****

**INF 212 Algorithms and Programming II**

**2020-2021 Spring**

**Electronic Engineering**

**57\_86\_94\_131\_153**

**P\_Can\_Erdoğan\_Tutar\_Köse\_Koyunçal**

**BankaAplikasyonu**

**Team Members**

|  |  |  |
| --- | --- | --- |
| **School ID** | **Name** | **Surname** |
| **1901022006** | **İrem** | **Can** |
| **1901022038** | **Selen** | **Erdoğan** |
| **1901022050** | **Merve** | **Tutar** |
| **1901022277** | **Mehmet** | **Köse** |
| **200102002090** | **Emircan** | **Koyunçal** |

|  |
| --- |
| PROJECT OBJECTIVE  Briefly describe the objectives of your project |
| Projemizdeki temel amaç, dünyanın her yerinde oldukça sık kullanılan, neredeyse herkesin telefonunda yüklü olan çeşitli banka aplikasyonlarının şu ana kadar öğrendiğimiz konular ile beraber perçinleyip ufak bir kopyasını oluşturmaktır. |
| PROJECT  Briefly describe the project |
| Hazırlamaya çalıştığımız bu uygulamada müşterilere kendi hesaplarına şifreleri sayesinde ulaşmasını sağlayarak tanımlı olan işlemleri yaptırmaya çalıştık. Bu işlemler:  1.Banka hesabi bilgileri goruntule  2.Kredi karti bilgileri goruntule  3.Fatura ode  4.Aylik ozet goruntule  5.Para Transferi  6.Doviz Islemleri olarak altıya ayrılır.  Kendi banka veyahut kredi hesabına ait olan limitler doğrultusunda sınırları olan müşteri yapmak istediği işlem veya işlemleri kolaylıkla yapar. |
| DESIGN  ALGORITHM  Design algorithm of the project |
| 1)Kullanıcı müşteri numarasını ve şifresini girerek kendi hesabına ulaşır.  2)Kullanıcı yapmak istediği işlemi seçer (1.Banka hesabi bilgileri goruntule,2.Kredi karti bilgileri goruntule,3.Fatura ode,4.Aylik ozet goruntule,5.Para Transferi,6.Doviz Islemleri).  3.1)Banka hesabı bilgileri kullanıcıya gösterilir.(Hesap numarasi,Guncel bakiye)  3.2)Kredi kartı bilgileri kullanıcıya gösterilir.(Kart limiti,Guncel borç,Kullanilabir tutar)  3.3)Kullanıcının üzerinde bulunan tüm faturalar kullanım tutarları ve ödenecek bilgileri ile kullanıcıya listelenir. (1-Elektrik 2-Su3-Dogalgaz 4-Internet) Hangi faturayı ödemek istediği kullanıcıya sorulur.  3.3.1)Elektrik faturası için ödenecek tutar bilgisi ve ödeme yöntemi sorulur. (1-Hesap kartı,2-Kredi kartı) İstediği yöntemi seçen kullaniciya banka veyahut kredi kartı limit durumuna göre ödeme işleminin yapılıp yapılmadığı bilgisi döndürülür.  3.3.2.3.4)Yukarıda elektrik faturası için tanımlanan tüm adımlar su doğalgaz ve internet faturaları için de geçerli olur.  4)Aylık özet görüntülemek isteyen kullanıcıya tüm fatura bilgileri tüm kart bilgileri ve tüm güncel döviz ve tl hesap bilgileri döndürülür.  5) Para transferi yapmak isteyen kullanıcıya transfer edilecek müşterinin numarası sorulur. Müşteri numarası doğrulanınca gönderilmek istenen tutar belirlenir. Limit durumlarına göre kullanıcıya işlemin gerçekleştirilip gerçekleştirilemediği bilgisi döndürülür.  6)Döviz işlemleri yapmak isteyen kullanıcıya(1.Doviz Hesabi Bilgi2.Doviz AL/SAT) yapmak istediği işlem sorulur.  6.1)Güncel döviz bilgileri ve kullanıcıya ait güncel döviz hesap bilgileri döndürülür.  6.2)Kullaniciya hangi döviz kuruna ait işlem yapmak istediği sorulduktan sonra alış satış emri verilmesi beklenir. Buna göre yeni kart bakiyeleri güncellenir. |
| UML DIAGRAM  UML diagram of the project |
|  |
| HEADER FILE(s)  Hedeader file(s) of the project |
| FATURA.H  #ifndef FATURA\_H  #define FATURA\_H  #include <iostream>  using namespace std;  class Elektrik{  friend ostream& operator<<(ostream& out,Elektrik &oth);  private:  double oTutar;  int eSayac;//Ne kadar Elektrik kullandigini tutsun  int aboneNo;  double birimf;  public:  Elektrik();  Elektrik(double oTutar, int eSayac, int aboneNo, double birimf);  ~Elektrik();  Elektrik(const Elektrik& oth);  Elektrik& operator=(const Elektrik& oth);  Elektrik& operator+(const Elektrik& oth);//faturaları toplar    void setOtutar(double oTutar);  void setEsayac(int eSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getEsayac();  int getAboneno();  double getBirimf();  };  class Su{  friend ostream& operator<<(ostream& out,Su &oth);  private:  double oTutar;  int suSayac;  int aboneNo;  double birimf;  public:  Su();  Su(double oTutar, int suSayac, int aboneNo, double birimf);  ~Su();  Su(const Su& oth);  Su& operator=(const Su& oth);  Su& operator+(const Su& oth);    void setoTutar(double oTutar);  void setSusayac(int suSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getSusayac();  int getAboneno();  double getBirimf();    };  class Dogalgaz{  friend ostream& operator<<(ostream& out,Dogalgaz &oth);  private:  double oTutar;  int dSayac;  int aboneNo;  double birimf;  public:  Dogalgaz();  Dogalgaz(double oTutar, int dSayac, int aboneNo, double birimf);  ~Dogalgaz();  Dogalgaz(const Dogalgaz& oth);  Dogalgaz& operator+(const Dogalgaz &oth);    void setoTutar(double oTutar);  void setDsayac(int dSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getdsayac();  int getAboneno();  double getBirimf();    };  //class Telefon;  class Internet{  friend ostream& operator<<(ostream& out,Internet &oth);  private:  double oTutar;  int iSayac;  int aboneNo;  double birimf;  public:  Internet();  Internet(double oTutar, int iSayac, int aboneNo, double birimf);  ~Internet();  Internet(const Internet& oth);  Internet& operator+(const Internet &oth);    void setoTutar(double oTutar);  void setisayac(int iSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getisayac();  int getAboneno();  double getBirimf();    };  #endif  KREDIKART.H  #ifndef KREDIKART\_H  #define KREDIKART\_H  #include <iostream>  using namespace std;  //#include "User.h"  class Kart{  friend ostream& operator<<(ostream& os,Kart& obj );  private:  int borc;  int limit;//kredi kartinin limiti - kullanilabilir bakiye tutari hesaplamak icin    public:  Kart();  Kart(int borc,int limit);  ~Kart();  Kart(const Kart& oth);  Kart& operator=(const Kart& oth);  Kart operator+(const Kart& oth);    void setBorc(int yeniBorc);  int getBorc();      int kullanilabilirTutar();  void setLimit(int limit);  int getLimit();    };  #endif  USER.H  #ifndef USER\_H  #define USER\_H  #include <iostream>  #include <string>  #include "KrediKart.h"  #include "Fatura.h"  #include "doviz.h"  using namespace std;  class User{  friend ostream& operator<<(ostream& out,User &oth);  friend void showKullaniciBilgi(User &oth);  friend void showKrediKartiBilgi(User &oth);  friend void showDovizBilgi(User &oth);  friend void operator+( User& oth);  private:  int musNo;  string musName;  string musSname;  int hesapNo;  int password;  double bakiye;  double dolar\_bakiye;  double euro\_bakiye;  double toplamFatura;    public:  int pwordhak;  User();  User(int musNo, string musName, string musSname, int hesapNo, int password, double bakiye, int hak, double dolar, double euro);  ~User();  User(const User& oth);  User& operator=(const User& oth);      void setmusNo(int yeniNo);  void setmusName(string yeniName);  void setmusSname(string yeniSname);  void setHno(int yeniNo);  void setPw(int yenipw);  void setToplamFatura(double );    void setBakiye(double yenib);  void ekleBakiye(int ekleb);  void cekBakiye(int cekb);  void setdBakiye(double yenid);  void seteBakiye(double yenie);      int getmusNo();  string getmusName();  string getmusSname();  int getHno();  int getPw();  double getBakiye();  int ispw(int sifre);  double getDolarb();  double getEurob();  double getToplamFatura();    Kart k;  Elektrik e;  Su s;  Dogalgaz d;  Internet i;  doviz dov;  };  #endif  DOVIZ.H  #include <iostream>  using namespace std;  class doviz{  private:  double dolar;  double euro;  public:  doviz();  doviz(double \_dolar, double \_euro);  doviz(const doviz& y);  doviz& operator =(const doviz& y);  doviz& operator +(doviz& y);    void set();  doviz& get();  double getdolar();  double geteuro();  }; |
| TESTING  Briefly describe how you test your code |
| Kullancıya hesabına oluşması için gerekli bilgiler sorulur.    Eğer girilen bilgiler doğruysa kullanıcı hesabına erişim sağlanır. Eğer doğru değilse doğru bilgiler alınana kadar kullanıcıya gerekli sorular sorulur.    Banka hesap bilgilerini görüntülemek isteyen kullanıcıya:    Kredi kartı bilgilerini görüntülemek isteyen kullanıcıya:    Aylık özet seçeneğini seçen kullanıcıya:    Fatura ödemek isteyen kullanıcıya:    Elektrik faturası ödemek isteyen kullanıcıya : (KREDİ KARTI)    Elektrik faturası ödemek isteyen kullanıcıya: (HESAP KARTI)    (aynı işlemler tüm fatura ödeme işlemleri için geçerlidir.)  Tüm faturaları veyahut birden fazla faturayı ödeyen kullanıcıya aylık özet geri dönüşü:    Ödenen faturalar sıfırlanır ve aylık toplam fatura tutarından gerekli eksilmeler yapılır.  Para transferi yapmak isteyen kullanıcıya:    (Göndermek istenen tutar girilir.Ödeme yöntemi seçilir. Limit durumuna göre kullanıcıya gerekli bilgiler döndürülür.)  Döviz işlemleri yapmak isteyen kullanıcıya:    Döviz alıp satmak isteyen kullanıcıya: |
| USER’S GUIDE  (Describe briefly how the user can use your program (input parameters of program, output of program, possible limitations, possible errors, etc.) |
| Kullanıcı müşteri numarasını girerek programı kullanmaya başlar. Müşteri numarası doğruysa program şifre ister. Kullanıcın 3 ADET şifre girme hakkı mevcuttur. Üç adetten fazla yanlış şifre giren kullanıcı:    Eğer kullanıcı doğru bilgileri girdiyse program kullanıcıya ana menüyü çıktı verir.  Yapmak istediği işlemi seçen kullanıcı girdiği seçeneğe uygun olarak işlem yapmaya başlar. Kullanıcı her zaman ana menüye ulaşma ve programı bitirme seçeneğine sahiptir. |
| CONCLUSION AND REMARKS  You can or can(not) put any remarks on your work. For example;  is your program works or not, if not why  which troubles during performance of project you faced with and how you overcome them;  is your program user friendly and how you can improve it;  what you achieved during performance of project; and so |
| Program istenildiği gibi çalışmaktadır. Bu projeyi oluştururken ilk zorlandığımız mesele her kullanıcıya ait gerekli olan tüm bilgileri tanımlamakta oldu. Bu meseleyi #include <stdlib.h> kütüphanesindeki rand() fonksiyonunu kullanarak her kullanıcıya banka ve kredi kartı bakiye, borç, limit, kullanılan elektrik su doğalgaz internet tutarlarını random olarak atayarak hallettik. Fatura tutarları, atanan kullanım tutarları doğrultusunda sabit değerler ile çarpılarak hesaplandı.  Menüyü oluşturuken kullanıcıya anlaşılır bir yapı kurmaya özen gösterdik. Harcanan tutarlar, bakiyeye eklenen tutarlar anlaşılır bir şekilde çıktı verildi. |
| REFERENCES  Put the list of references and sources (books, links to websites, videos, etc), which you used for project. |
| * Proje 3(calorieTracker) * <https://stackoverflow.com/> * <https://www.tutorialspoint.com/index.htm> * <https://www.geeksforgeeks.org/> * <https://www.yusufsezer.com.tr/> * <https://www.mustafayemural.com/uml-egitimi/> |

|  |
| --- |
| APPENDIX |
| PROJECT CODES  Put here code of your project. |
| FATURA.CPP  #include <iostream>  #include <string>  #include "Fatura.h"  //ELEKTRİK  Elektrik::Elektrik(){  this->oTutar=0.0;  this->eSayac=0;  this->aboneNo=0;  this->birimf=2.1;//SABIT  }  Elektrik::Elektrik(double oTutar, int eSayac, int aboneNo, double birimf){  this->oTutar=oTutar;  this->eSayac=eSayac;  this->aboneNo=aboneNo;  this->birimf=birimf;  }  Elektrik::Elektrik(const Elektrik& oth){  oTutar=oth.oTutar;  eSayac=oth.eSayac;  aboneNo=oth.aboneNo;  birimf=oth.birimf;  }  Elektrik& Elektrik::operator=(const Elektrik& oth){  if(this!=&oth){  oTutar=oth.oTutar;  eSayac=oth.eSayac;  aboneNo=oth.aboneNo;  birimf=oth.birimf;  }  return \*this;  }  Elektrik Elektrik::operator+(const Elektrik& oth){  Elektrik temp;  temp.oTutar=oTutar+oth.oTutar;  return temp;  }  void Elektrik::setOtutar(double oTutar){  this->oTutar=oTutar;  }  void Elektrik::setEsayac(int eSayac){  this->eSayac=eSayac;  }  void Elektrik::setAboneno(int aboneNo){  this->aboneNo=aboneNo;  }  void Elektrik::setBirimf(double birimf){  this->birimf=birimf;  }  double Elektrik::getOtutar(){  return this->oTutar;  }  int Elektrik::getEsayac(){  return this->eSayac;  }  int Elektrik::getAboneno(){  return this->aboneNo;  }  double Elektrik::getBirimf(){  return this->birimf;  }  ostream& operator<<(ostream& out,Elektrik &oth){  cout<<"\n>Fatura Bilgileri<"<<endl;  cout<<oth.getAboneno()<<" Numarali Abone\nElektrik Faturasi Guncel Durum"<<endl;  cout<<"Kullanilan Elektrik -> "<<oth.getEsayac()<<"kw "<<endl;  cout<<"Odenecek Tutar -> "<<oth.getOtutar()<<endl;cout<<endl;  }  //SU  Su::Su(){  this->oTutar=0.0;;  this->suSayac=0;  this->aboneNo=0;  this->birimf=5.1;//SABIT    }  Su::Su(double oTutar, int suSayac, int aboneNo, double birimf){  this->oTutar=oTutar;  this->suSayac=suSayac;  this->aboneNo=aboneNo;  this->birimf=birimf;  }  Su::Su(const Su& oth){  oTutar=oth.oTutar;  suSayac=oth.suSayac;  aboneNo=oth.aboneNo;  birimf=oth.birimf;  }  Su& Su::operator=(const Su& oth){  if(this!=&oth){  oTutar=oth.oTutar;  suSayac=oth.suSayac;  aboneNo=oth.aboneNo;  birimf=oth.birimf;  }  return \*this;  }  Su Su::operator+(const Su& oth){  Su temp;  temp.oTutar=oTutar+oth.oTutar;  return temp;  }  void Su::setoTutar(double oTutar){  this->oTutar=oTutar;  }  void Su::setSusayac(int suSayac){  this->suSayac=suSayac;  }  void Su::setAboneno(int aboneNo){  this->aboneNo=aboneNo;  }  void Su::setBirimf(double birimf){  this->birimf=birimf;  }  double Su::getOtutar(){  return oTutar;  }  int Su::getSusayac(){  return this->suSayac;  }  int Su::getAboneno(){  return this->aboneNo;  }  double Su::getBirimf(){  return this->birimf;  }  ostream& operator<<(ostream& out,Su &oth){  cout<<oth.getAboneno()<<" Numarali Abone\nSu Faturasi Guncel Durum"<<endl;  cout<<"Kullanilan Su -> "<<oth.getSusayac()<<"L "<<endl;  cout<<"Odenecek Tutar -> "<<oth.getOtutar()<<endl;cout<<endl;  }  //DOĞALGAZ  Dogalgaz::Dogalgaz(){  oTutar=0.0;  dSayac=0;  aboneNo=0;  birimf=11.3;  }  Dogalgaz::Dogalgaz(double oTutar, int dSayac, int aboneNo,double birimf){  this->oTutar=oTutar;  this->dSayac=dSayac;  this->aboneNo=aboneNo;  this->birimf=birimf;  }  Dogalgaz::Dogalgaz(const Dogalgaz& oth){  oTutar=oth.oTutar;  dSayac=oth.dSayac;  aboneNo=oth.aboneNo;  birimf=oth.birimf;  }  Dogalgaz Dogalgaz::operator+(const Dogalgaz &oth){  Dogalgaz temp;  temp.oTutar=oTutar+oth.oTutar;  return temp;  }  void Dogalgaz::setoTutar(double oTutar){  this->oTutar=oTutar;  }  void Dogalgaz::setDsayac(int dSayac){  this->dSayac=dSayac;  }  void Dogalgaz::setAboneno(int aboneNo){  this->aboneNo=aboneNo;  }  void Dogalgaz::setBirimf(double birimf){  this->birimf=birimf;  }  double Dogalgaz::getOtutar(){  return this->oTutar;  }  int Dogalgaz::getdsayac(){  return this->dSayac;  }  int Dogalgaz::getAboneno(){  return this->aboneNo;  }  double Dogalgaz::getBirimf(){  return this->birimf;  }  ostream& operator<<(ostream& out,Dogalgaz &oth){  cout<<oth.getAboneno()<<" Numarali Abone\nDogalgaz Faturasi Guncel Durum"<<endl;  cout<<"Kullanilan Dogalgaz -> "<<oth.getdsayac()<<"mkup "<<endl;  cout<<"Odenecek Tutar -> "<<oth.getOtutar()<<endl;cout<<endl;  }  //İNTERNET  Internet::Internet(){  oTutar=0.0;  iSayac=0;  aboneNo=0;  birimf=6.2;//SABIT  }  Internet::Internet(double oTutar, int iSayac, int aboneNo, double birimf){  this->oTutar=oTutar;  this->iSayac=iSayac;  this->aboneNo=aboneNo;  this->birimf=birimf;  }  Internet::Internet(const Internet& oth){  oTutar=oth.oTutar;  iSayac=oth.iSayac;  aboneNo=oth.aboneNo;  birimf=oth.birimf;  }  Internet Internet::operator+(const Internet &oth){  Internet temp;  temp.oTutar=oTutar+oth.oTutar;  return temp;  }  void Internet::setoTutar(double oTutar){  this->oTutar=oTutar;  }  void Internet::setisayac(int iSayac){  this->iSayac=iSayac;  }  void Internet::setAboneno(int aboneNo){  this->aboneNo=aboneNo;  }  void Internet::setBirimf(double birimf){  this->birimf=birimf;  }  double Internet::getOtutar(){  return this->oTutar;  }  int Internet::getisayac(){  return this->iSayac;  }  int Internet::getAboneno(){  return this->aboneNo;  }  double Internet::getBirimf(){  return this->birimf;  }  ostream& operator<<(ostream& out,Internet &oth){  cout<<oth.getAboneno()<<" Numarali Abone\nInternet Faturasi Guncel Durum"<<endl;  cout<<"Kullanilan Internet -> "<<oth.getisayac()<<"gb "<<endl;  cout<<"Odenecek Tutar -> "<<oth.getOtutar()<<endl;cout<<endl;  }  Internet::~Internet(){}  Elektrik::~Elektrik(){}  Dogalgaz::~Dogalgaz(){}  Su::~Su(){}  FATURA.H  #ifndef FATURA\_H  #define FATURA\_H  #include <iostream>  using namespace std;  class Elektrik{  friend ostream& operator<<(ostream& out,Elektrik &oth);  private:  double oTutar;  int eSayac;//Ne kadar Elektrik kullandigini tutsun  int aboneNo;  double birimf;  public:  Elektrik();  Elektrik(double oTutar, int eSayac, int aboneNo, double birimf);  ~Elektrik();  Elektrik(const Elektrik& oth);  Elektrik& operator=(const Elektrik& oth);  Elektrik operator+(const Elektrik& oth);//faturaları toplar    void setOtutar(double oTutar);  void setEsayac(int eSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getEsayac();  int getAboneno();  double getBirimf();  };  class Su{  friend ostream& operator<<(ostream& out,Su &oth);  private:  double oTutar;  int suSayac;  int aboneNo;  double birimf;  public:  Su();  Su(double oTutar, int suSayac, int aboneNo, double birimf);  ~Su();  Su(const Su& oth);  Su& operator=(const Su& oth);  Su operator+(const Su& oth);    void setoTutar(double oTutar);  void setSusayac(int suSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getSusayac();  int getAboneno();  double getBirimf();    };  class Dogalgaz{  friend ostream& operator<<(ostream& out,Dogalgaz &oth);  private:  double oTutar;  int dSayac;  int aboneNo;  double birimf;  public:  Dogalgaz();  Dogalgaz(double oTutar, int dSayac, int aboneNo, double birimf);  ~Dogalgaz();  Dogalgaz(const Dogalgaz& oth);  Dogalgaz operator+(const Dogalgaz &oth);    void setoTutar(double oTutar);  void setDsayac(int dSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getdsayac();  int getAboneno();  double getBirimf();    };  //class Telefon;  class Internet{  friend ostream& operator<<(ostream& out,Internet &oth);  private:  double oTutar;  int iSayac;  int aboneNo;  double birimf;  public:  Internet();  Internet(double oTutar, int iSayac, int aboneNo, double birimf);  ~Internet();  Internet(const Internet& oth);  Internet operator+(const Internet &oth);    void setoTutar(double oTutar);  void setisayac(int iSayac);  void setAboneno(int aboneNo);  void setBirimf(double birimf);    double getOtutar();  int getisayac();  int getAboneno();  double getBirimf();    };  #endif  KREDIKART.CPP  #include "KrediKart.h"  ostream& operator<<(ostream& os,Kart& obj ){  os<<"kredi kartinizdaki kullanilabilir tutar: "<<obj.kullanilabilirTutar();  return os;  }    int Kart::kullanilabilirTutar(){  return this->getLimit() - this->getBorc() ;  }  Kart::Kart(){  borc=0;  limit=0;  }  Kart::~Kart(){}  Kart::Kart(int borc, int limit){  this->borc=borc;  this->limit=limit;  }    Kart::Kart(const Kart &oth){  borc=oth.borc;  limit=oth.limit;  }    Kart& Kart::operator=(const Kart &oth){  borc=oth.borc;  limit=oth.limit;  return \*this;  }    Kart Kart::operator+(const Kart &oth){  Kart temp;  temp.borc=borc+ oth.borc;  return temp;  }    void Kart::setBorc(int yeniBorc){  borc=yeniBorc;  }    int Kart::getBorc(){  return borc;  }    void Kart::setLimit(int limit){  this->limit=limit;  }    int Kart::getLimit(){  return limit;  }  KREDIKART.H  #ifndef KREDIKART\_H  #define KREDIKART\_H  #include <iostream>  using namespace std;  class Kart{  friend ostream& operator<<(ostream& os,Kart& obj );  private:  int borc;  int limit;//kredi kartinin limiti - kullanilabilir bakiye tutari hesaplamak icin    public:  Kart();  Kart(int borc,int limit);  ~Kart();  Kart(const Kart& oth);  Kart& operator=(const Kart& oth);  Kart operator+(const Kart& oth);    void setBorc(int yeniBorc);  int getBorc();      int kullanilabilirTutar();  void setLimit(int limit);  int getLimit();    };  #endif  USER.CPP  #include "User.h"  User::User()  {  this->musNo=0;  this->musName="Emptyn";  this->musSname="Emptys";  this->hesapNo=0;  this->password=4444;  this->bakiye=0;  dolar\_bakiye = 0;  euro\_bakiye = 0;  pwordhak = 3;    }  User::User(int musNo, string musName, string musSname, int hesapNo, int password, double bakiye, int hak, double dolar, double euro)  {  this->musNo=musNo;  this->musName=musName;  this->musSname=musSname;  this->hesapNo=hesapNo;  this->password=password;  this->bakiye=bakiye;  dolar\_bakiye = dolar;  euro\_bakiye = euro;  pwordhak = hak;  }  User::~User(){}  User::User(const User& oth)  {  this->musNo=oth.musNo;  this->musName=oth.musName;  this->musSname=oth.musSname;  this->hesapNo=oth.hesapNo;  this->password=oth.password;  this->bakiye=oth.bakiye;  dolar\_bakiye = oth.dolar\_bakiye;  euro\_bakiye = oth.euro\_bakiye;  pwordhak = oth.pwordhak;  }  User& User::operator=(const User& oth)  {  if(this != &oth){  this->musNo=oth.musNo;  this->musName=oth.musName;  this->musSname=oth.musSname;  this->hesapNo=oth.hesapNo;  this->password=oth.password;  this->bakiye=oth.bakiye;  dolar\_bakiye = oth.dolar\_bakiye;  euro\_bakiye = oth.euro\_bakiye;  pwordhak = oth.pwordhak;    }  return \*this;  }  void operator+( User& oth)  {  //User temp;  //oth.toplamFatura= oth.e.getOtutar() + oth.d.getOtutar() + oth.s.getOtutar()+ oth.i.getOtutar();  oth.setToplamFatura(oth.e.getOtutar() + oth.d.getOtutar() + oth.s.getOtutar()+ oth.i.getOtutar());  cout <<"\n\n\tBu ay icerisinde yapmaniz gereken toplam fatura tutari: " << oth.getToplamFatura() <<endl;  }  void User::setToplamFatura(double fatura){this->toplamFatura=fatura;}  void User::setmusNo(int yeniNo){this->musNo=yeniNo;}  void User::setmusName(string yeniName){this->musName=yeniName;}  void User::setmusSname(string yeniSname){this->musSname=yeniSname;}  void User::setHno(int yeniNo){this->hesapNo=yeniNo;}  void User::setPw(int yenipw){this->password=yenipw;}  void User::setBakiye(double yenib){this->bakiye=yenib;}  void User::ekleBakiye(int ekleb){this->bakiye+=ekleb;}  void User::cekBakiye(int cekb){  if(this->bakiye > cekb){  this->bakiye -=cekb;  }  }  void User::setdBakiye(double yenid){this->dolar\_bakiye=yenid;}  void User::seteBakiye(double yenie){this->euro\_bakiye=yenie;}  int User::getmusNo(){return this->musNo;}  string User::getmusName(){return this->musName;}  string User::getmusSname(){return this->musSname;}  int User::getHno(){return this->hesapNo;}  int User::getPw(){return this->password;}  double User::getBakiye(){return this->bakiye;}  double User::getDolarb(){return this->dolar\_bakiye;}  double User::getEurob(){return this->euro\_bakiye;}  double User::getToplamFatura(){return this->toplamFatura;}  void showKullaniciBilgi(User &oth)  {  cout<<oth.getmusNo()<<" Numarali Musteri Bilgileri\n"<<endl;  cout<<"Ad-Soyad -> "<<oth.getmusName()<<" "<<oth.getmusSname()<<endl;  cout<<"Hesap Numarasi ->"<<oth.getHno()<<endl;  cout<<"Guncel Bakiye -> "<<oth.getBakiye()<<endl;cout<<endl;  }  void showKrediKartiBilgi(User &oth)  {  cout<<"Kredi Karti Guncel Durum\n"<<endl;  cout<<"Kart Limiti -> "<<oth.k.getLimit()<<endl;  cout<<"Guncel Borc -> "<<oth.k.getBorc()<<endl;  cout<<"Kullanilabilir Tutar -> "<<oth.k.kullanilabilirTutar()<<endl;  }  void showDovizBilgi(User &oth)  {  cout<<"Doviz Hesabi Guncel Durum\n"<<endl;  cout<<"Dolar Bakiyesi -> "<<oth.getDolarb()<<endl;  cout<<"Guncel Dolar Kuru -> "<<oth.dov.getdolar()<<endl;cout<<endl;  cout<<"Euro Bakiyesi -> "<<oth.getEurob()<<endl;  cout<<"Guncel Euro Kuru -> "<<oth.dov.geteuro()<<endl;    }  ostream& operator<<(ostream& out, User& oth)  {  out<<oth.getmusNo()<<" Numarali Musteri Bilgileri\n"<<endl;  out<<"Ad-Soyad -> "<<oth.getmusName()<<" "<<oth.getmusSname()<<endl;  out<<"Hesap Numarasi ->"<<oth.getHno()<<endl;  out<<"Guncel Bakiye -> "<<oth.getBakiye()<<endl;cout<<endl;    out<<"Kredi Karti Guncel Durum\n"<<endl;  out<<"Kart Limiti -> "<<oth.k.getLimit()<<endl;  out<<"Guncel Borc -> "<<oth.k.getBorc()<<endl;  out<<"Kullanilabilir Tutar -> "<<oth.k.kullanilabilirTutar()<<endl;    out<<"\n>Fatura Bilgileri<"<<endl;  out<<oth.e.getAboneno()<<" Numarali Abone\nElektrik Faturasi Guncel Durum"<<endl;  out<<"Kullanilan Elektrik -> "<<oth.e.getEsayac()<<"kw "<<endl;  out<<"Odenecek Tutar -> "<<oth.e.getOtutar()<<endl;cout<<endl;    out<<oth.s.getAboneno()<<" Numarali Abone\nSu Faturasi Guncel Durum"<<endl;  out<<"Kullanilan Su -> "<<oth.s.getSusayac()<<"L "<<endl;  out<<"Odenecek Tutar -> "<<oth.s.getOtutar()<<endl;cout<<endl;    out<<oth.d.getAboneno()<<" Numarali Abone\nDogalgaz Faturasi Guncel Durum"<<endl;  out<<"Kullanilan Dogalgaz -> "<<oth.d.getdsayac()<<"mkup "<<endl;  out<<"Odenecek Tutar -> "<<oth.d.getOtutar()<<endl;cout<<endl;    out<<oth.i.getAboneno()<<" Numarali Abone\nInternet Faturasi Guncel Durum"<<endl;  out<<"Kullanilan Internet -> "<<oth.i.getisayac()<<"gb "<<endl;  out<<"Odenecek Tutar -> "<<oth.i.getOtutar()<<endl;cout<<endl;    out<<"Doviz Hesabi Guncel Durum\n"<<endl;  out<<"Dolar Bakiyesi -> "<<oth.getDolarb()<<endl;  out<<"Guncel Dolar Kuru -> "<<oth.dov.getdolar()<<endl;cout<<endl;  out<<"Euro Bakiyesi -> "<<oth.getEurob()<<endl;  out<<"Guncel Euro Kuru -> "<<oth.dov.geteuro()<<endl;cout<<endl;    return out;  }  int User::ispw(int sifre)  {  if ( password == sifre)  {  return 1;  }  else{  pwordhak--;  return 0;  }  }  USER.H  #ifndef USER\_H  #define USER\_H  #include <iostream>  #include <string>  #include "KrediKart.h"  #include "Fatura.h"  #include "doviz.h"  using namespace std;  class User{  friend ostream& operator<<(ostream& out,User &oth);  friend void showKullaniciBilgi(User &oth);  friend void showKrediKartiBilgi(User &oth);  friend void showDovizBilgi(User &oth);  friend void operator+( User& oth);  private:  int musNo;  string musName;  string musSname;  int hesapNo;  int password;  double bakiye;  double dolar\_bakiye;  double euro\_bakiye;  double toplamFatura;    public:  int pwordhak;  User();  User(int musNo, string musName, string musSname, int hesapNo, int password, double bakiye, int hak, double dolar, double euro);  ~User();  User(const User& oth);  User& operator=(const User& oth);      void setmusNo(int yeniNo);  void setmusName(string yeniName);  void setmusSname(string yeniSname);  void setHno(int yeniNo);  void setPw(int yenipw);  void setToplamFatura(double );    void setBakiye(double yenib);  void ekleBakiye(int ekleb);  void cekBakiye(int cekb);  void setdBakiye(double yenid);  void seteBakiye(double yenie);      int getmusNo();  string getmusName();  string getmusSname();  int getHno();  int getPw();  double getBakiye();  int ispw(int sifre);  double getDolarb();  double getEurob();  double getToplamFatura();    Kart k;  Elektrik e;  Su s;  Dogalgaz d;  Internet i;  doviz dov;  };  #endif  DOVIZ.CPP  #include "doviz.h"  doviz::doviz(){  dolar = 9.0;  euro = 10.0;  }  doviz::doviz(double \_dolar, double \_euro)  {  dolar = \_dolar;  euro = \_euro;  }  doviz::doviz(const doviz& y){  dolar = y.dolar;  euro = y.euro;  }  doviz& doviz::operator = (const doviz& y)  {  dolar = y.dolar;  euro = y.euro;  return \*this;  }  doviz& doviz::operator + (doviz& y)  {  dolar += y.dolar;;  euro += y.euro;  \*this;  }  void doviz::set()  {  cout << "dolar fiyati giriniz" << endl;  cin >> dolar;  cout << "euro fiyati giriniz" << endl;  cin >> euro;  }  double doviz::getdolar(){ return dolar ; }  double doviz::geteuro(){ return euro ; }  DOVIZ.H  #include <iostream>  using namespace std;  class doviz{  private:  double dolar;  double euro;  public:  doviz();  doviz(double \_dolar, double \_euro);  doviz(const doviz& y);  doviz& operator =(const doviz& y);  doviz& operator +(doviz& y);    void set();  doviz& get();  double getdolar();  double geteuro();  };      MAIN.CPP  #include "User.h"  #include <stdlib.h>  #include <iostream>  using namespace std;  int bas\_bul(int pword);  int main()  {  int size=5;  User\* user = new User[size];    user[0].setmusName("Lale");  user[1].setmusName("Buse");  user[2].setmusName("Ali");  user[3].setmusName("Mehmet");  user[4].setmusName("Mila");    user[0].setmusSname("Gundogan");  user[1].setmusSname("Dogan");  user[2].setmusSname("Albayrak");  user[3].setmusSname("Koyun");  user[4].setmusSname("Demir");    user[0].setPw(12345);  user[1].setPw(45678);  user[2].setPw(41111);  user[3].setPw(44444);  user[4].setPw(67891);    user[0].setmusNo(1);  user[1].setmusNo(2);  user[2].setmusNo(3);  user[3].setmusNo(4);  user[4].setmusNo(5);    user[0].setHno(264571);  user[1].setHno(264572);  user[2].setHno(264573);  user[3].setHno(264574);  user[4].setHno(264575);    user[0].d.setAboneno(78364291);  user[1].d.setAboneno(78364292);  user[2].d.setAboneno(78364293);  user[3].d.setAboneno(78364294);  user[4].d.setAboneno(78364295);    user[0].e.setAboneno(67253181);  user[1].e.setAboneno(67253182);  user[2].e.setAboneno(67253183);  user[3].e.setAboneno(67253184);  user[4].e.setAboneno(67253185);    user[0].i.setAboneno(56142071);  user[1].i.setAboneno(56142072);  user[2].i.setAboneno(56142072);  user[3].i.setAboneno(56142074);  user[4].i.setAboneno(56142075);    user[0].s.setAboneno(45031961);  user[1].s.setAboneno(45031962);  user[2].s.setAboneno(45031963);  user[3].s.setAboneno(45031964);  user[4].s.setAboneno(45031965);  user[0].pwordhak = 3;  user[1].pwordhak = 3;  user[2].pwordhak = 3;  user[3].pwordhak = 3;  user[4].pwordhak = 3;  for(int i=0; i<size; i++){  user[i].setBakiye(rand()%100000);  }  for(int i=0; i<size;i++){  user[i].d.setDsayac(rand()%100);  user[i].e.setEsayac(rand()%100);  user[i].i.setisayac(rand()%100);  user[i].s.setSusayac(rand()%100);  }  for(int i=0; i<size;i++){  user[i].d.setoTutar(user[i].d.getBirimf()\*user[i].d.getdsayac());  user[i].e.setOtutar(user[i].e.getBirimf()\*user[i].e.getEsayac());  user[i].i.setoTutar(user[i].i.getBirimf()\*user[i].i.getisayac());  user[i].s.setoTutar(user[i].s.getBirimf()\*user[i].s.getSusayac());  }  for (int i=0; i<size ; i++){  user[i].setdBakiye(rand()%100);  user[i].seteBakiye(rand()%100);  }    for(int i=0; i<3;i++)  {  user[i].k.setBorc(rand()%100);  user[i].k.setLimit(rand()%1000);  }  int no,sifre,sec,sec2;  int sec3=0;  int sec4=0;  int miktar2=0;  int choice,flag;  int sifre\_status=99, pwordhak = 3;  cout << "\n\n\t\t\t\t\t\*\*\*\*\*GEBZE BANKASI\*\*\*\*\*"<<endl;  cout << "\n\t\t\t\tGebze Bankasi Internet Bankaciligina Hosgeldiniz!" << endl;    while(flag=1){  cout << "\n\n Lutfen musteri numaranizi giriniz" << endl;  cin >>no;  if(no>0 && no <6){  break;  }  else{  cout << "Gecersiz musteri no girdiniz. Tekrar deneyiniz." << endl;  flag=-1;  }  }  gosifre:  for(int i=0; i<size; i++)  {      if(user[i].getmusNo()==no && &user[i].pwordhak)  {  try  {  cout << " Lutfen sifrenizi giriniz." << endl;  cin >>sifre;  sifre\_status = bas\_bul(sifre);    if ( sifre\_status == 0)  throw 0;  while(user[i].ispw(sifre) == 0 && user[i].pwordhak > 0 )  {  cout<<"\tYanlis sifre girdiniz, lutfen yeniden deneyin. \n" << " Kalan deneme hakkiniz " << user[i].pwordhak << endl;  cin>>sifre;//anahtar dogru girilene kadar kullanicinin anahtari tekrar girmesi istenir  sifre\_status = bas\_bul(sifre);  if ( sifre\_status == 0)  throw 0;  }  if( user[i].pwordhak <= 0)  {  cout << "\tSifre hakkiniz bittigi icin hesabiniz kilitlenmistir. Lutfen musteri hizmetleri ile iletisime geciniz." << endl;  goto hakbitti;  }  if(user[i].ispw(sifre) && user[i].pwordhak>0 )  {  system("cls");  cout <<"\t\t\nSaglikli Gunler, " <<user[i].getmusName()<<" "<<user[i].getmusSname()<<"\n\n\t\t------------HOSGELDINIZ------------\n"<< endl;      cout << "\n\t\tLutfen yapmak istediginiz islemi seciniz." << endl;  flag = 1;  while(flag == 1)  {  cout<<endl<<" \t1.Banka hesabi bilgileri goruntule"<< endl<< "\t2.Kredi karti bilgileri goruntule"<<endl  << "\t3.Fatura ode "<<endl<< "\t4.Aylik ozet goruntule "<<endl<<"\t5.Para Transferi"<<endl<<"\t6.Doviz Islemleri"<<endl<<"\t7. Cikis"<<endl;  cin >> choice;  switch (choice)  {  case 1 ://banka hesabi  system("cls");  showKullaniciBilgi(user[i]);    break;  case 2 ://kredi karti  system("cls");  showKrediKartiBilgi(user[i]);    break;  case 3 ://fatura ode  system("cls");  cout<<user[i].e;  cout<<user[i].s;  cout<<user[i].s;  cout<<user[i].i;  cout << "\tLutfen hangi faturayi odemek istediginizi seciniz." << endl << "\t1-Elektrik\n\t2-Su\n\t3-Dogalgaz\n\t4-Internet" << endl;  cin>>no;  switch (no)  {  case 1 ://elektrik  system("cls");  cout << "\n\t\t---Elektrik Faturasi---" << endl;  cout << "\tOdemeniz gereken tutar-> " << user[i].e.getOtutar() << endl;  cout << "\tHangi odeme yontemini secersiniz?" << endl << "\t1-Kredi Karti\n\t2-Hesap Karti" << endl;  cin >>sec;  if(sec==1)  {  cout << "\t---Kredi Karti" << endl;  if(user[i].e.getOtutar() > user[i].k.kullanilabilirTutar())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].k.getLimit()-user[i].e.getOtutar();  user[i].k.setLimit(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "Kredi kartinizda kalan limit tutari: " << kalan << endl;  user[i].e.setOtutar(0);  user[i].e.setEsayac(0);  }  }  else if(sec==2)  {  cout << "\t---Hesap karti" << endl;  if(user[i].e.getOtutar() > user[i].getBakiye())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].getBakiye()-user[i].e.getOtutar();  user[i].setBakiye(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "Hesabinizda kalan bakiye: " << kalan << endl;  user[i].e.setOtutar(0);  user[i].e.setEsayac(0);  }  }    break;  case 2 ://su  system("cls");  cout << "\n\t\t---Su Faturasi---" << endl;  cout << "Odenemeniz gereken tutar " << user[i].s.getOtutar() << endl;  cout << "\tHangi odeme yontemini secersiniz?" << endl << "\t1-Kredi Karti\n\t2-Hesap Karti" << endl;  cin >>sec;  if(sec==1)  {  cout << "\t---Kredi karti" << endl;  if(user[i].s.getOtutar() > user[i].k.kullanilabilirTutar())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].k.getLimit()-user[i].s.getOtutar();  user[i].k.setLimit(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "kredi kartinizda kalan limit tutari: " << kalan << endl;  user[i].s.setoTutar(0);  user[i].s.setSusayac(0);    }  }  else if(sec==2)  {  cout << "\t---Hesap karti" << endl;  if(user[i].s.getOtutar() > user[i].getBakiye())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].getBakiye()-user[i].s.getOtutar();  user[i].setBakiye(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "Hesabinizda kalan bakiye: " << kalan << endl;  user[i].s.setoTutar(0);  user[i].s.setSusayac(0);  }  }    break;  case 3 ://dogalgaz  system("cls");  cout << "\n\t\t---Dogalgaz Faturasi---" << endl;  cout << "Odenemeniz gereken tutar " << user[i].d.getOtutar() << endl;  cout << "\tHangi odeme yontemini secersiniz?" << endl << "\t1-Kredi Karti\n\t2-Hesap Karti" << endl ;  cin >>sec;  if(sec==1)  {  cout << "\t---Kredi karti" << endl;  if(user[i].d.getOtutar() > user[i].k.kullanilabilirTutar())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].k.getLimit()-user[i].d.getOtutar();  user[i].k.setLimit(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "kredi kartinizda kalan limit tutari: " << kalan << endl;  user[i].d.setoTutar(0);  user[i].d.setDsayac(0);  }  }  else if(sec==2)  {  cout << "\t---Hesap karti" << endl;  if(user[i].d.getOtutar() > user[i].getBakiye())  {  cout << "fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].getBakiye()-user[i].d.getOtutar();  user[i].setBakiye(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "Hesabinizda kalan bakiye: " << kalan << endl;  user[i].d.setoTutar(0);  user[i].d.setDsayac(0);  }  }    break;  case 4 ://internet  system("cls");  cout << "\n\t\t---Internet Faturasi---" << endl;  cout << "Odenemeniz gereken tutar " << user[i].i.getOtutar() << endl;  cout << "\tHangi odeme yontemini secersiniz?" << endl << "\t1-Kredi Karti\n\t2-Hesap Karti" << endl;  cin >>sec;  if(sec==1)  {  cout << "\t---Kredi karti" << endl;  if(user[i].i.getOtutar() > user[i].k.kullanilabilirTutar())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].k.getLimit()-user[i].i.getOtutar();  user[i].k.setLimit(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "kredi kartinizda kalan limit tutari: " << kalan << endl;  user[i].i.setoTutar(0);  user[i].i.setisayac(0);  }  }  else if(sec==2)  {  cout << "\t---Hesap karti" << endl;  if(user[i].i.getOtutar() > user[i].getBakiye())  {  cout << "Fatura tutariniz kullanilabilir tutarinizdan fazladir." << endl;  }  else  {  int kalan;  kalan =user[i].getBakiye()-user[i].i.getOtutar();  user[i].setBakiye(kalan);  cout << "Fatura odemeniz basariyla gerceklesti." << endl << "Hesabinizda kalan bakiye: " << kalan << endl;  user[i].i.setoTutar(0);  user[i].i.setisayac(0);  }  }    break;  default :  cout<<"Yanlis yada eksik tuslama yaptiniz."<< endl;  flag = 1;  }  break;  case 4 ://aylik ozet  system("cls");  cout<<user[i];  operator+(user[i]);    break;  case 5 :  system("cls");  int x,sec,tutar;  go\_no:  try{  cout<<"\n\tPara transferi yapmak istediginiz alicinin numarasini giriniz."<<endl;  cin>>x;  if(x<1 || x>5) throw 1;  }  catch(int deger){  cout<<"Bu musteri numarasina ait alici bulunamadi."<<endl;  goto go\_no;    }  cout<<"Alici :"<<user[x].getmusName()<<" "<<user[x].getmusSname()<<endl;  cout<<"Lutfen gondermek istediginiz tutari giriniz."<<endl;  cin>>tutar;  cout<<"hesap kartinizdaki kullanilabilir tutar: "<<user[i].getBakiye()<<endl;  cout<<user[i].k<<endl;//ostream  cout<<"\nOdeme araci secin"<<endl<<"1.Hesap karti\n2.Kredi karti"<<endl;  cin>>sec;  if(sec==1){  if(tutar > user[i].getBakiye()){  cout << "Gondermek istediginiz tutar kullanilabilir tutarinizdan fazladir. Isleminiz gerceklestirilemiyor." << endl;  }  else{  int kalan;  kalan =user[i].getBakiye()-tutar;  user[i].setBakiye(kalan);  cout << "Para transferiniz basariyla gerceklesti." << endl << "Hesabinizda kalan bakiye: " << kalan<<endl;  }  }  else if(sec==2){  if(tutar > user[i].k.kullanilabilirTutar()){  cout << "Gondermek istediginiz tutar kullanilabilir tutarinizdan fazladir. Isleminiz gerceklestirilemiyor." << endl;  }  else{  int kalan;  kalan =user[i].k.getLimit()-tutar;  user[i].k.setLimit(kalan);  cout << "Para transferiniz basariyla gerceklesti." << endl << "Kredi kartinizda kalan limit tutari: " << kalan<<endl;  }  }  break;  case 6:  system("cls");    cout<<"\n\n\t\tDoviz Islemleri"<<endl;  cout<<"\nLutfen yapmak istediginiz islemi giriniz."<<endl;cout<<"1.Doviz Hesabi Bilgi\n2.Doviz AL/SAT\n"<<endl;  cout<<"->";cin>>sec2;cout<<endl;  switch(sec2){  case 1:  showDovizBilgi(user[i]);  break;  case 2:  cout<<"1.Euro Islemleri\n2.Dolar Islemleri\n"<<endl;  cin>>sec3;  if(sec3==1){  cout<<"Guncel Euro Kuru ->"<<user[i].dov.geteuro()<<endl;  cout<<"1.AL\n2.SAT\n"<<endl;  cin>>sec4;  if(sec4==1){    int flagD=0;  while(flagD==0){  cout<<"Bakiyeniz-> "<<user[i].getBakiye();cout<<endl;  cout<<"Ne kadar Almak Istiyorsunuz-> ";cin>>miktar2;cout<<endl;  if(miktar2\*user[i].dov.geteuro() <= user[i].getBakiye()){  user[i].seteBakiye(user[i].getEurob()+miktar2);  user[i].setBakiye(user[i].getBakiye()-miktar2\*user[i].dov.geteuro());  flagD++;  cout<<"Islem Sonrasi Yeni Bakiyeniz-> "<<user[i].getBakiye();cout<<endl;  }  else{  flagD++;  cout<<"Yetersiz Bakiye veya Yanlis Girdi Lutfen Yeniden denemek icin 0 e cikmak icin 99 ye basin!!-> ";cin>>flagD;  cout<<endl;    }    }  }  else if(sec4==2){  int flagD=0;  while(flagD==0){  cout<<"Bakiyeniz-> "<<user[i].getBakiye();cout<<endl;  cout<<"Euro Bakiyeniz-> "<<user[i].getEurob();cout<<endl;  cout<<"Ne kadar Satmak Istiyorsunuz-> ";cin>>miktar2;cout<<endl;  if(miktar2 <= user[i].getEurob()){  user[i].seteBakiye(user[i].getEurob()-miktar2);  user[i].setBakiye(user[i].getBakiye()+miktar2\*user[i].dov.geteuro());  flagD++;  cout<<"Islem Sonrasi Yeni Bakiyeniz-> "<<user[i].getBakiye();cout<<endl;  cout<<"Islem Sonrasi Yeni Euro Bakiyeniz-> "<<user[i].getEurob();cout<<endl;  }  else{  flagD++;  cout<<"Yetersiz Bakiye veya Yanlis Girdi Lutfen Yeniden denemek icin 0 e cikmak icin 99 ye basin!!-> ";cin>>flagD;  cout<<endl;    }    }  }  }  else if(sec3==2){  cout<<"Guncel Dolar Kuru ->"<<user[i].dov.getdolar()<<endl;  cout<<"1.AL\n2.SAT\n"<<endl;  cin>>sec4;  if(sec4==1){    int flagD=0;  while(flagD==0){  system("cls");  cout<<"Bakiyeniz-> "<<user[i].getBakiye();cout<<endl;  cout<<"Ne kadar Almak Istiyorsunuz-> ";cin>>miktar2;cout<<endl;  if(miktar2\*user[i].dov.getdolar() <= user[i].getBakiye()){  user[i].setdBakiye(user[i].getDolarb()+miktar2);  user[i].setBakiye(user[i].getBakiye()-miktar2\*user[i].dov.getdolar());  flagD++;  cout<<"Islem Sonrasi Yeni Bakiyeniz-> "<<user[i].getBakiye();  }  else{  flagD++;  cout<<"Yetersiz Bakiye veya Yanlis Girdi Lutfen Yeniden denemek icin 0 e cikmak icin 99 ye basin!!-> ";cin>>flagD;  cout<<endl;    }    }  }  else if(sec4==2){  int flagD=0;  while(flagD==0){  system("cls");  cout<<"Bakiyeniz-> "<<user[i].getBakiye();cout<<endl;  cout<<"Dolar Bakiyeniz-> "<<user[i].getDolarb();cout<<endl;  cout<<"Ne kadar Satmak Istiyorsunuz-> ";cin>>miktar2;cout<<endl;  if(miktar2 <= user[i].getDolarb()){  user[i].setdBakiye(user[i].getDolarb()-miktar2);  user[i].setBakiye(user[i].getBakiye()+miktar2\*user[i].dov.getdolar());  flagD++;  cout<<"Islem Sonrasi Yeni Bakiyeniz-> "<<user[i].getBakiye();  cout<<"Islem Sonrasi Yeni Dolar Bakiyeniz-> "<<user[i].getDolarb();cout<<endl;  }  else{  flagD++;  cout<<"Yetersiz Bakiye veya Yanlis Girdi Lutfen Yeniden denemek icin 0 e cikmak icin 99 ye basin!!-> ";cin>>flagD;  cout<<endl;    }    }  }      }  break;  }  break;  case 7:  hakbitti:  cout<<"Cikis yapildi..";  flag=0;  break;  default :  cout<<"Gecersiz giris yaptiniz.!!!!!!"<< endl ;  flag=1;  }  }  }  }  catch( int hata )  {  cout << "Eksik tuslama yaptiniz.\n "<< endl;  goto gosifre;  }  }      }      delete [] user;  return 0 ;  }  int bas\_bul(int pword)  {  int temp1;  int basamak1=1;  temp1=pword;  while(temp1 > 10)  {  temp1 /=10;  basamak1++;  }  if( basamak1 != 5){  return 0;  }  return 99;  } |