Specifikáció

Cél:

A program egy amerikai mintájú bankrendszert valósít meg, amely lehetővé teszi ügyfelek regisztrálását különböző adatok(regisztráció dátuma, számlaszám, név, cím, születési dátum, életkor, állampolgárság, telefonszám, számla típusa – különböző kamatokkal, kezdőegyenleg) eltárolásával. Emellett lehetőség van adatmódosításra(cím, telefonszám), tranzakciókra(készpénzfelvétel, befizetés, ügyfelek közötti átutalás), regisztrált ügyfél adatainak lekérdezésére számlaszám vagy név alapján, regisztrált ügyfél törlésére, illetve regisztrált ügyfelek listájának megtekintésére.

Felület:

A program CLI-t, azaz parancssoros felületet használ, menüpontokat a felkínált numerikus(szám) billentyűkkel tudunk választani. A program nem kezel szóközt adatok bevitelénél, így a nevet, címet ’\_’ karakterrel választhatjuk el, dátumoknál a ’nn/hh/éééé’ formátumot alkalmazzuk.

Funkciók:

**Menü**: Itt választhatjuk ki, mit szeretnénk csinálni, a felsorolt lehetőségek közül, hozzárendelt szám alapján. A lehetőségek a következők:

**1 Új ügyfél regisztrálása**: Megadjuk a kért adatokat, azt a program fájlba kiírja és eltárolja.

Példa bemenet:

[Reg date:] 12/09/2023

[Account no:] 1

[Name:] Kiss\_Zsolt

[Address:] 45\_Feszek\_utca\_4400\_Nyiregyhaza

[Phone:] 06701985671

[Date of birth:] 02/03/2000

[Age:] 23

[Citizenship:] Magyar

[Account type: (felsoroljuk a lehetőségeket)] Savings

[Deposit:] 1000

Példa kimenet:

Siker: „Account created!” + adatok eltárolása

Ha van már ezen a számon ügyfél: „Account no already in use!”

Egyéb hiba esetén: „Invalid input!”

**2 Adatmódosítás**: Az ügyfelek címe és tel. száma változhat, így ezeket lehet módosítani.

Példa bemenet:

[Enter acc no:] 1

[What would you like to change? 1: address 2: phone] 2

[Enter new phone number] 0620444871

Példa kimenet:

Siker: „Changes saved!”

Nincs megadott számlaszámon ügyfél: „Record not found!”

Egyéb hiba: „Invalid!”.

**3 Tranzakciók**: Befizethetünk saját számlára, felvehetünk pénzt, utalhatunk létező ügyfelek számlái között.

Példa bemenet:

[What would you like to do? 1: withdraw 2: deposit 3: transfer] 1

[Enter the amount you want to withdraw:] 200

Példa kimenet:

Siker: „Successful withdrawal!” + kivett összeget levonja az egyenlegből

Nincs elég pénz a számlán: „Insufficient funds!”

Példa bemenet:

[What would you like to do? 1: withdraw 2: deposit 3: transfer] 2

[Enter the amount you want to deposit:] 200

Példa kimenet:

Siker: „Successful deposit!” + befizetett összeget hozzáadja az egyenleghez

Példa bemenet:

[What would you like to do? 1: withdraw 2: deposit 3: transfer] 3

[Enter recipient acc no:] 2

[Enter the amount you want to transfer:] 200

Példa kimenet:

Siker: „$200 successfully transferred!” + erről a számláról levonja, a másikhoz hozzáadja

Nincs elég pénz a számlán: „Insufficient funds!”

Nem létezik a számla amire utalunk: „Account does not exist!”

Mindhárom esetben egyéb hiba: „Invalid!”

**4 Adatok lekérdezése**: Megadjuk a választott azonosítási módot, és megkapjuk az információt.

Példa bemenet:

[Do you want to search by acc no: 1 or name: 2 ?] 1

[Enter acc no:] 1

Példa kimenet:

Siker: Kiírja az 1-es számlaszámú ügyfél összes adatát

Ha nincs ilyen ügyfél: „Record not found!”

Egyéb hiba: „Invalid!”

**5 Ügyfél törlése**: Töröljük az ügyfelet a rendszerből.

Példa bemenet:

[Enter the account you want to delete:] 1

Példa kimenet:

Siker: „Account deleted!”

Nem létező ügyfél: „Record not found!”

**6 Ügyfelek megtekintése**: Megkapjuk a regisztrált ügyfelek listáját pár adattal kiegészítve.

**7 Kilépés:** Kilép a programból.

A megadott C program egy egyszerű bankkezelési rendszer, alapvető funkciókkal, mint új számla létrehozása, meglévő számla információinak frissítése, tranzakciók végrehajtása (befizetés, kivét, átutalás), számla részleteinek ellenőrzése, meglévő számla törlése, ügyfelek listájának megtekintése és a rendszerből való kilépés. A program egy fájlt (record.dat) használ ügyféladatainak tárolására.

Itt van egy bontás a program kulcsfontosságú elemeiről:

**Adatszerkezetek:**

Struktúrák:

struct date: Egy dátumot reprezentál a month (hónap), day (nap) és year (év) mezőkkel.

struct account: Az ügyfélinformációkat reprezentálja olyan mezőkkel, mint a name (név), acc\_no (számlaszám), age (kor), address (cím), citizenship (állampolgárság), phone (telefonszám), acc\_type (számlatípus), amt (összeg), dob (születési dátum), deposit (befizetés dátuma) és withdraw (kivétel dátuma).

**Függvények:**

float interest(float t, float amount, int rate): Kiszámolja a kamatot az idő (t), a tőkeösszeg (amount) és a kamatláb (rate) alapján.

void fordelay(int j): Késlelteti a program végrehajtását meghatározott idővel.

void menu(): Megjeleníti a főmenüt, és kezeli a felhasználói bemeneteket a különböző lehetőségek közötti navigáláshoz.

void close(): Megjeleníti a bezárási üzenetet és befejezi a programot.

void new\_acc(): Új számla létrehozása, felhasználói adatok bekérésével.

void view\_list(): Megjeleníti az összes ügyfélszámla listáját.

void edit(): Ügyféladatainak szerkesztése számlaszám alapján.

void transact(): Befizetés, kivétel vagy átutalás végrehajtása a felhasználói bemenet alapján.

void erase(): Ügyfélszámla törlése a számlaszám alapján.

void see(): Ügyfélszámla részleteinek megjelenítése számlaszám vagy név alapján.

**Fájlkezelés:**

A program fájlműveleteket (fopen, fclose, fprintf, fscanf, remove, rename) használ az ügyféladatok olvasására és írására szolgáló record.dat fájllal.

Megjegyzés:

A program egy meghatározott jelszót feltételez a bankrendszerhez való hozzáféréshez, ami: "finomnyul".

A fájlkezelési műveleteket egy record.dat nevű fájlon végzi, amelyben tárolja az ügyféladatait.

A változás jogát fenntartjuk, a specifikáció a program írása folyamán bővülhet!