. Increase_ID cash: ID'ye ait hesaba 'cash' para ekler
28             6. Decrease_ID cash: ID'ye ait hesaba 'cash' para çeker
29             7. Set_dd_mm_yy: Sistemin su anki tarihini düzenler.
30             8. ShowAccount: ID'ye ait hesabin yaptığı son 5 işlemi gösterir
31             9. ShowIDs: Sistemdeki tüm hesap numaralarını listeler
32             10. Sortition: Özel hesaplar için kura çeker ve sonucu gösterir
33             11. Cıkis
34             5
35             id giriniz
36             32
37             kullanıcı kayıtlı değil lutfen doğru bir id giriniz
38             id giriniz
```

Windows'u Etkinleştir
Windows'u etkinleştirmek için Ayarlar'a gidin.