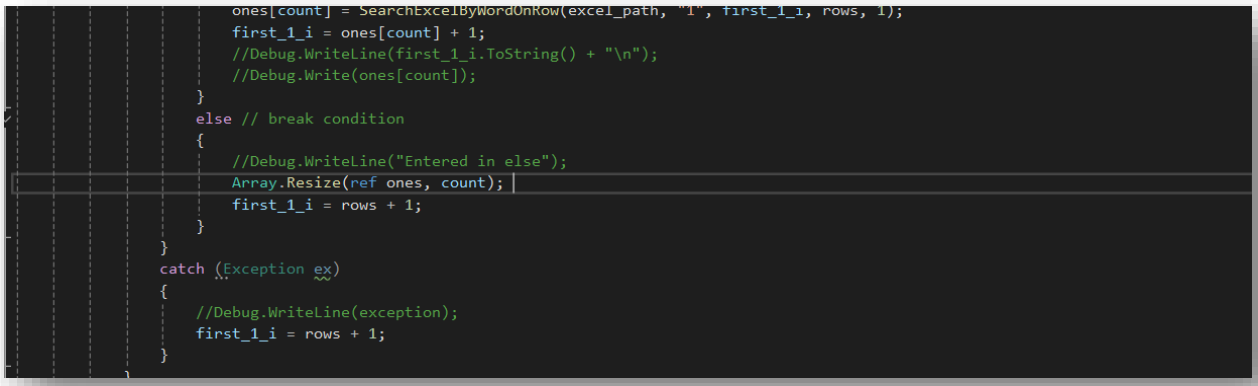
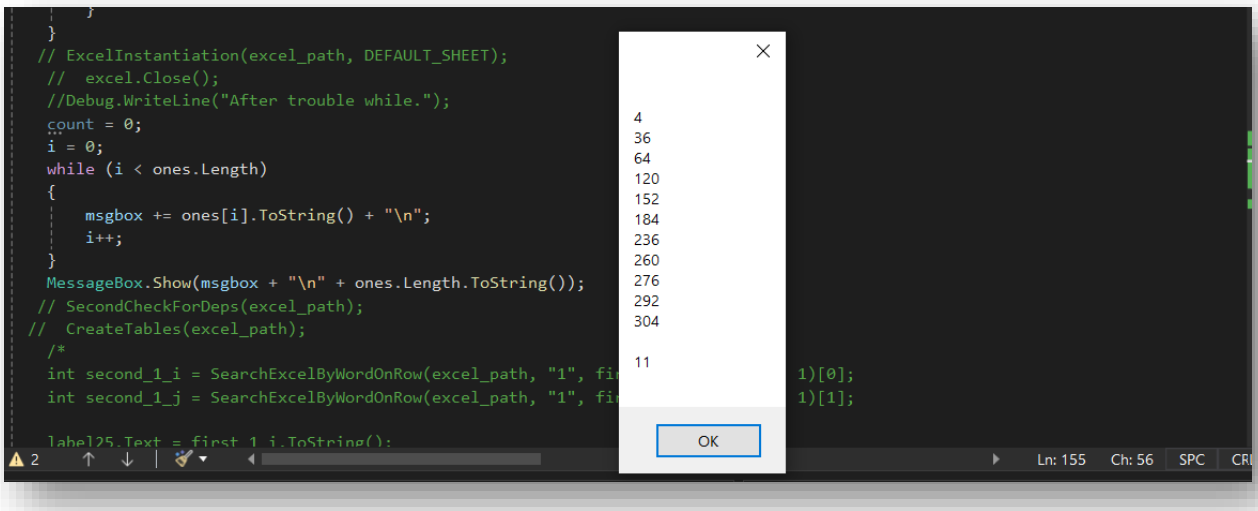
Rezumat 20.08.2024

- Găsirea unei metode optime de a rezolva cazurile în care nu se identifică și specializările care nu încep cu anul 1 exclude varianta brută în care fiecare departament este comparat cu celelalt pentru a determina care dintre ele nu a fost înregistrat în vectorul ones[], care conține locația "1-urilor" din sheet-ul orarului.
- Vectorul ones[] se populează în timpul încărcării formularului, aici am greșit făcând supoziția că toate departamentele încep cu anul 1, în cadrul IESC, TSTC începe cu anul 3, singurul departament căruia nu este asociat un 1 pe linie cu el. Acesta este singura excepție deoarece celelalte departamente pentru master/doctorat încep cu anul 1, iar nu cu un altul.
- Am scurtat vectorul ones[] care reținea și valori inutile nule. Următoarele două imagini clarifică acest aspect.



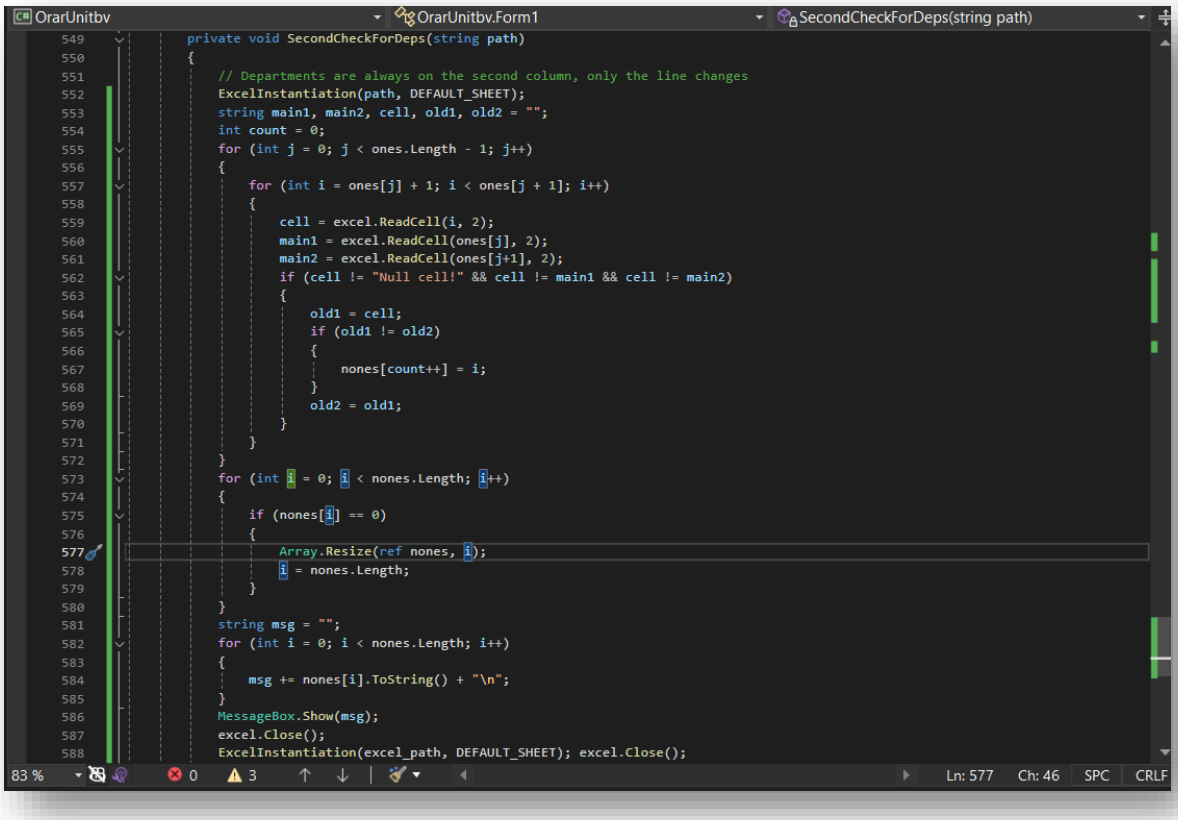


- Acum putem stabili cu exactitate numărul de departamente care încep cu anul 1, și anume 11, dar rămâne excepția TSTC, care începe cu anul 3. Deci, în mod real ar fi 12 departamente.

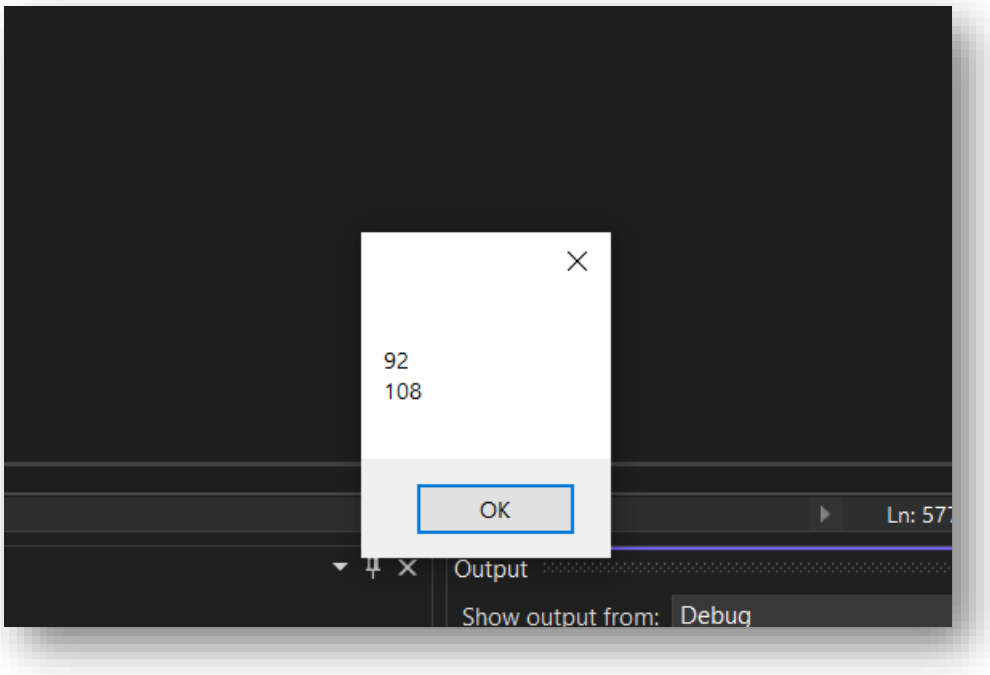
Acest lucru se poate rezolva dacă se știu a priori numărul de departamente, adică acestea nu sunt găsite în sheet-ul orarului, dar aceasta soluție necesită un input adițional, pe lângă calea unde se găsește sheet-ul cu care lucrăm, deci nu este o soluție optimă. Utilizatorul trebuie să selecteze calea unde se află sheet-ul fără alte complicații, iar softul trebuie să prelucreze datele corespunzător.

- Reducând dimensiunea alocată vectorului ones[], putem face compararea brută, deși nu este optimă niciodată, dintre fiecare celulă și celula adiacentă din ones[].

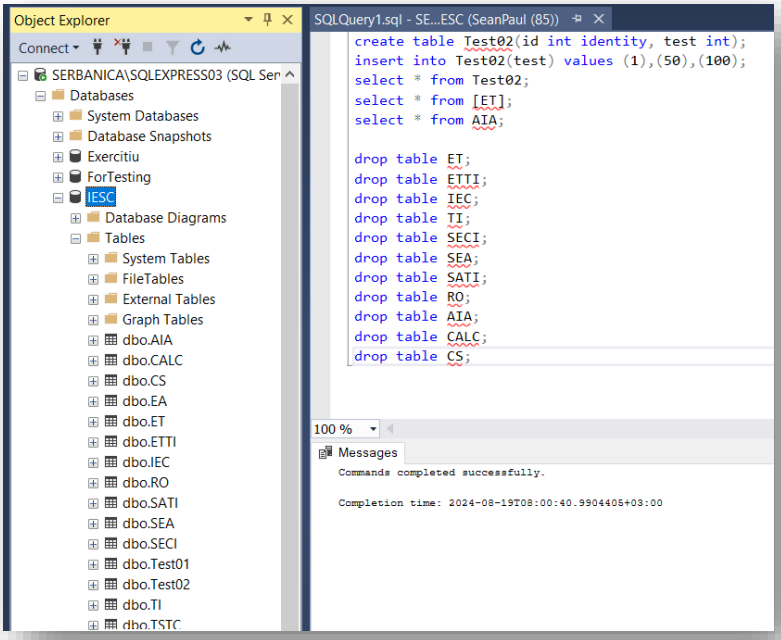
Am renunțat la metoda de comparare fiecarei celule cu fiecare, prin urmare metoda pe care am implementat-o este asemănătoare în sensul în care se compară multe celule, dar acum compararea se face între două intervale bine stabilite determinate de vectorul ones[]. Așadar compar toate celule cu limitele și cu celula nulă să vad dacă există specializare care nu începe cu anul 1. Pe lângă TSTC mai există și EA, deci o excepție în plus. Dacă între AIA și ET, se află, spre exemplu, TSTC, atunci aceasta neconcordanță va fi semnalată, iar coordonata de linie/rând a acestui departament va fi înregistrată în vectorul nones[], iar aceasta nu se va înregistra succesiv de mai multe ori corespunzător mai multor ani de studiu, ci va fi marcată doar o singura dată. Acest vector va fi trunchiat pentru a elimina valorile nule din instanțiere după cum urmează:



Rezumat 20.08.2024

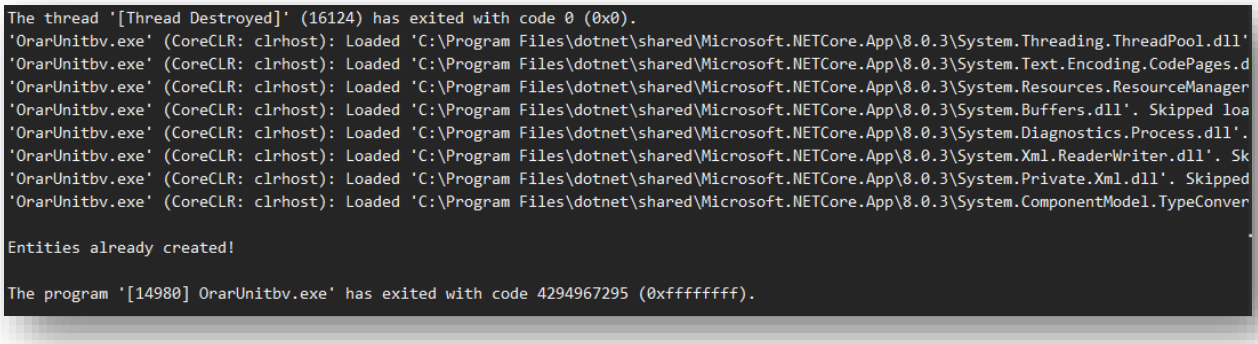


- În următoare imagine sunt prezentate entitățile create cu ajutorul ambilor vectori, ones[] și nones[]. Acesta este prima rulare, deci entitățile vor fi generate în încărcarea formularului.

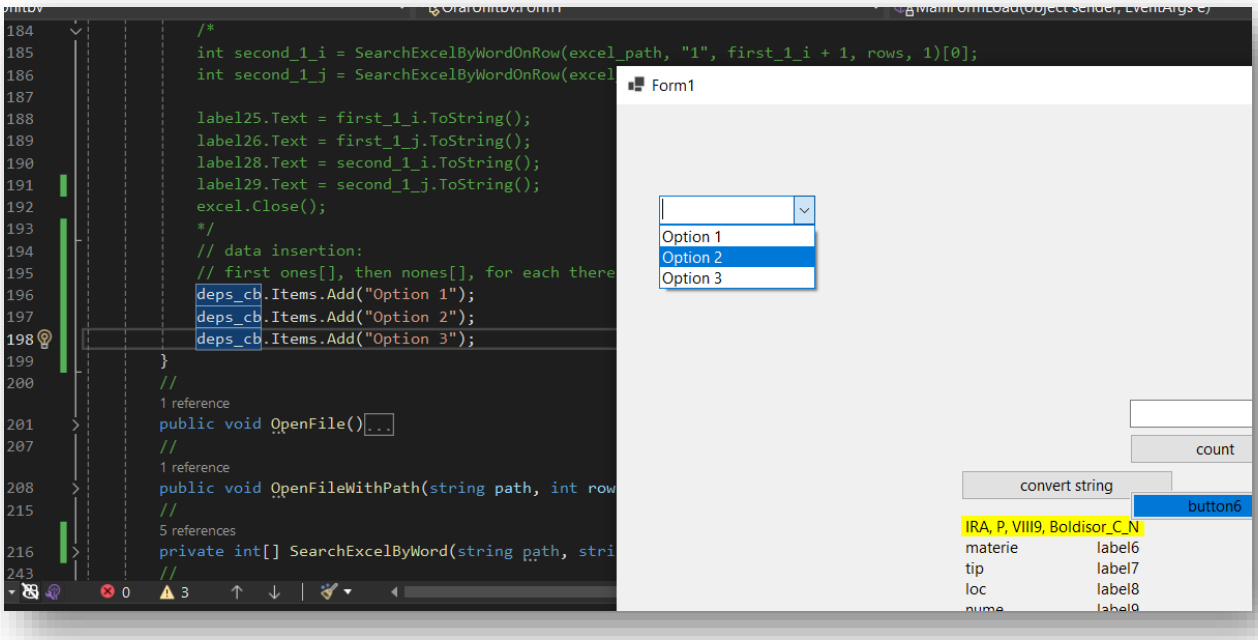


Rezumat 20.08.2024

- A doua rulare ne va transmite un mesaj care ne anunță că entitățile din imaginea de mai sus există deja, aceasta este, de fapt, un sistem de gestiune a excepțiilor SQL. Codul de excepție 208 este cel care întoarece mesajul că entitățile menționate deja există în baza de date, astfel se trece peste acest pas, iar timpul de start, de încărcare se micșorează.

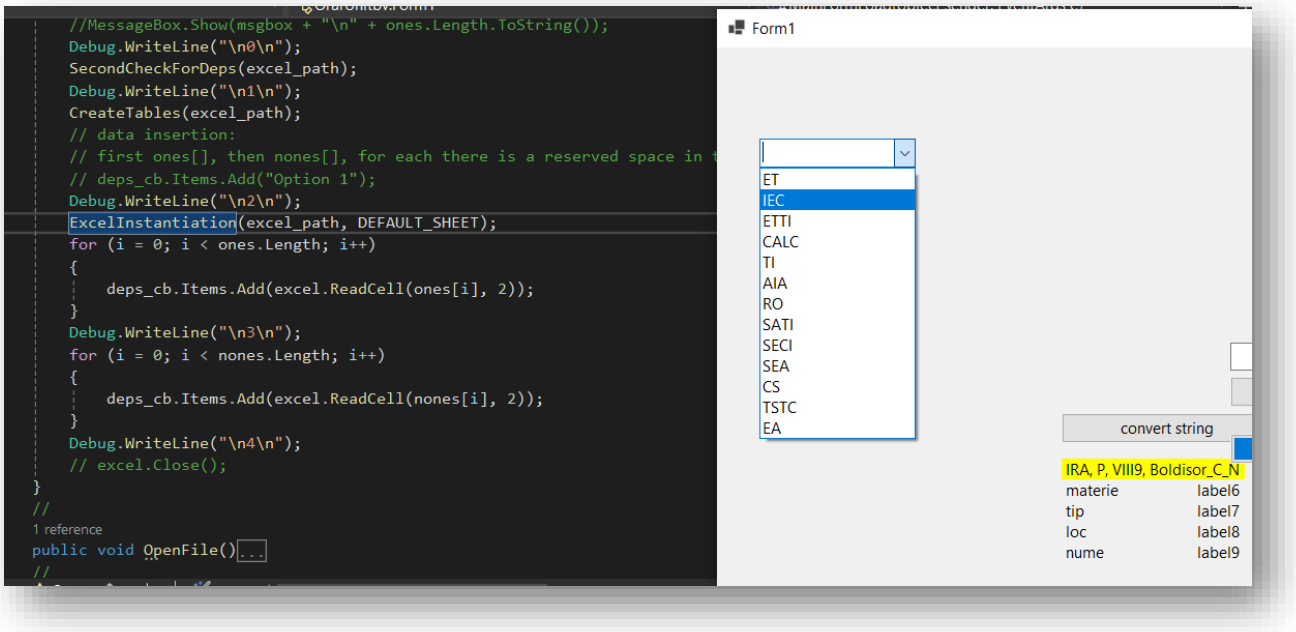


- Pentru a ușura popularea entităților voi face ca popularea tabelului să depindă de utilizator, astfel tabelele nu vor fi populate automat în momentul încărcării formularului sau în urma apăsării unui buton toate împreună. Aici se impune și verificarea prealabilă a datelor introduse pentru a se evita redunța, adică să nu inserăm aceleași date de două ori. Strucutura tabelului rămâne după cum am descris-o și înfățișat-o în rezumatul precedent. Astfel voi alege să lucrez cu un item grafic de tipul ComboBox, aspect ilustrat în imaginea următoare:



Rezumat 20.08.2024

- Aplicat pe sheet-ul orarului, lucrurile vor arăta în felul următor:

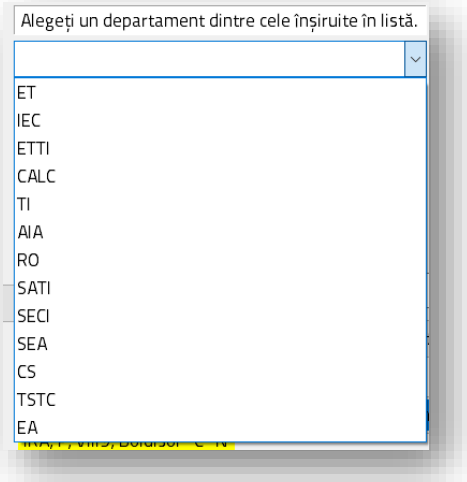


- Cu departamentul ales vom merge mai departe pentru a-i popula câmpurile din baza de date. Popularea simultană a tuturor tabelelor a tuturor departamentelor este ușor anevoioasă din pricina resurselor consumate, deja până în acest punct, sigur că până vom face degrevarea și defalcarea funționalităților pe baza a mai multor elemente grafice. În acest stadiu toate funcționalitățile sunt declanșate secvențial în încărcarea formularului.
- Funcția/metoda care se ocupă de introducerea departamentelor într-un vector este înfățișată în captura de mai jos. Am observat că este mai practic să lucrez cu un vector de string-uri interne decât să accesez resursa la fiecare citire. Accesarea celulelor orarului trebuie făcută cât mai puțin posibil, așadar, de pildă, abrevierile departamentelor vor fi reținute într-un vector local, iar coordonatele vor fi lăsate în vectorii ones[] și nones[]. Astfel pentru crearea entităților nu vor mai fi folosite două comenzi, ci doar una bazată pe vectorul deps[]. Până acum tabelele erau create pe baza două comenzi identice, diferite fiind doar coordonatele (indecșii de linie) ai vectorilor responsabili cu reținerea acestora. Procedând în acest mod voi evita instanțierea și procesul său invers, "deinstațierea" sau închiderea instanței. Menționez că nu este obligatoriu ca aceasta să fie închisă, dar așa se previne supraîncărcarea memoriei. La 100 de rulări, fără închiderea instanței, voi regăsi în Task Manager 100 de instanțe de Excel în background, fapt nepermis. Evitarea instanțierilor inutile degrevează situația în sensul în care nu mai trebuie ținut cont de procesul invers de "deinstanțiere." Tot ce este deschis, trebuie să fie și închis.

Rezumat 20.08.2024

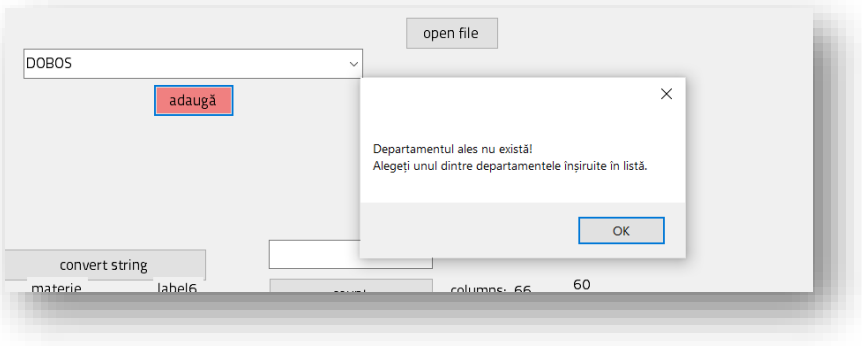
```
internal void DepsFill()
{
    ExcelInstantiation(EXCEL_PATH, DEFAULT_SHEET);
    int c = 0, i;
    for (i = 0; i < ones.Length; i++ )
    {
        deps[c++] = excel.ReadCell(ones[i], 2);
    }
    for (i = 0; i < nones.Length; i++)
    {
        deps[c++] = excel.ReadCell(nones[i], 2);
    }
    int length = deps.Length;
    for (i = 0; i < length; i++)
    {
        if (deps[i] == null)
        {
            Array.Resize(ref deps, i);
            i = length;
        }
    }
    //
    string msg = "";
    for (i = 0; i < deps.Length; i++)
    {
        msg += deps[i] + "\n";
    }
    MessageBox.Show(msg);
    //
    excel.Close();
}
//
1 reference
internal void OpenFile()
```

- Pentru a nu îngreuna popularea entităților, poate ulterior chiar crearea lor, acestea vor fi inserate pe baza utilizatorului acesta va umple deci tabelele cu date. Se poate face și totul deodată, dar nu este eficient ca timp. Numai generarea tuturor entităților și căutarea prin sheet-ul orarului este costisitoare. Imaginile de mai jos arată cum se poate folosi un ComboBox pentru a împiedica utilizatorul, deși el poate în continuare încerca să introducă alte departamente sau să greșească. Departamentele nu sunt hard-codate, sunt preluate din sheet-ul orarului.

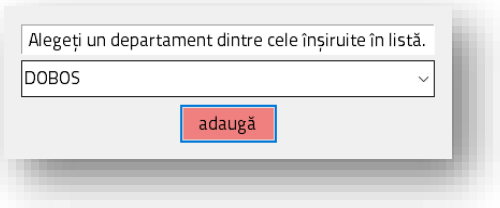


Rezumat 20.08.2024

- Mesajul de eroare:



- Mesajul de avertizare care apare după două secunde și jumătate care indică utilizatorului să aleagă o varinta din listă.



- În continuare, pentru a ușura introducerea datelor, voi crea o clasă Schedule.cs cu care voi instanția obiecte și voi introduce datele într-o manieră mai cursivă. Părțile de cod de mai jos ilustrează acest fapt.

```
1 reference
internal class Schedule
{
    // follows the setup of the entity to create
    private string tab, abvr, type, prof, day,
        time, place, stds;
    0 references
    public Schedule(string tabi, string abvri, string typei, string profi, string dayi
    {
        // constructor
        this.tab = tabi;
        this.abvr = abvri;
        this.type = typei;
        this.prof = profi;
        this.day = dayi;
        this.time = timei;
        this.place = placei;
        this.stds = stdsi;
    }
    0 references
    internal void setTab(string tabi) { this.tab = tabi; }
    0 references
    internal void SetAbvr(string abvri) { this.abvr = abvri; }
    0 references
    internal void SetType(string typei) { this.type = typei; }
    0 references
    internal void SetProf(string profi) { this.prof = profi; }
```

Rezumat 20.08.2024

```
        internal void setTime(string timei) { this.time = timei; }
        0 references
        internal void setPlace(string placei) { this.place = placei; }
        0 references
        internal void setStds(string stdsi) { this.stds = stdsi; }
        0 references
        internal String getTab() { return this.tab; }
        0 references
        internal String getAbvr() { return this.abvr; }
        0 references
        internal String getType() { return this.type; }
        0 references
        internal String getProf() { return this.prof; }
        0 references
        internal String getDay() { return this.day; }
        0 references
        internal String getTime() { return this.time;}
        0 references
        internal String getPlace() { return this.place; }
        0 references
        internal String getStds() { return this.stds; }
        0 references
        internal String To_String()
        {
            return "Schedule" + "(tab: " + this.tab + ",\n" +
                                "abvr: " + this.abvr + ",\n" +
                                "type: " + this.type + ",\n" +
                                "prof: " + this.prof + ",\n" +
                                "day: " + this.day + ",\n" +
                                "time: " + this.time + ",\n" +
                                "place: " + this.place + ",\n" +
                                "stds: " + this.stds + ")";
        }
    }
```

- Facem presupunerea că numerotarea pe coloane se face de la 4 la 38, adică 35 de celule, iar liniile sunt marcate de vectorii ones[] și nones[]. Dar pentru a putea extragele datele din sheet-ul orarului cei doi vectori trebuie combinați și sortați pentru că se poate întâlni situația rară ca între două ones-uri să găsim o coordonată a unui nones și astfel tabelul ar fi populat eronat cu datele corespunzătoare altor departamente.
- Rămâne să fac următoarele zile algoritmul de prelucrare a celulelor.