Varjú Ákos
SSNTUX

Splitwise

- A program célja nyilvántartani eseményeket, azoknak a résztvevőit és részvevőinek a kiadásait, majd a „pénzügyeket” arányosan elosztani a részvevők között úgy, hogy mindenki kiveszi a részét az esemény költségeiből egyenlően.
- Lehet különböző eseményeket pl: síelés, szülinap, hétvégi buli stb… létrehozni. Ezekhez pedig hozzáadni a résztvevőket. A résztvevőknek lehetnek kiadásai, ezek összege az esemény költsége, melyet a résztvevők egyenlően elosztanak.
- A program fő funkciója a „Calculate”, mely kiírja részvevők és a kiadások alapján, hogy mennyi pénzt kinek, hogy kell elosztani, hogy minden résztvevő az adott eseményben egyenlően jöjjön ki.

Felhasználói kézikönyv:

A program megnyitásakor a grafikus felület főképernyője fogadja a felhasználót. Ebben az ablakban lehetőségünk van a következőkre:

1. Megtekinteni az Eseményeket. Ehhez az „events” fülre kattintva megjelennek az Események.
2. „Load” gombra kattintva az adatok betöltése.
3. „Save” gombra kattintva az adatok mentése.
4. „Create” gomb megnyomásával: Új Event ~ Esemény létrehozása. Meg kell adni egy nevét és egy dátumot. A név egyedi azonosító, két Eseménynek nem lehet ugyan az a neve. A program nem hozza létre, ha nem egyedi a név.
5. „Delete” gombra kattintva a ComboBoxban kiválasztott Eseményt törli.
6. „Open” gombra kattintva a ComboBoxban kiválasztott Eseményt megnyitja.

Varjú Ákos
SSNTUX

Ha a felhasználó kiválasztott egy Eseményt, majd megnyitja azt egy új ablak jelenik meg. Itt lehetőségünk van a következőkre:

- Résztvevők kezelése:

    1. Résztvevő hozzáadása: „Participant name” mezőbe meg kell adni egy nevet, melynek egyedinek kell lennie különben a hozzáadás sikertelen. Ezen felül nem lehet használni szóközt, ha mindenképpen teljes neveket szeretnénk használni, használjon „- vagy _” -t. Az „add” gombra kattintva létrehozza az új résztvevőt.
    2. Résztvevők megtekintése: „Participants” ComboBoxra kattintva megjelennek a résztvevők.
    3. Résztvevők törlése: A ComboBox-ban kiválasztott elemet törölni a „Delete” gomb megnyomásával érhetjük el. Természetesen ilyenkor az adott résztvevő kiadásai is törlődnek.

- Kiadások kezelése:
    1. A kiadásokat a „Expenses” ComboBox-ra kattintva tekintheti meg a felhasználó.
    2. Kiadást hozzáadni „Expense name” és „value”, illetve „Paid by” ComboBox-ot ki kell tölteni majd az „Add” gombra kattintva beviszi a felhasználó az új kiadást.
        a. Paid by: Kinek a kiadása
        b. Expense name: Kiadás neve
        c. Value: kiadás mértéke: Ne használjon negatív számokat
    3. Kiadás törlése: A ComboBoxban kiválasztott elemet törli.

- Calculate avagy a végeredmény meghatározása
    1. Megtekintheti a felhasználó az Esemény részleteit:
        a. Event neve
        b. Dátuma
        c. Résztevők száma
        d. Event költsége
        e. Átlag
    2. A végeredmény meghatározása: „Calculate” gomb megnyomásával ki értélelődnek a tranzakciók, melyek a „Transactions” ComboBoxban megjelenek.
    3. A végeredmény megtekintése: „Tranzactions” ComboBox-ra kattintva megjelennek a fizetendő összegek és hogy ki kinek, hogy tartozik.

A program sikeres működéséhez, lennie kell egy Data.txt nevű fájlnak, ha ez nem létezik hozzon létre egyet. Majd futtassa újra a programot.

Varjú Ákos
SSNTUX

Osztályok:

- **Sys:**
- Felelőssége:
  1. tárolni a „Event” -eket
  2. lebonyolítania fájlból olvasását és fájlba írást. A fájl, amiből adatokat olvas/ ír egy Data.txt nevű szöveges dokumentum.
- Attribútum:
  1. ArrayList<Events> events: ebbe tarolja az Eventeket
- Metódusok:
  1. addEvent:
     Paraméterként kap egy „name” és egy „date” Stringet. A Eventek neve kulcs, mely alapján egyértelműen azonosítható. Metódus igazzal tér vissza, ha még nincs ilyen nevű Event és létrehozza azt, hamissal, ha már létezik ilyen nevű Event.

  2. delEvent:
     A paraméterként kapott Eventet törli.

  3. SaveSys:
     Fájl kiírása Data.txt- be.

  4. loadSys:
     Fájl beolvasása a Data.txt-ből.

  5. findEvent:
     A paraméterként kapott String alapján keresi majd visszaadja a keresett Eventet.

Varjú Ákos
SSNTUX

- **Event:**
- Felelősségek:
  1. Tárolni az adott Event résztvevőit
  2. Számon tartani az esemény költségét
  3. Szamon tartani az átlagot
- Attribútumok:
  1. String name: Event neve
  2. String date: Event dátuma
  3. ArrayList<> participants: Event résztvevői
  4. ArrayList<> transactions: Event tranzakciói
  5. double buget: Event kerete
  6. double average: átlag
- Metódusok:
  1. Event: konstruktor

  2. addParticipant:
     Paraméterként kapott String „name” alapján hozzad egy résztvevőt. A résztvevő neve egyedi ezt vizsgálva, ha továbbra is egyedi igazzal, ha van mar ilyen nevű felhasználó hamissal tér vissza.

  3. delParticipant:
     Paraméterként kapott név alapján törli az adott Participant-t.

  4. update:
     bugdet és average-t frissíti

  5. szamol: később bővebben lásd algoritmus

  6. Pnames:
     Visszaad egy String tömböt a résztvevők nevével.

  7. Expnames:
     Visszaad egy String tömböt a kiadások nevével.

  8. findP:
     Visszaadja a paraméterként kapott String „name” nevű résztvevőt.

Varjú Ákos
SSNTUX

- **Participant**:
- Felelősségek:
    1. Számon tartani a résztvevő kiadásait, általa fizetett összeget és a különbséget.
- Attribútumok:
    1. String name: Participant neve
    2. ArrayList<> expenses: Participant kiadásai/Expensei
    3. double kulonbseg: esemény kerete – átlag= kulonbseg ez alapján számol a szamol() metódus.
    4. Event insideof: Az Event aminek a résztevője az adott Participant
    5. double balance: résztvevő által fizetett összeg
- Metódusok:
    1. Participant: konstruktor

    2. addExpense: Expense name egyediséget vizsgálja, a paraméterként kapott Stringgel, ha egyedi a kapott Stringgel hozzáadj egy új Expenset aminek az értéke a paraméterben kapott value, ha nem egyedi, nem adja hozza.

    3. delExpense: a paraméterként kapott Expenset törli

    4. updateP: A résztvevők adatait frissíti

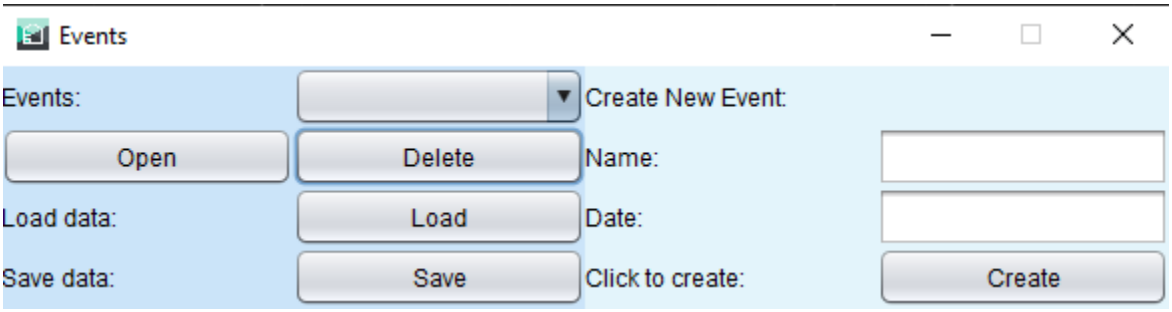
    5. findExp: Név az egyértelműen azonosítja a kiadást. Ez alapján keres és visszaadja a keresendő Expense-t.

- **Expense:**
- Felelőségek: Egy kiadást reprezentál, tárolja, hogy ki fizette, mi a neve és a kiadás értéket.
- Attribútumok:
    1. String name: Expense neve
    2. Participant payedby: Az a résztvevő, akihez a kiadás tartozik.
    3. double amount: Expense értéke
- Metódusok:
    1. Expense: konstuktor

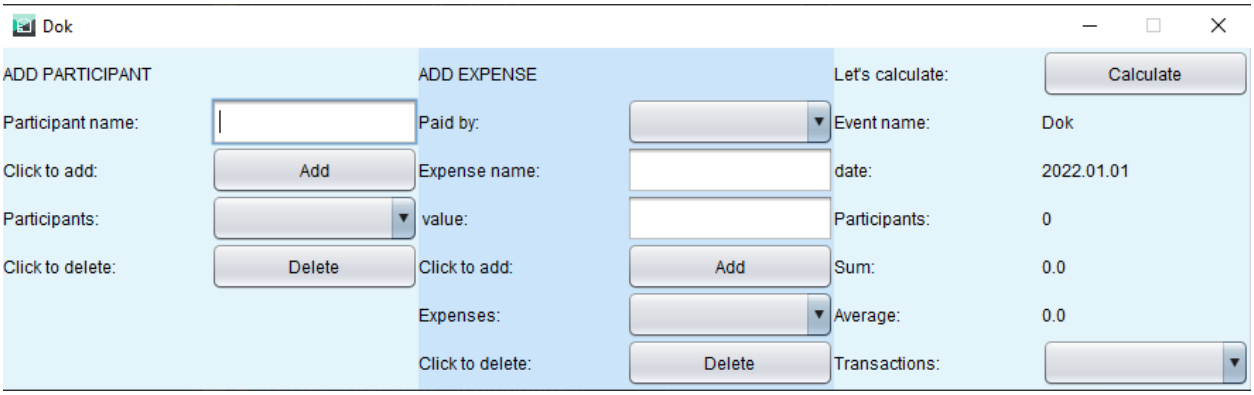
Varjú Ákos
SSNTUX

- **Transaction:**
- Felelőséhek: Az eredmény tranzakciók listája. Egy ilyen tranzakciót reprezentál
- Attribútumok:
  1. Participant paid_by: Az a Participant aki pénzt ad
  2. Participant paid_to: Az a Participant akinek pénzt adnak
  3. double amount: összeg
- Metódusok:
  1. Transaction: konstuktor

- **EventsFrame:**
- Felelőségek: Az Eventek kezelése. Új hozzáadása, törlés és Event megnyitás. Adatok mentése és törlése.

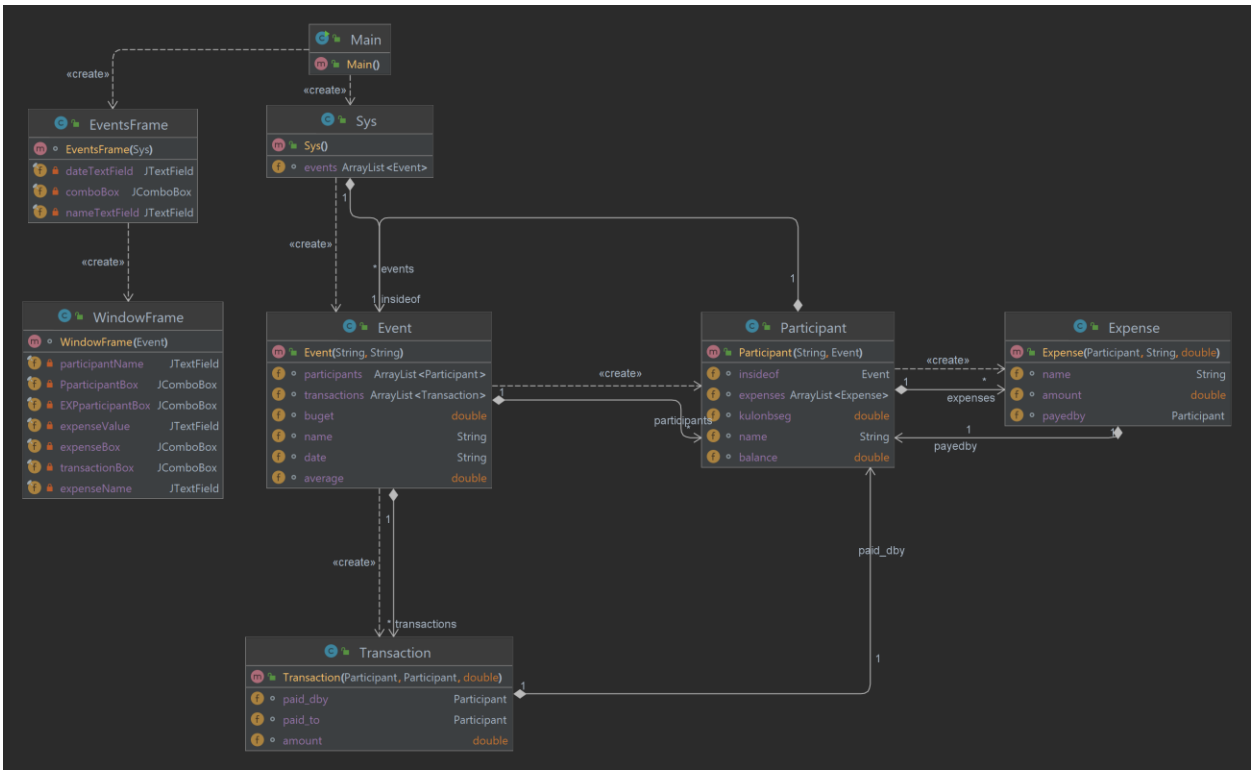


- **WindowFrame:**
- Felelősségek: Az adott Event kezelése. Részvevők hozzáadása, törlése. Kiadások hozzáadása, törlése. Végeredmény és adatok megjelenítése.



Varjú Ákos
SSNTUX

❖ Osztálydiagram:



Algoritmus:

1. Az eseménynek a keret attribútuma, ami nem más, mint a kiadások összege
2. Az átlag a keret osztva (részvevők száma). Ennyibe fog minden egyes résztvevőnek kerülni az esemény.
3. Minden résztvevőnek van egy saját egyenlege~” balance”: ami a kiadásainak az összegét jelenti.
4. Ezen felül van egy „**különbség**” = átlag-egyenleg attribútuma a résztvevőnek. Pl.: (mindenkinek 1000ft-ba kerül az esemény, de az adott résztvevő 2000ft-ot fizetett→különbség: -1000ft azaz tartoznak neki)
   A különbség 3 féle értéket vehet fel.
   a. Ha negatív: tartoznak neki
   b. Ha ez 0 annyi a résztvevő kiadása, mint az átlag. Az adott résztvevő befizette a részét, akkor kész.
   c. Ha pozitív: tartozik

Varjú Ákos
SSNTUX

Algoritmus:

Ciklus végig megy az összes részvevőn.

Belemegy az adott részvevőbe és vizsgálja azt.
Ha különbség=0 //ki van egyenlítve
nem csinálunk semmit, tovább a következő résztvevőre
Ha különbség>0 -- > //tartozik
nem csinálunk semmit, tovább a következő résztvevőre
Ha különbség<0 -- > //tartoznak neki,
Ciklus végig megy az összes többi résztvevőn
vizsgája azok különbségét, ha különbség>0 azaz tartozik akkora szűk keresztmetszettel*** létrehoz egy tranzakciót, és lefonja/hozzáadja az részvevő különbségéhez/egyenlegéhez így közeledik a 0 hoz. Aki tartozik az fizet annak, akinek tartoznak.

Ciklus vége

Ciklus vége

Keresztmetszet: a külső ciklus résztvevőjének különbsége és a belső ciklus résztvevőjének különbsége közül a kisebb.