

Colegiul Național „Mihai Eminescu”, Oradea.

Profilul matematică-informatică intensiv informatică

LUCRARE DE ATESTAT

-Examen auto-

Profesor coordonator: Cristina Bolea

Candidat: Marinescu Raul, Clasa a XII-a F

Promoția 2021

Cuprins

Cap I Motivarea alegerii lucrării

Cap II Descrierea lucrării

1. Reguli
2. Interfațele
3. Descrierea elementelor de pe interfață

Cap III Ilustrarea codului

Cap I Motivarea alegerii lucrării

În clasa a XII-a, mai ales la această vârstă, tinerii încep să fie conștienți de responsabilitățile și îndatoririle pe care trebuie să le respecte. Cei mai mulți devin majori și primul lor gând este acela de a începe studiile și orele de condus în vederea obținerii carnetului de sofat. Și eu am trecut prin această perioadă așa că, având cunoștințele și experiența necesară proaspăt asimilată, m-am decis ca lucrarea mea de atestat să fie o simulare pentru adolescenții care vor susține un examen auto pentru a-i ajuta în pregătire lor. Este o bună modalitate pentru un viitor sofer de a evalua cunoștințele acumulate până în momentul de față.

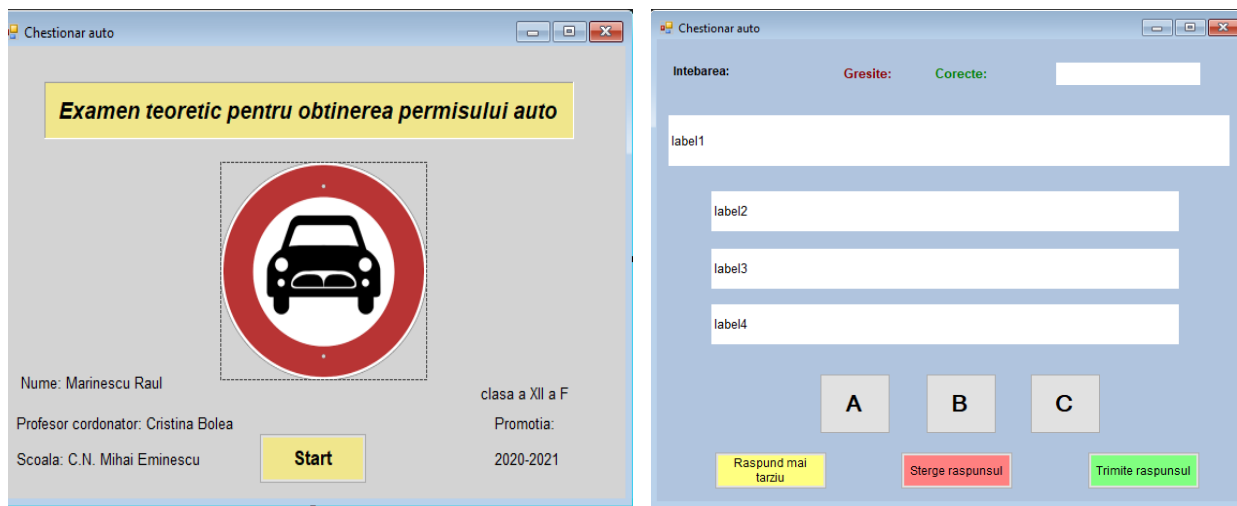
Cap II Descrierea lucrării

1. Reguli

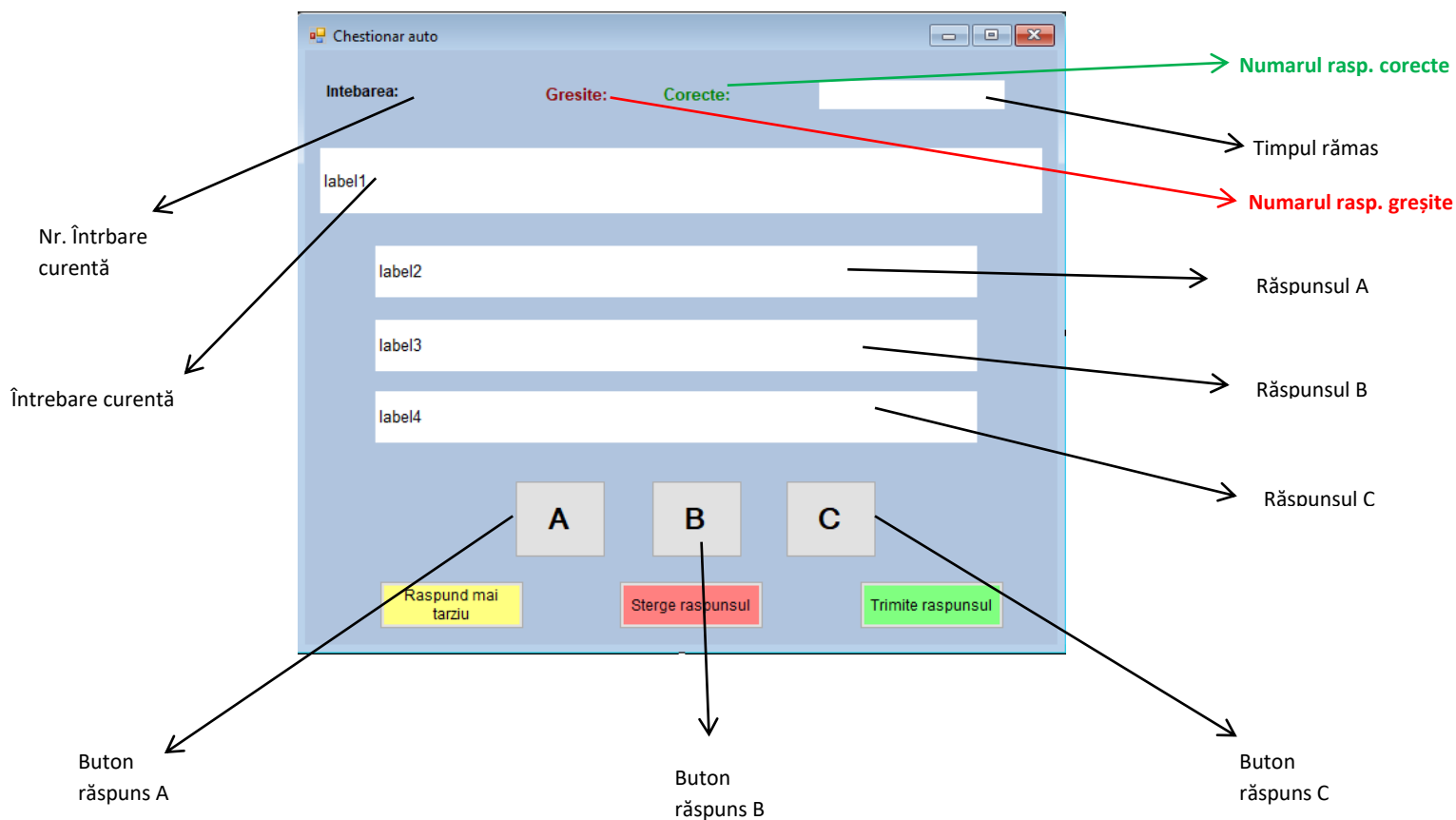
Întreg atestatul se rezumă la o serie de 26 de întrebări, ca și proba teoretică de examen în vederea obținerii permisului de conducere. Practic este un quiz cu 3 variante de răspuns, dintre care doar una este corectă (la examenul adevărat întrebările pot să aibă răspunsuri multiple). Timpul, maxim alocat pentru răspunderea la cele 26 de întrebări este de 30 de minute. În momentul apăsării butonului de START cronometrul pornește și prima întrebare este afișată într-un nou form.

- Pentru a selecta un răspuns, se efectuează click pe litera corespunzătoare cu cea de la începutul fiecărui răspuns;
- Pentru a selecta alt răspuns se selectează altă literă sau se poate șterge răspunsul de pe butonul „Ștergere răspuns”;
- Pentru a finaliza răspunsul se apasă butonul „Trimite răspuns”;
- Un răspuns trimis nu mai poate fi modificat;
- Pentru a trece peste întrebarea curentă și revenirea ulterioară după ce s-a răspuns la toate întrebările la acea întrebare se apasă butonul „Răspund mai târziu”;
- Dacă numărul de răspunsuri greșite este mai mare decât 4, candidatul va fi declarat respins;
- Examenul se promovează dacă numărul de răspunsuri corecte este mai mare sau egal cu 22;

2. Interfațele



3. Descrierea elementelor de pe interfață



Cap III Ilustrarea codului

```

private DateTime startTime;
int nrIntrebari;
int nrRaspCorect = 0;
int nrRaspGresit = 0;
int nrIntrebCurent = 1;
int RaspCurent=0;
interface
struct question
{
    public string text, ra, rb, rc;
    public int corect;
    public bool rasp;
}

}

question[] Q;
bool ok;
StreamReader f = new StreamReader("Intrebari.txt");
interface
void citire()
{
    string s;
    s = f.ReadLine();
    nrIntrebari = Convert.ToInt32(s);
    Q = new question[nrIntrebari + 1];
    for(int i=1;i<=nrIntrebari;i++)
    {
        Q[i].text = f.ReadLine();
        Q[i].ra = f.ReadLine();
        Q[i].rb = f.ReadLine();
        Q[i].rc = f.ReadLine();
        Q[i].corect = Convert.ToInt32(f.ReadLine());
        Q[i].resp = false;
    }
}
}

```

La începutul programului avem variabilele ce contorizează numărul întrebării curente, numărul răspunsurilor greșite și corecte și numărul total de întrebări. Pentru a reține răspunsul corect, dacă la întrebare s-a acordat un răspuns și fiecare variantă de răspuns, am declarat o structură de vector ce v-a avea dimensiunea numărului de întrebări. După structură avem declarat fișierul ce conține întrebările, variantele de răspuns și răspunsul corect și funcția de citire a fișierului.

```

1 reference
private void A_Click(object sender, EventArgs e)
{
    RaspCurent = 1;
    RaspunsA.BackColor= System.Drawing.Color.YellowGreen;
    RaspunsB.BackColor = System.Drawing.Color.White;
    RaspunsC.BackColor = System.Drawing.Color.White;
}
1 reference
private void B_Click(object sender, EventArgs e)
{
    RaspCurent = 2;
    RaspunsB.BackColor = System.Drawing.Color.YellowGreen;
    RaspunsA.BackColor = System.Drawing.Color.White;
    RaspunsC.BackColor = System.Drawing.Color.White;
}
1 reference
private void C_Click(object sender, EventArgs e)
{
    RaspCurent = 3;
    RaspunsC.BackColor = System.Drawing.Color.YellowGreen;
    RaspunsA.BackColor = System.Drawing.Color.White;
    RaspunsB.BackColor = System.Drawing.Color.White;
}
1 reference
private void button6_Click(object sender, EventArgs e)
{
    RaspCurent = 0;
    RaspunsA.BackColor = System.Drawing.Color.White;
    RaspunsB.BackColor = System.Drawing.Color.White;
    RaspunsC.BackColor = System.Drawing.Color.White;
}
1 reference

```

În această imagine sunt reprezentate butoanele de răspuns și ce se întâmplă în momentul în care fiecare este apăsăat. Pentru a evidenția mai bine care răspuns este selectat, acesta are culoarea verde în momentul selectării. Pentru a lucra mai ușor cu numere am atribuit fiecărei litere indicele său din alfabet, așadar, pentru A avem numărul 1, pentru B-2 și pentru C-3.

Aici este reprezentată colorarea răspunsului B când acesta este selectat.

```
private void button4_Click(object sender, EventArgs e)
{
    int i;
    Q[nrIntrbCurent].rasp = true;
    if (RaspCurent != 0)
    {
        if (RaspCurent == Q[nrIntrbCurent].corect)
            nrRaspCorect++;
        else
            nrRaspGresit++;
        if (nrRaspGresit > 4)
        {
            MessageBox.Show("Respins");
            timer1.Stop();
            Application.Exit();
        }
        else
        {
            ok = true;
            for(i=1;i<=nrIntrebari;i++)
                if(Q[i].rasp==false) { ok = false; break; }
            if (ok == true)
            {
                if (nrRaspCorect >= 22)
                    MessageBox.Show("Felicitari! Ai fost admis!");
                else
                    MessageBox.Show("Punctaj insuficient! Ai fost respins!");
                timer1.Stop();
                Application.Exit();
            }
            else
            {

```

Acesta este codul pentru butonul de "Trimite raspunsul". Variabila RaspCurent are mereu valoarea 0 până când este selectat un răspuns. Când acesta este apăsat se verifică dacă a fost selectat vre-un răspuns. În caz contrar nu se întâmplă nimic până când nu este trimis un răspuns valid.

În acest caz, primul lucru este de a marca întrebarea în vectorul de verificare a răspunsului ca fiind true, adică a fost atribuit un răspuns la aceasta. Dacă răspunsul este corect, variabila de răspunsuri corecte crește cu 1 iar în caz contrar variabila de răspunsuri greșite crește cu 1. Mai departe se face verificarea de fiecare

dată dacă numărul răspunsurilor greșite este mai mare decât 4. Dacă acest lucru se întâmplă programul ia sfârșit deoarece candidatul este picat.

În caz contrar se verifică dacă s-a acordat câte un răspuns la fiecare întrebare prin acel for și se verifică dacă candidatul este admis sau respins.

```

else
{
    ok = true;
    for(i=1;i<=nrIntrebari;i++)
        if(Q[i].rasp==false) { ok = false; break; }
    if (ok == true)
    {
        if (nrRaspCorect >= 22)
            MessageBox.Show("Felicitari! Ai fost admis!");
        else
            MessageBox.Show("Punctaj insuficient! Ai fost respins!");
        timer1.Stop();
        Application.Exit();
    }
    else
    {
        if (nrIntrbCurent == nrIntrebari)
            i = 1;
        else
            i = nrIntrbCurent + 1;

        while (i <= nrIntrebari)
        {
            if (Q[i].rasp == false)
            { nrIntrbCurent = i; break; }
            i++;
            if (i == nrIntrebari + 1) i = 1;
        }
        intrebare(nrIntrbCurent);
    }
}

RaspCurent = 0;
RaspunsA.BackColor = System.Drawing.Color.White;
RaspunsB.BackColor = System.Drawing.Color.White;
RaspunsC.BackColor = System.Drawing.Color.White;

```

Dacă există întrebări fără răspuns, partea încadrată în chenarul galben caută următoarea întrebare care nu are răspuns. Când numărul întrebării curente este egal cu 26 acesta trebuie resetat la 1 pentru a verifica întrebările la care s-au ales optinunea de a răspunde mai târziu.

La sfarsitul secvenței răspunsul este resetat la 0 și întrebarea selectată revine la culoarea inițială.

Codul butonului “Răspund mai tarziu” sare peste întrebarea k respectivă lăsând în vectorul de verificare a răspunsului k valoarea false. Caută printr-o valoare auxiliară “i” următoarea valoare fără răspuns după care se opreste si se apelează funcția întrebare.

K -este un numar oarecare;

```

private void button5_Click(object sender, EventArgs e)
{
    int i;
    if (nrIntrbCurent == nrIntrebari) i = 1;
    else
        i=nrIntrbCurent+1;
    while(i<=nrIntrebari)
    {
        if(Q[i].rasp==false)
        {
            nrIntrbCurent = i;
            break;
        }
        i++;
        if (i == nrIntrebari + 1) i = 1;
    }
    intrebare(nrIntrbCurent);
}

```

```

private void întrebare(int k)
{
    Intrebare.Text = Q[k].text;
    RaspunsA.Text = Q[k].ra;
    RaspunsB.Text = Q[k].rb;
    RaspunsC.Text = Q[k].rc;
    label1.Text = "Intrebarea: " + k;
    label2.Text = "Gresite: " + nrRaspGresit;
    label3.Text = "Corecte: " + nrRaspCorect;
}

```

Funcția întrebare afișează în form noua întrebare. Adică schimbă valorile din labeluri si variabile cu cele corespunzătoare din vectorului de structură în functie de parametrul primit. Paramentru este numărul întrebării curente.

Funcția timer1_Tick setează cronometrul la 30 de minute și în cazul in care cele 30 de minute au fost depășite se verifică numărul de răspunsuri corecte acumulate până în acel moment. Dacă acesta este mai mare sau egal cu 22 candidatul este considerat admis iar în caz contrar respins.

```

private void timer1_Tick(object sender, EventArgs e)
{
    int sec = (int)(DateTime.Now - startTime).TotalSeconds;
    int h = (30*60 - sec) / 60;
    int m = (30*60 - sec) % 60;
    if (sec <= 30*60) timpul.Text = String.Format("{0}:{1}", h,m);
    else
    {
        timer1.Stop();
        MessageBox.Show("Timpul a expirat!");
        if(nrRaspCorect>=22)
            MessageBox.Show("Felicitari! Ai fost admis!");
        else
            MessageBox.Show("Punctaj insuficient! Ai fost respins!");
    }
}

```