****

**ATESTAT PROFESIONAL**

**INFORMATICĂ**

#### Soft învățare și relaxare

**Îndrumător:**

##### **Prof. Maier Cornelia**

##### ***Elev:***

**Adam Paul Adrian**

**XII Matematică-Informatică**

**2020**

Cuprins:

[1. Motivarea lucrării 1](#_Toc513577553)

[2. Prezentarea generală a temei 1](#_Toc513577554)

[3. Resurse hardware și software necesare 1](#_Toc513577555)

[4. Teoria programării orientate pe obiect și a bazelor de date 2](#_Toc513577556)

[4.1. Principiile programării orientate pe obiect 2](#_Toc513577557)

[4.2. Concepte de bază ale programării vizuale 2](#_Toc513577558)

[4.3. Accesarea și prelucrearea datelor prin intermediul SQL Server 3](#_Toc513577559)

[5. Descrierea aplicației 5](#_Toc513577560)

[5.1 Formul „MeniuPrincipal”](#_Toc513577561) 5

[5.2. Formul „InformatiiAtestat”](#_Toc513577562) 5

[5.3. Formurile „Form1”, „MeniuLabirint”, „MeniuNivPredef”](#_Toc513577563) 6

[5.4. Formurile „JoacaCuMouse”, „JoacaCuMouseDrag”](#_Toc513577564) 8

[5.5. Formurile „Niv1Predef”, „Niv2Predef”, „Niv3Predef”](#_Toc513577565) 9

[5.6. Formul „NivRandom”](#_Toc513577566) 13

[5.7. Formurile „DragUsor”, „DragMediu”, „DragGreu”](#_Toc513577566) 18

[5.8. Formul „Teorie”](#_Toc513577566) 21

[5.9. Formurile „Cap1”, „Cap2”, „Cap3”](#_Toc513577566) 21

[5.10. Formul „QuizVerificare”](#_Toc513577566) 22

[5.11. Formurile „QuizCap1”, „QuizCap2”, „QuizCap3”](#_Toc513577566) 22

[6. Webografie](#_Toc513577567) 25

1. Motivarea lucrării

Această lucrare are ca scop crearea unei platforme speciale de învățare, ușor accesibilă de către orice persoană. Am ales să fac această aplicație deoarece, în trecut, am căutat un astfel de program și am găsit foarte puține.

În acest material este descris procesul prin care a trecut această platformă pentru a fi creată, de la modelul fizic și până la aplicația propriu-zisă. Pentru o mai bună receptare a conținutului proiectului, m-am folosit de print-screen-uri cu rol explicativ și informative asupra acestui proiect de atestat.

1. Prezentarea generală a temei

După cum am menționat mai sus, această platformă prezintă un mediu de învățare inovativ, împărțit în 3 categorii: „Învățare”, „Quiz-uri de verificare” și „Recreație”. După ce înveți ceva nou îți poți testa cunoștințele dobândite, iar dacă ai nevoie de o pauză te poți juca unul dintre jocurile puse la dispoziție de producătorul platformei. Oamenii nu se opresc niciodată din învățat, de aceea acest proiect își propune să creeze un mod mai plăcut de a învăța.

1. Resurse hardware și software necesare

În vederea rulării aplicației, este necesar un calculator obișnuit care să aibă instalat orice versiune de Windows începând cu Windows 7. De asemenea, este necesar programul Microsoft Visual Studio anul 2017 sau 2019, fără de care aplicația nu poate fi rulată.

1. Teoria programării orientate pe obiect și a bazelor de date
   1. **Principiile programării orientate pe obiect**

Ideea POO este de a crea programele ca o colecţie de obiecte, unităţi individuale de cod care interacţionează unele cu altele, în loc de simple liste de instrucţiuni sau de apeluri de proceduri.

Obiectele POO sunt, de obicei, reprezentări ale obiectelor din viaţa reală (domeniul problemei), astfel încât programele realizate prin tehnica POO sunt mai uşor de înţeles, de depanat şi de extins decât programele procedurale. Aceasta este adevărată mai ales în cazul proiectelor software complexe şi de dimensiuni mari.

Principiile POO sunt:

* Abstractizarea – principiu care permite identificarea caracteristicilor şi comportamentului obiectelor ce ţin nemijlocit de domeniul problemei. Rezultatul este un model. În urma abstractizării, entităţile din domeniul problemei se definesc prin clase.
* Încapsularea – numită şi ascunderea de informaţii, este caracterizată prin 2 aspecte: a. Gruparea comportamentelor şi caracteristicilor într-un tip abstract de date; b. Definirea nivelului de acces la datele unui obiect.
* Moştenirea – organizează şi facilitează polimorfismul şi încapsularea permiţând definirea si crearea unor clase specializate plecând de la clase (generale) care sunt deja definite - acestea pot împărtăşi (şi extinde) comportamentul lor fără a fi nevoie de redefinirea aceluiaşi comportament.
* Polimorfismul – posibilitatea mai multor obiecte dintr-o ierarhie de clase de a utiliza denumiri de metode cu acelaşi nume dar, cu un comportament diferit.
  1. **Concepte de bază ale programării vizuale**

Programarea vizuală trebuie privită ca un mod de proiectare a unui program prin operare directă asupra unui set de elemente grafice (de aici vine denumirea de programare vizuală). Această operare are ca efect scrierea automată a unor secvenţe de program, secvenţe care, împreună cu secvenţele scrise textual vor forma programul.

Spunem că o aplicaţie este vizuală dacă dispune de o interfaţă grafică sugestivă şi pune la dispoziţia utilizatorului instrumente specifice de utilizare (drag, click, hint etc.).

Realizarea unei aplicaţii vizuale nu constă doar în desenare şi aranjare de controale, ci presupune în principal stabilirea unor decizii arhitecturale, decizii ce au la bază unul dintre modelele arhitecturale de bază.

În realizarea aplicaţiei mai trebuie respectate şi principiile proiectării interfeţelor:

* Simplitatea: Interfaţa trebuie să fie cât mai uşor de înţeles şi de învăţat de către utilizator şi să permită acestuia să efectueze operaţiile dorite în timp cât mai scurt. În acest sens, este vitală culegerea de informaţii despre utilizatorii finali ai aplicaţiei şi a modului în care aceştia sunt obişnuiţi să lucreze.
* Poziţia controalelor: Locaţia controalelor dintr-o fereastră trebuie să reflecte importanţa relativă şi frecvenţa de utilizare. Astfel, când un utilizator trebuie să introducă nişte informaţii – unele obligatorii şi altele opţionale – este indicat să organizăm controalele astfel încât primele să fie cele care preiau informaţii obligatorii.
* Consistenţa: Ferestrele şi controalele trebuie să fie afişate după un design asemănător („template”) pe parcursul utilizării aplicaţiei. Înainte de a implementa interfaţa, trebuie decidem cum va arăta aceasta, să definim „template”-ul.
* Estetica: Interfaţa trebuie să fie pe cât posibil plăcută şi atrăgătoare.
  1. **Accesarea și prelucrearea datelor prin intermediul SQL Server**

Accesarea şi prelucrarea datelor cu ajutorul mediului vizual Mediul de dezvoltare Visual Studio dispune de instrumente puternice şi sugestive pentru utilizarea bazelor de date în aplicaţii. Conceptual, în spatele unei ferestre în care lucrăm cu date preluate dintr-una sau mai multe tabele ale unei baze de date se află obiectele din categoriile Connection, Command, DataAdapter şi DataSet prezentate.

După crearea unei baze de date în SQL informaţiile înregistrate în tabela sau tabelele bazei de date pot fi utilizate într-o aplicaţie din Visual C# într-un formular sau într-o aplicaţie consolă. Vom prezenta acum modul în care se poate utiliza o bază de date într-un formular creat în Windows Forms.

Atunci când într-un formular utilizăm un tabel trebuie să avem posibilitatea de a utiliza funcţiile ce operează asupra datelor incluse în el. Toate instrucţiunile prezentate în capitolul Introducere în limbajul SQL pot fi accesate şi pe un formular. Prin "tragerea" unor obiecte din fereastra Data Sources în fereastra noastră nouă, se creează automat obiecte specifice. În partea de jos a figurii se pot observa obiectele de tip Dataset, TableAdapter, BindingSource, BindingNavigator şi, în fereastră, TableGridView. BindingNavigator este un tip ce permite, prin instanţiere, construirea barei de navigare care facilitează operaţii de deplasare, editare, ştergere şi adăugare în table etc.

De obicei, sursa de date este o bază de date, dar ar putea de asemenea să fie un fişier text, o foaie Excel, un fişier Access sau un fişier XML. In aplicaţiile tradiţionale cu baze de date, clienţii stabilesc o conexiune cu baza de date şi menţin această conexiune deschisă până la încheierea executării aplicaţiei. Conexiunile deschise necesită alocarea de resurse sistem. Atunci când menţinem mai multe conexiuni deschise server-ul de baze de date va răspunde mai lent la comenzile clienţilor întrucât cele mai multe baze de date permit un număr foarte mic de conexiuni concurente. ADO.NET permite şi lucrul în stil conectat dar şi lucrul în stil deconectat, aplicaţiile conectându-se la server-ul de baze de date numai pentru extragerea şi actualizarea datelor. Acest lucru permite reducerea numărului de conexiuni deschise simultan la sursele de date. ADO.NET oferă instrumentele de utilizare şi reprezentare XML pentru transferul datelor între aplicaţii şi surse de date, furnizând o reprezentare comună a datelor, ceea ce permite accesarea datelor din diferite surse de diferite tipuri şi prelucrarea lor ca entităţi, fără să fie necesar să convertim explicit datele în format XML sau invers.

Aceste caracteristici sunt determinate în stabilirea beneficiilor furnizate de ADO.NET: Interoperabilitate. ADO.NET poate interacţiona uşor cu orice componentă care suportă XML.

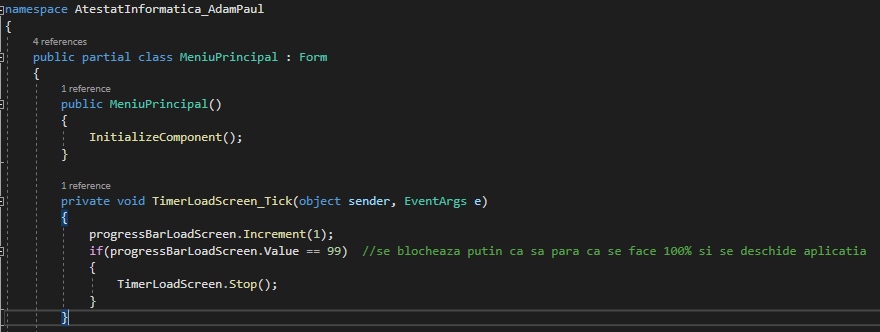
* Durabilitate. ADO.NET permite dezvoltarea arhitecturii unei aplicaţii datorită modului de transfer a datelor între nivelele arhitecturale.
* Programabilitate. ADO.NET simplifică programarea pentru diferite task-uri cum ar fi comenzile SQL, ceea ce duce la o creştere a productivităţii şi la o scădere a numărului de erori.
* Performanţă. Nu mai este necesară conversia explicită a datelor la transferul între aplicaţii, fapt care duce la creşte performanţelor acestora.
* Accesibilitate. Utilizarea arhitecturii deconectate permite accesul simultan la acelaşi set de date. Reducerea numărului de conexiuni deschise simultan determină utilizarea optimă a resurselor.

**!!! În proiectul meu de atestat nu am recurs la folosirea bazelor de date!**

1. Descrierea aplicației

Această aplicație este alcătuită din 22 de formuri Windows și 3 clase.

5.1 Formul „MeniuPrincipal”

 Acest form este un “loading screen”. De fiecare data când aplicația este deschisă, prima dată va apărea acest form cu mesajul “Welcome” și un *progressBar* care se încarcă folosind un *timer*.

5.2 Formul „InformațiiAtestat”

Acest form conține informații despre atestat pe care le întâlnim și pe prima pagină a acestui Word.

Pe lângă aceste informații, formul arată și data exacta în timpul rulării programului (oră, lună, an), conține un buton de mers înapoi la formul anterior și unul de închis aplicația. (!Data și butoanele de închis și mers înapoi sunt prezente în aproape toate formurile aplicației!)

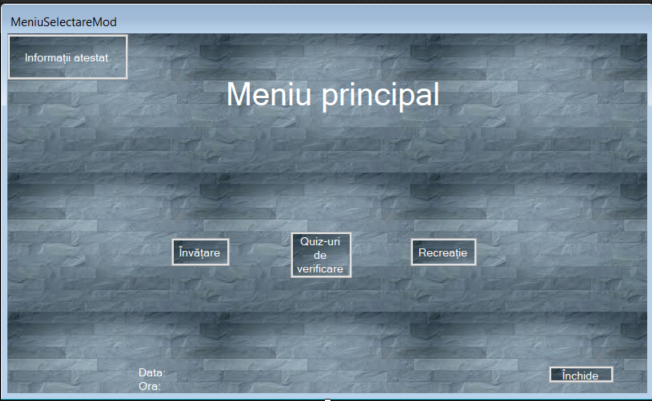


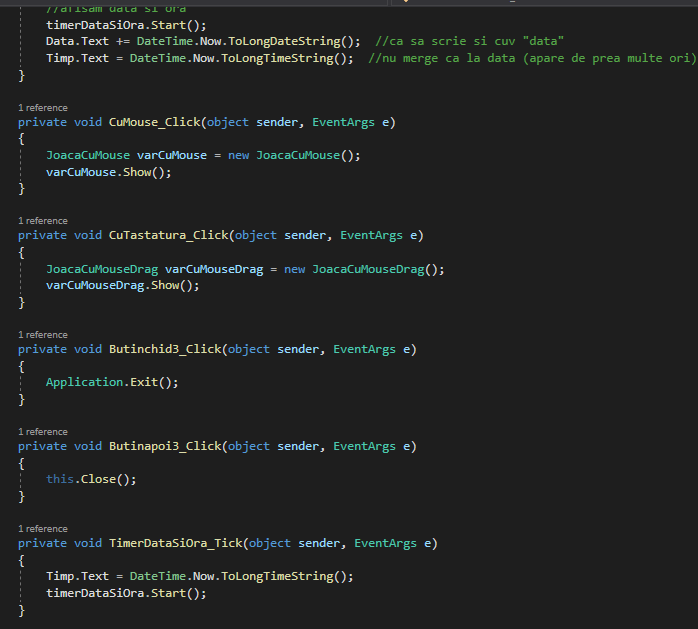


5.3 Formurile „Form1”, „MeniuLabirint”, „MeniuNivPredef”

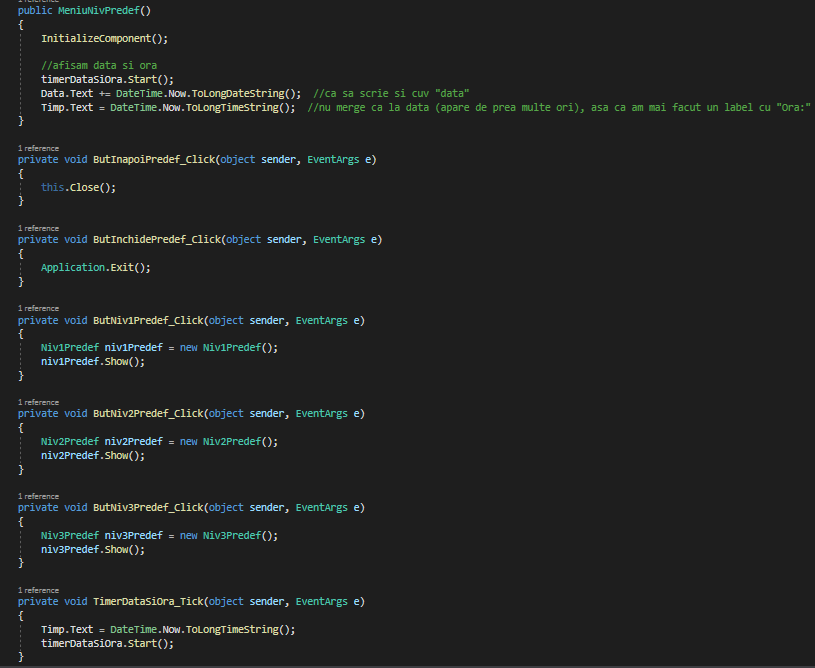
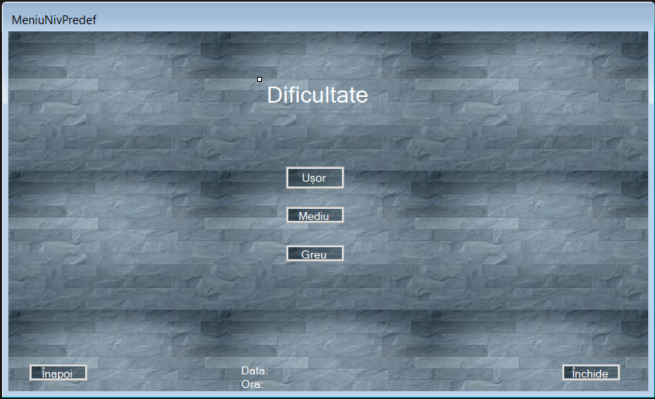
Aceste 3 formuri au rol de ghidare în aplicație.

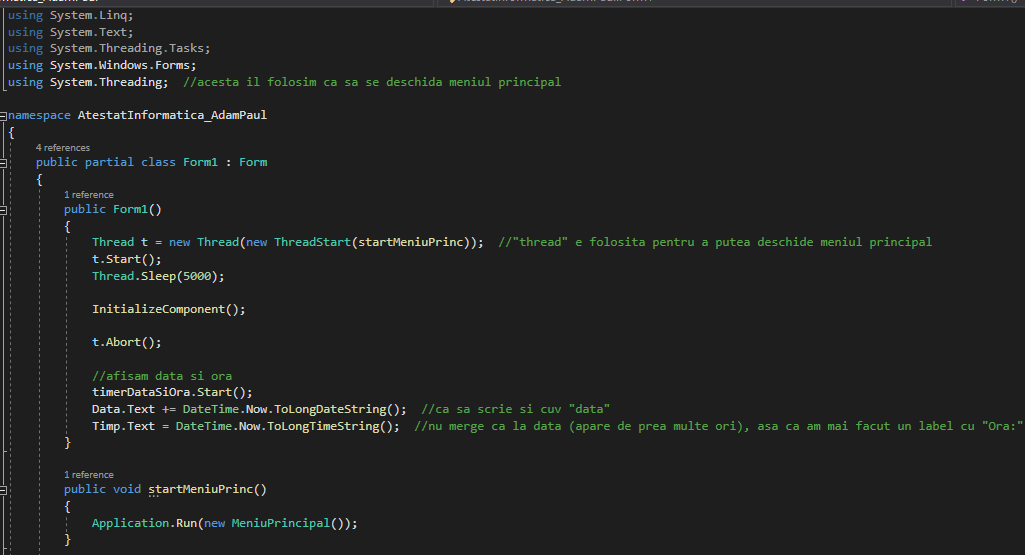
În “Form1” există patru opțiuni: cele trei pe care se bazează aplicația, adică “Învățare”, “Quiz-uri de verificare” și “Recreație” + “Informații atestat”.



În “MeniuLabirint” sunt două opțiuni: “Joacă cu mouse-ul\_Click” și “Joacă cu mouse-ul\_Drag”, alături de descrierea tipurilor de joc.

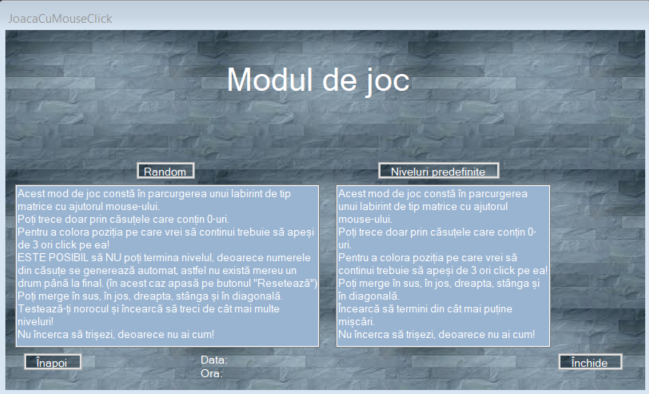


În “MeniuNivPredef” există posibilitatea alegerii nivelului de dificultate al jocului din “Joacă cu mouse-ul\_Click”.

!“Form1” este mereu primul form care se deschide când rulăm aplicația, astfel el conține cod suplimentar care se leagă cu *Loading Screen-ul.*

5.4 Formurile „JoacaCuMouse”, „JoacaCuMouseDrag”

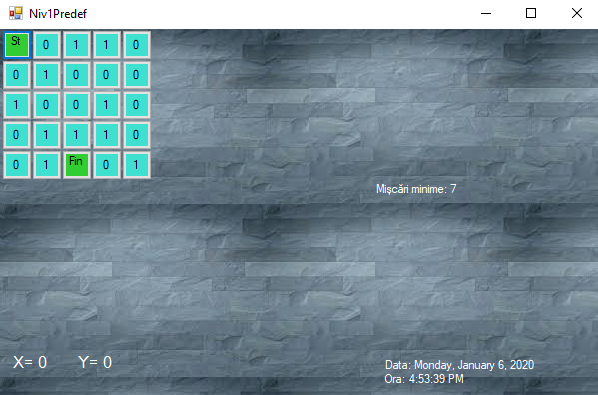
Formul “JoacaCuMouse”(Click) permite alegerea unuia dintre cele două moduri de joc (“Random” și “Niveluri predefinite”) și descrie tipurile jocului.

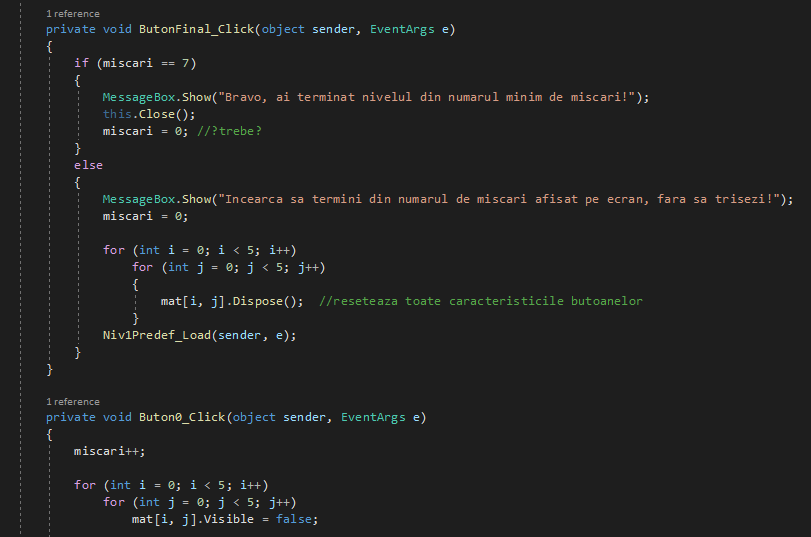
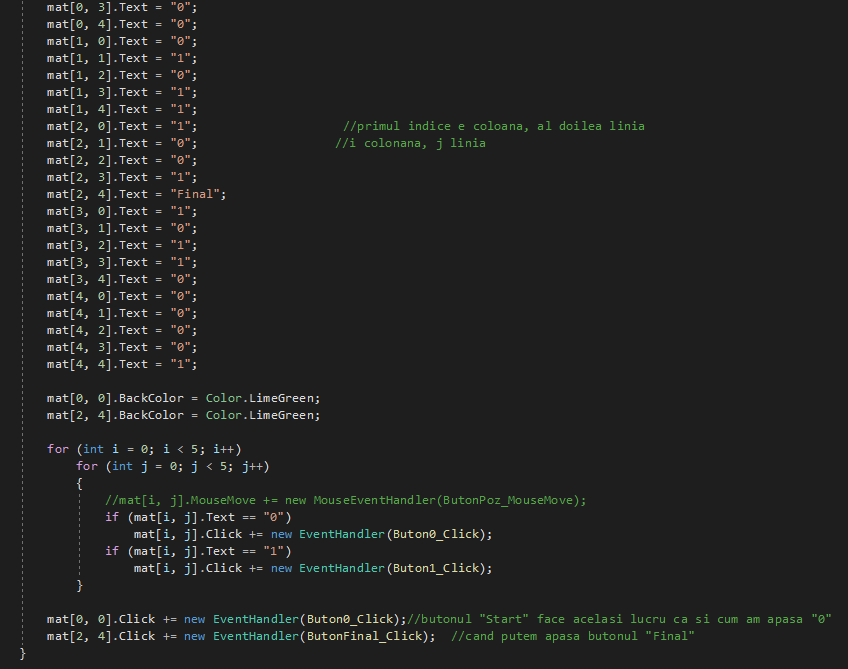
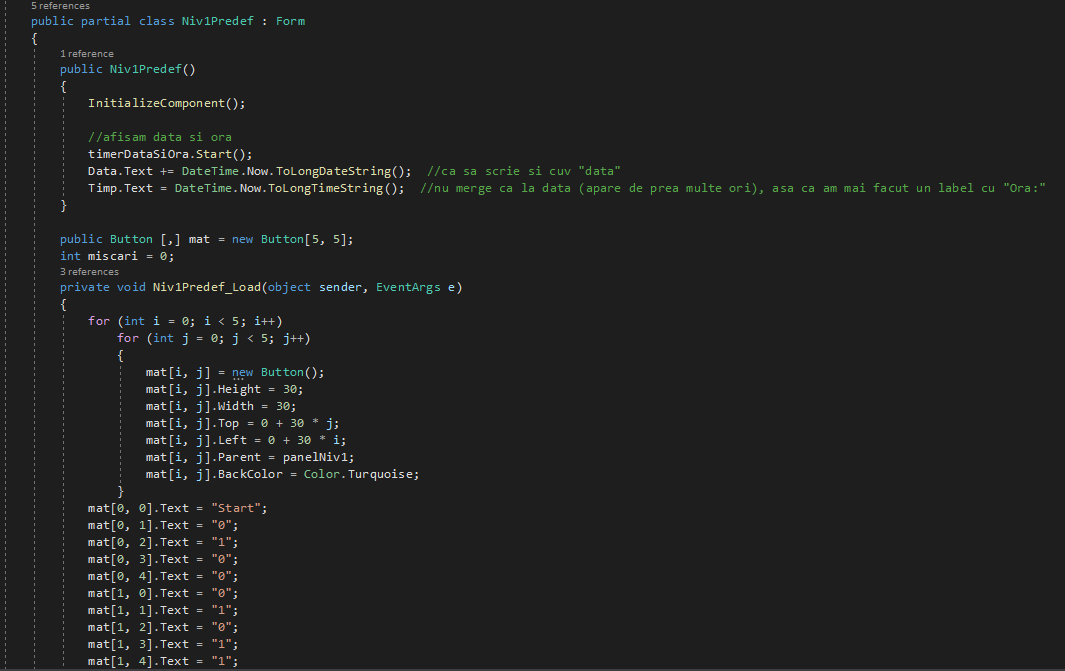


 Formul “JoacaCuMouseDrag” oferă posibilitatea alegerii gradului de dificultate, descrierea tipului de joc aflându-se în formul „MeniuLabirint”.

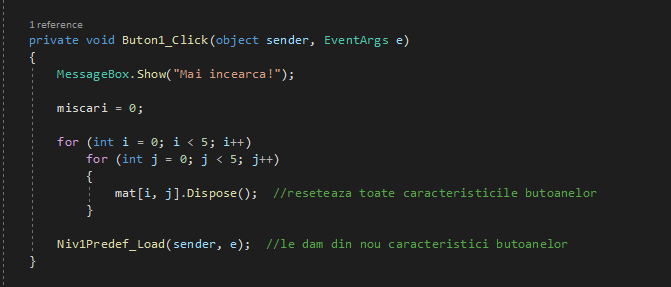
5.5 Formurile „Niv1Predef”, „Niv2Predef”, „Niv3Predef”

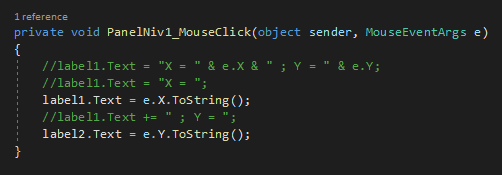
Aceste 3 formuri sunt în funcție de dificultate. Acestea sunt nivelurile predefinite ale tipului de joc “JoacaCuMouse\_Click”. Ele conțin un *panel* pe care sunt create niște *butoane* care trebuie traversate astfe încât să ajungem la finalul labirintului. Pe ecran este afișat din câte mișcări trebuie să terminăm nivelul.

 Exemplu de cod și interfață pentru “Niv1Predef”: (și celelalte două sunt aprox. la fel)

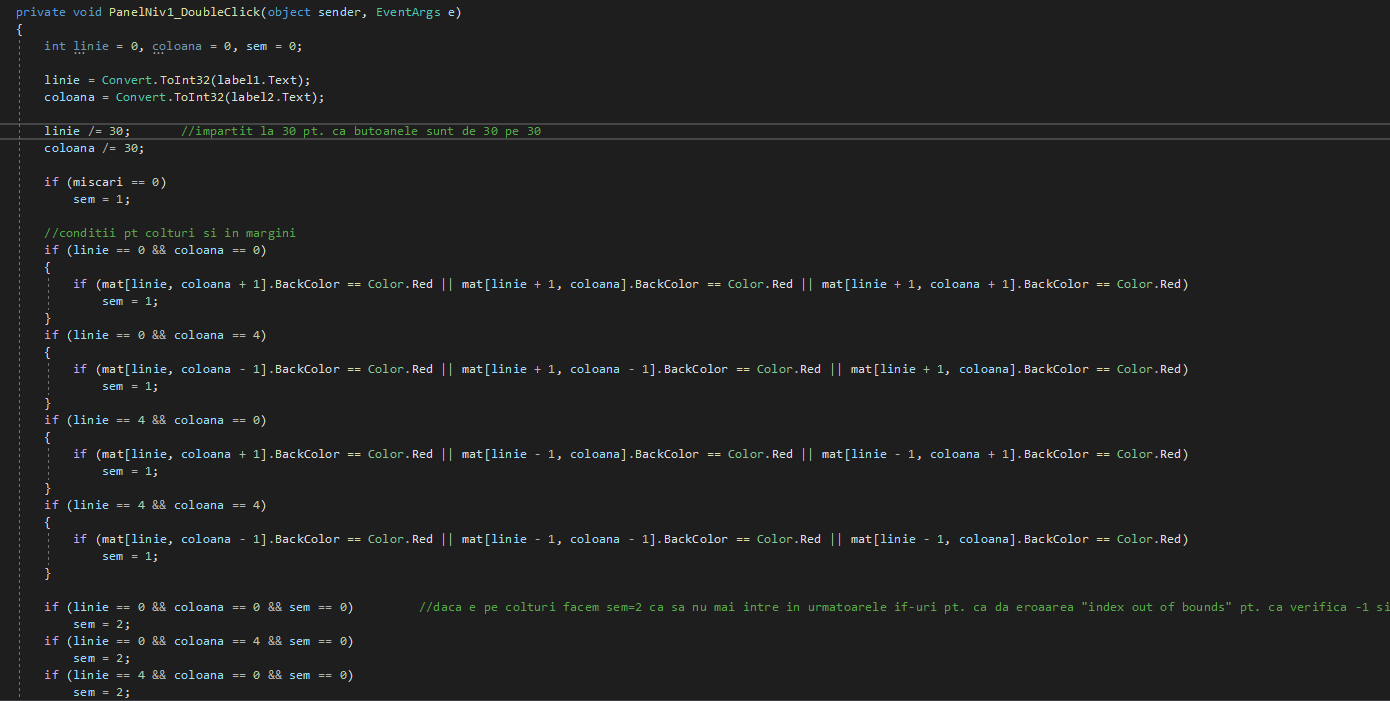
 În subprogramul Niv1Predef\_Load() creăm butoanele, le dăm caracteristici și le atribuim funcții (prezentate mai jos).

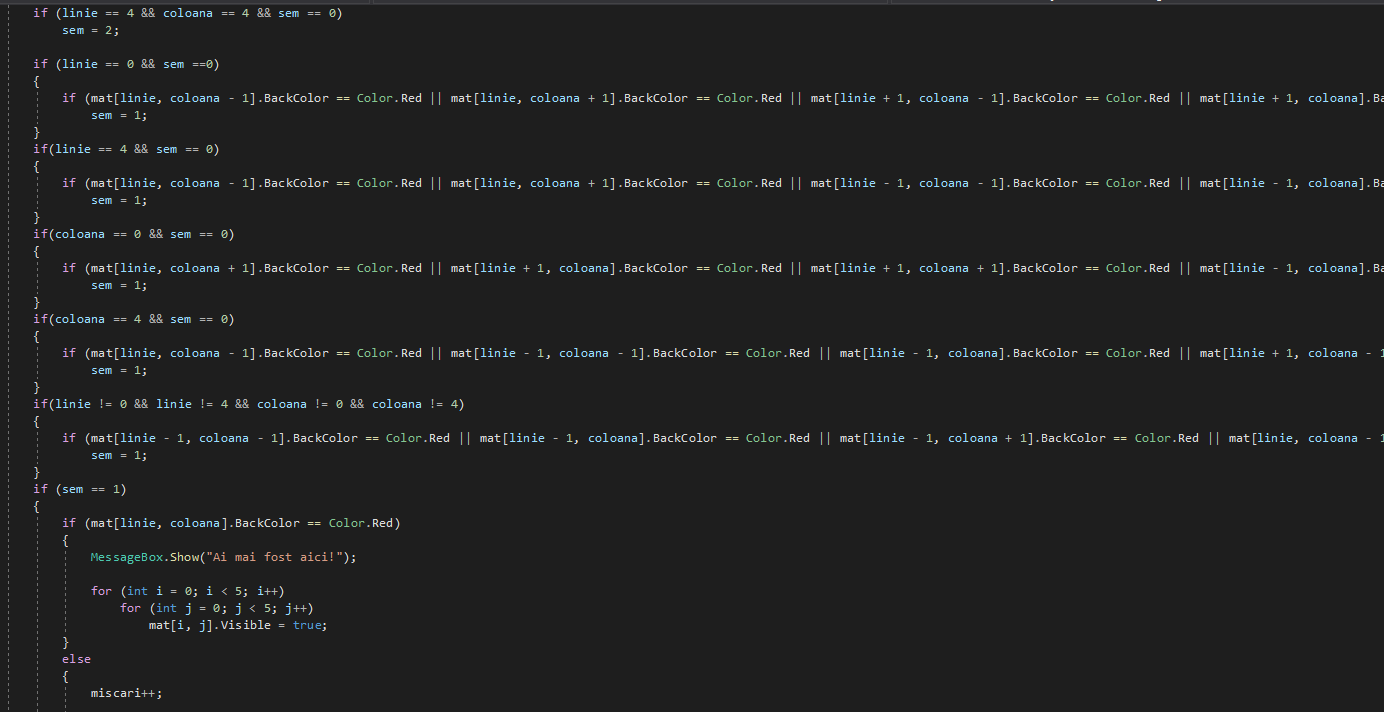
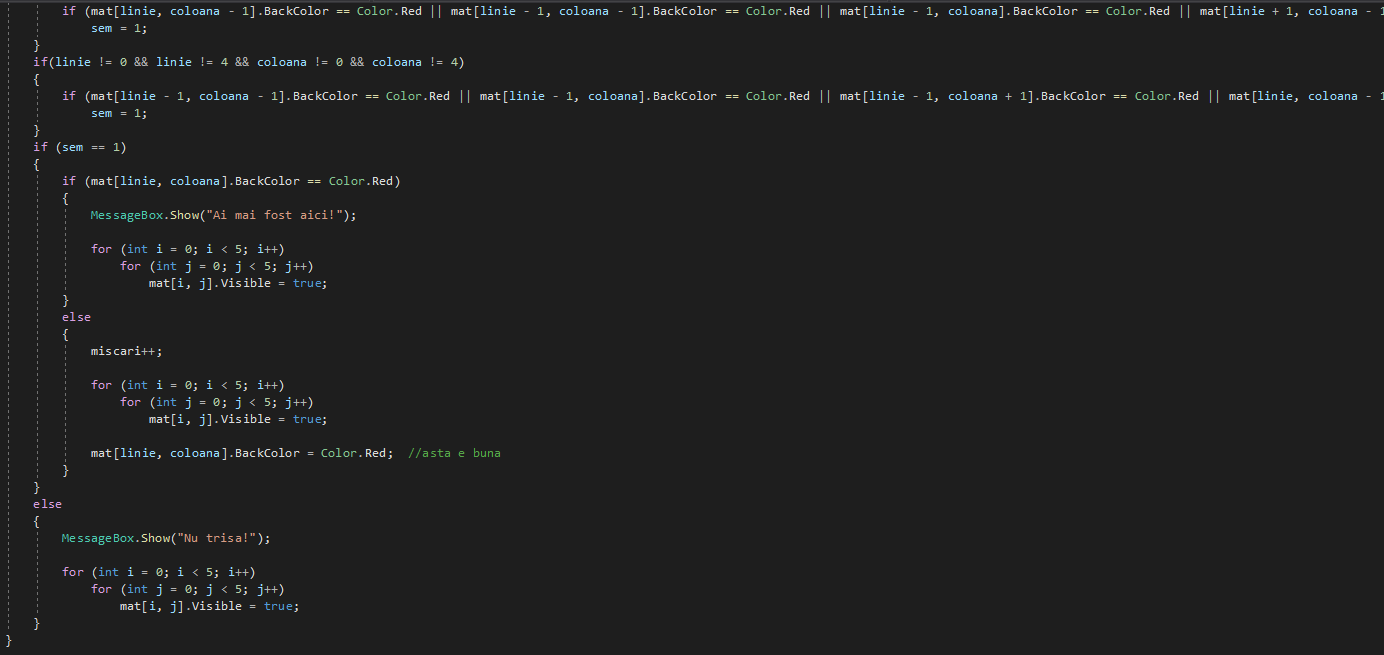
Funcția ButonFinal\_Click() este atribuită butonului cu textul „Final” și verifică dacă am terminat nivelul din numărul de mișcări indicat pe ecran, iar funcția Buton0\_Click() este atribuită butoanelor cu textul ”0” (poți trece) și face butoanele invizibile pentru a putea lua coordonatele panel-ului și pentru a afla pe ce buton am apăsat, lucru pe care o să îl prezint mai mai jos, la funcția PanelNiv1\_DoubleClick().

 Funcția Buton1\_Click() este atribuită butoanelor cu textul “1” (nu poți trece) și resetează caracteristicile butoanelor (reia jocul de la început).

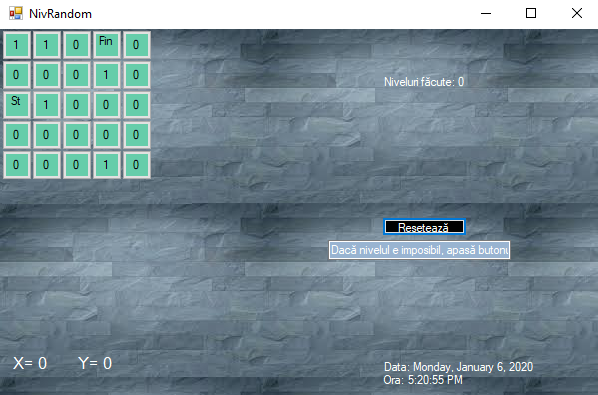


Subprogramul PanelNiv1\_MouseClick() preia coordonatele X și Y ale mouse-ului și le pune în două label-uri diferite.

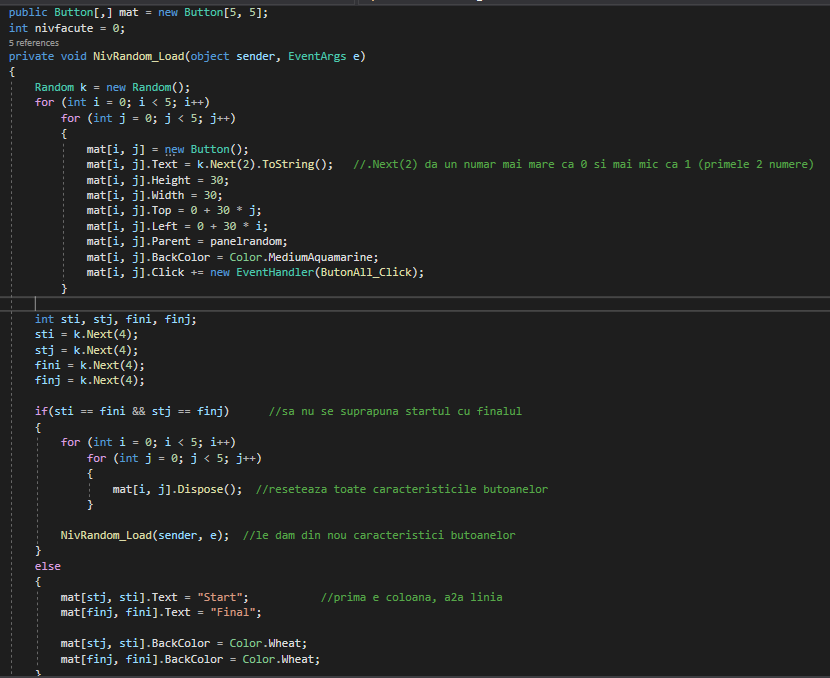


Subprogramul PanelNiv1\_DoubleClick() găsește linia și coloana butonului pe care am apăsat și verifică ca jucătorul să nu trișeze. Dacă jucătorul nu trișează, funcția va colora butonul apăsat cu roșu. După verificări, butoanele vor fi din nou vizibile.

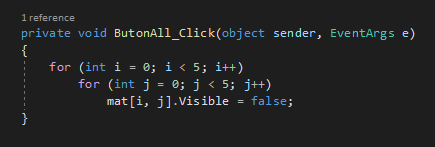
5.6 Formul „NivRandom”

 Acest form este nivelul random ale tipului de joc “JoacaCuMouse\_Click”. El conține un *panel* pe care sunt create niște *butoane* care trebuie traversate astfe încât să ajungem la finalul labirintului. Pe ecran este afișat câte nivele reușim să facem și mai este prezent butonul “Resetează”, pe care îl putem apăsa când labirintul nu are soluție.

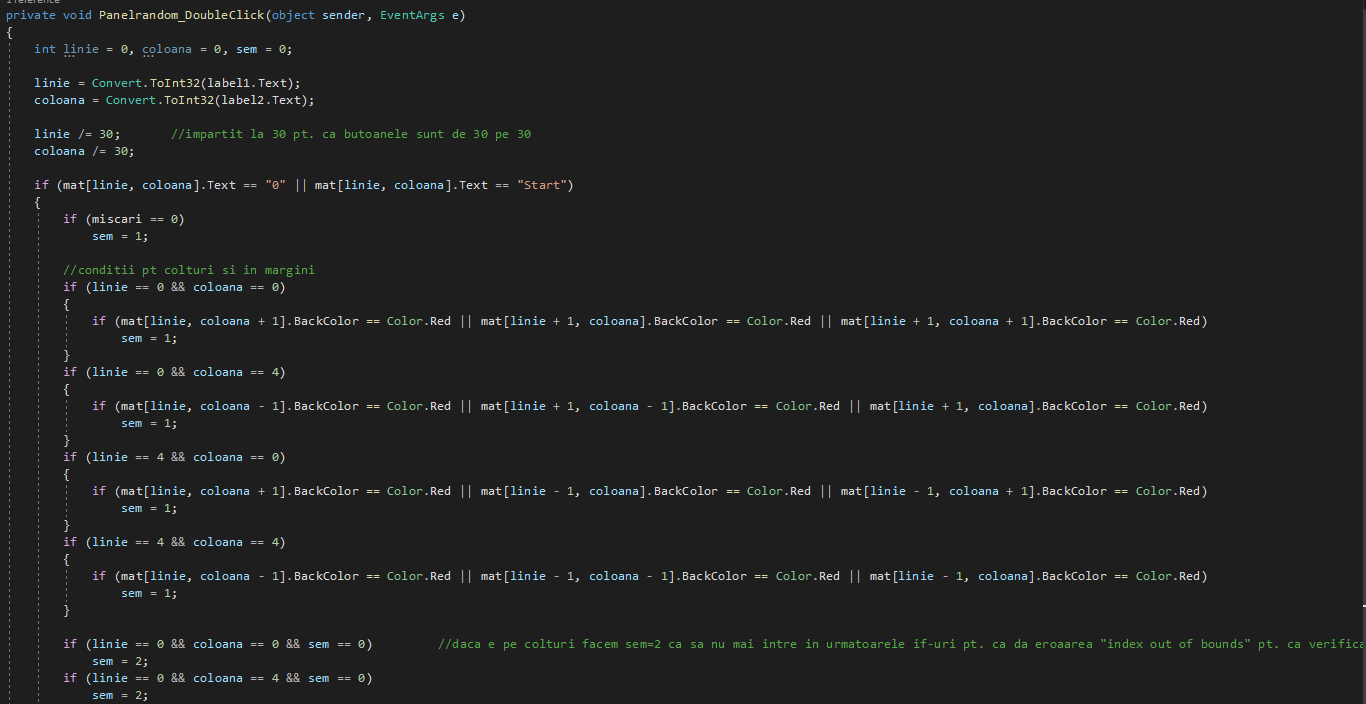
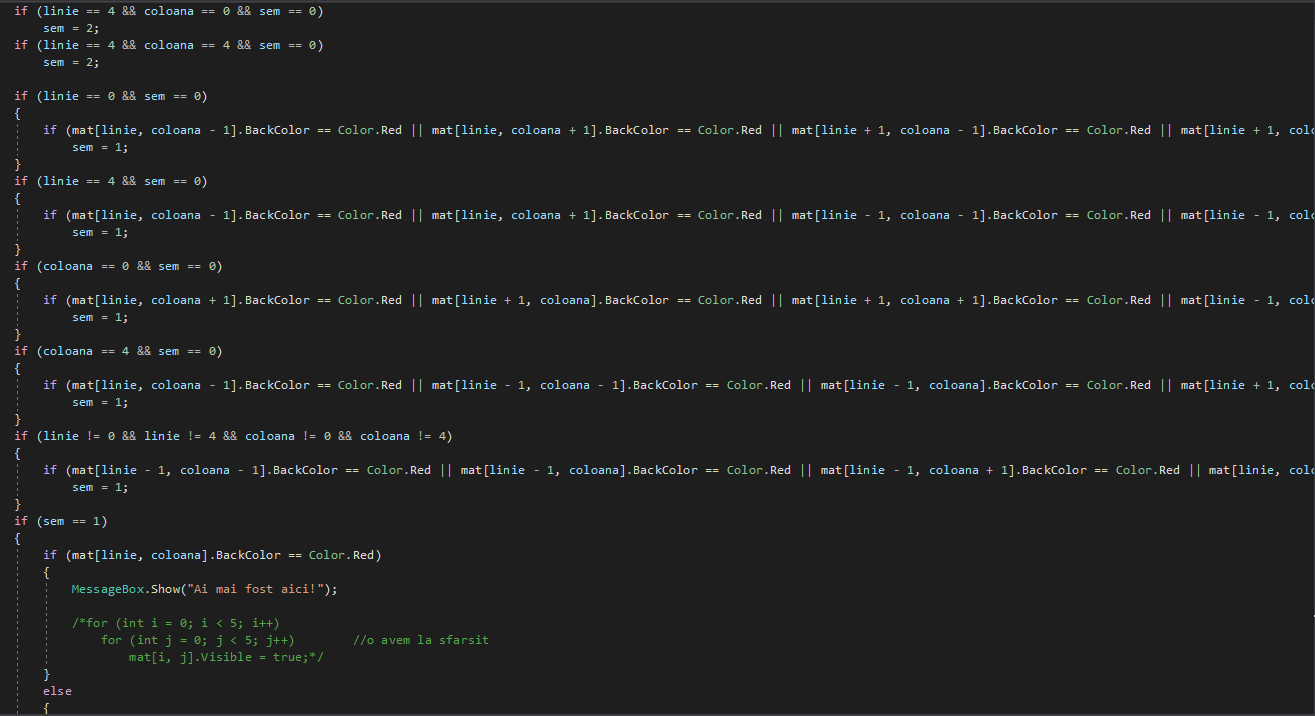
Aceasta este intefața acestui tip de joc.

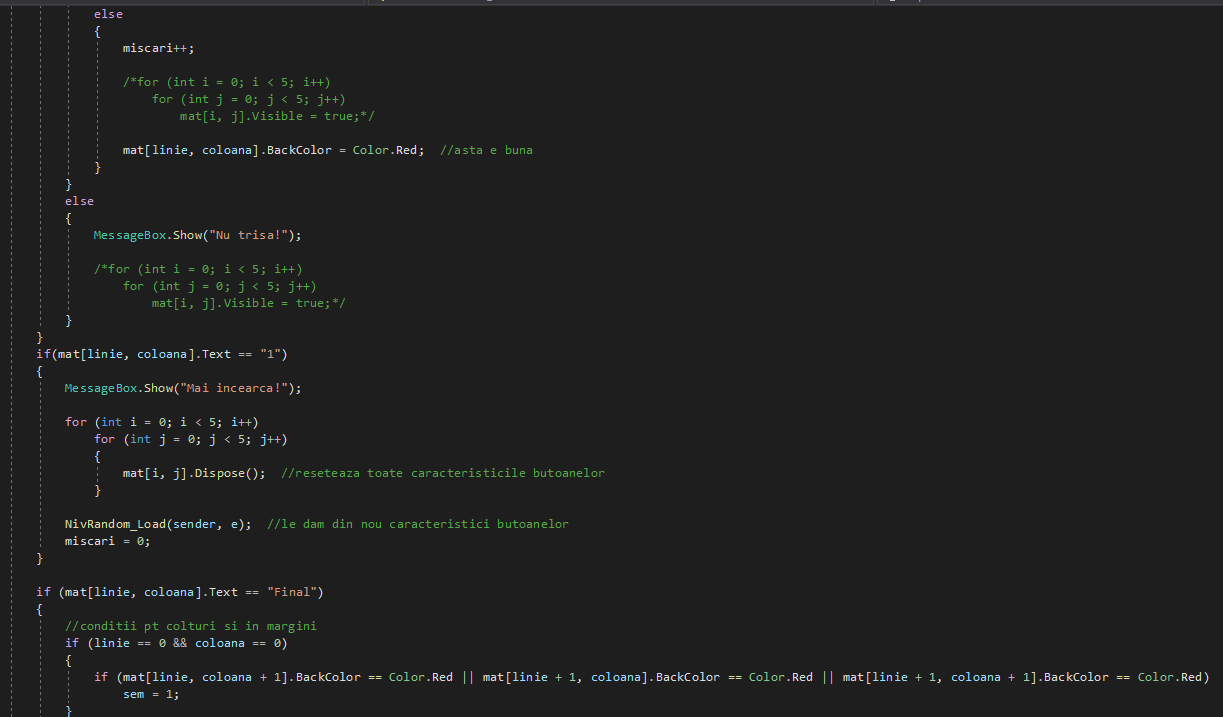
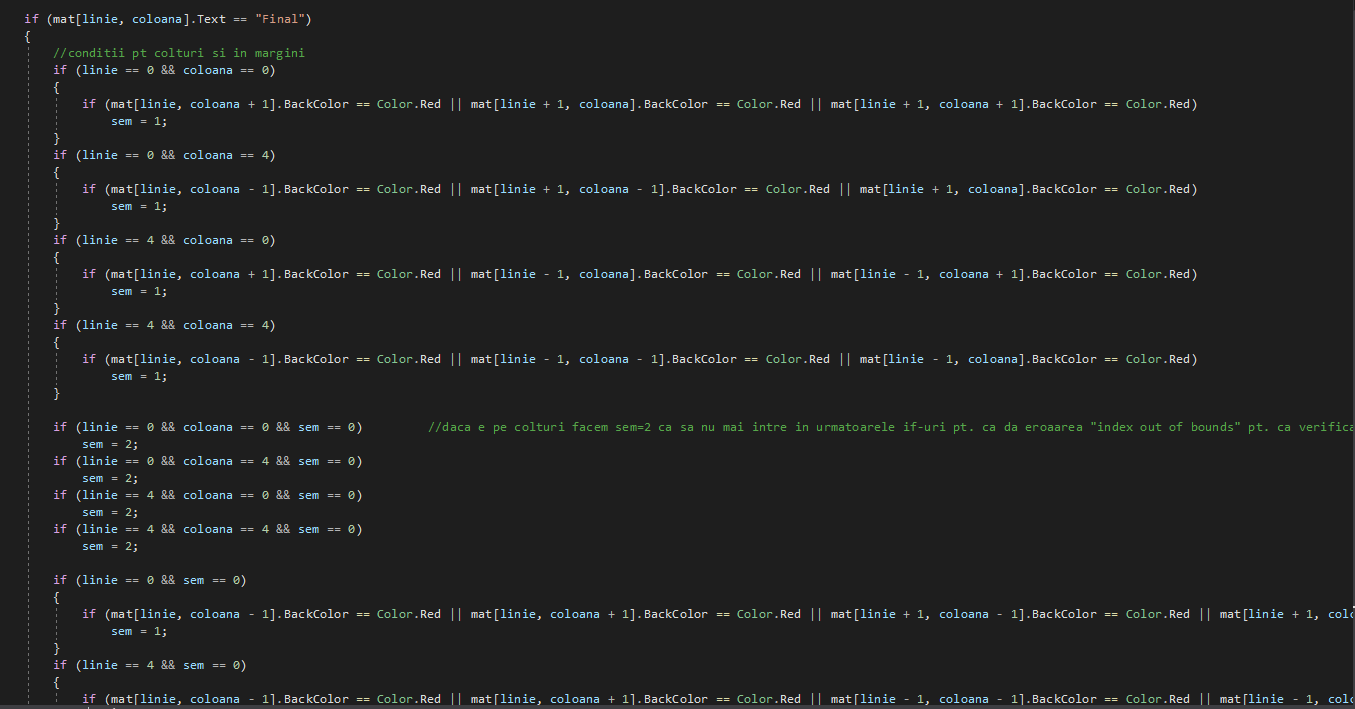
 Suprogramul NivRandom\_Load() creează butoanele, le dă caracteristici și folosește o variabilă de tip *Random* pentru a stabili ce butoane primesc “0”, “1”, “Start” și “Final”.

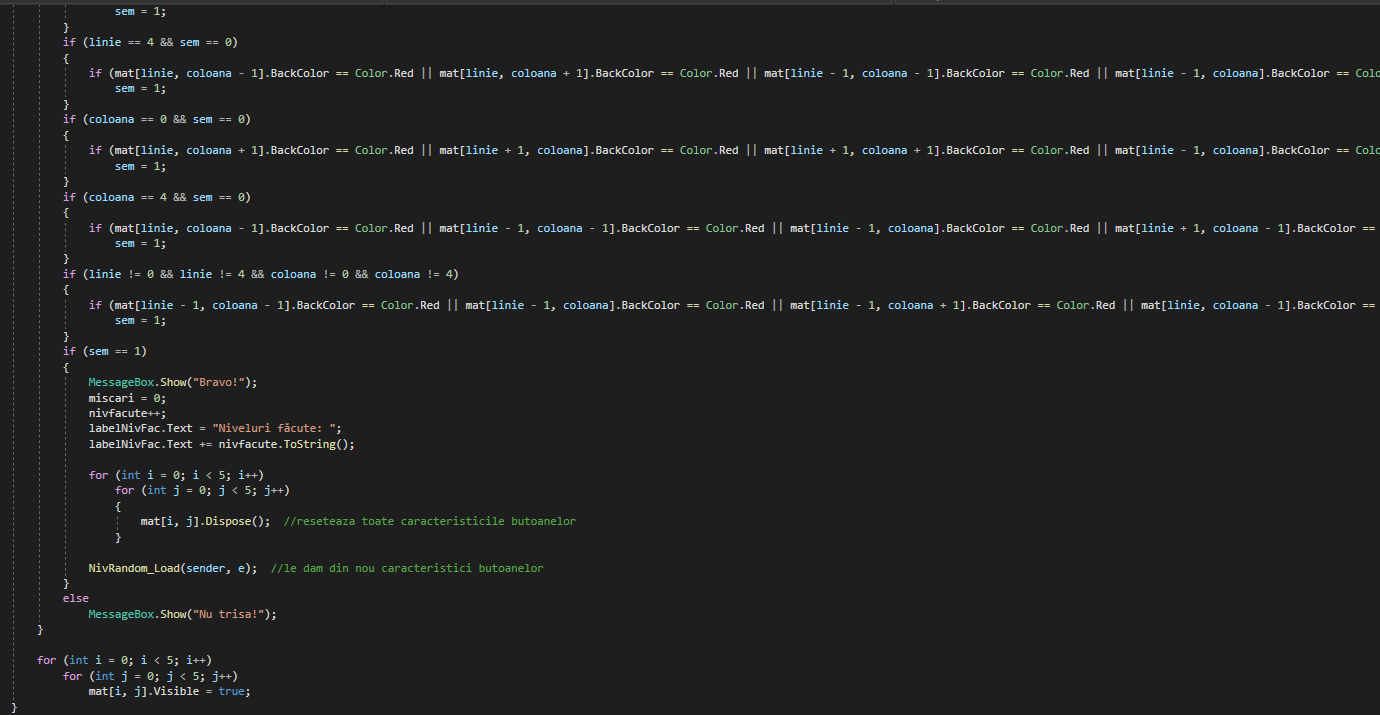
Totodată, funcția verifică dacă butoanele “Start” și “Final” sunt diferite. Dacă sunt aceleași (se suprapun), atunci butoanele se resetează și primesc alte caracteristici.

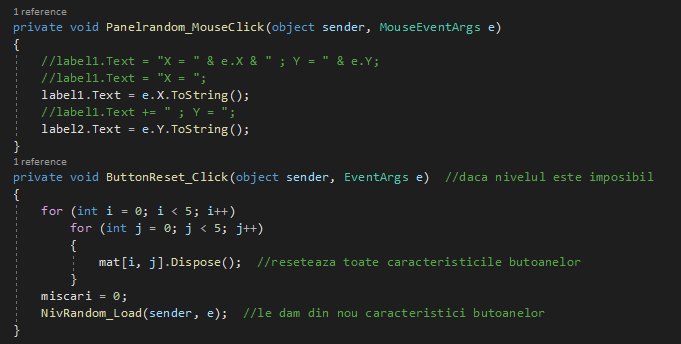


Funcția ButonAll\_Click() este folosită de fiecare data când este apăsat un buton și aceasta are rolul de a face butoanele să dispară de pe ecran pentru ca mai apoi (în funcția Panelrandom\_DoubleClick()) să aflăm linia și coloana butonului apăsat.



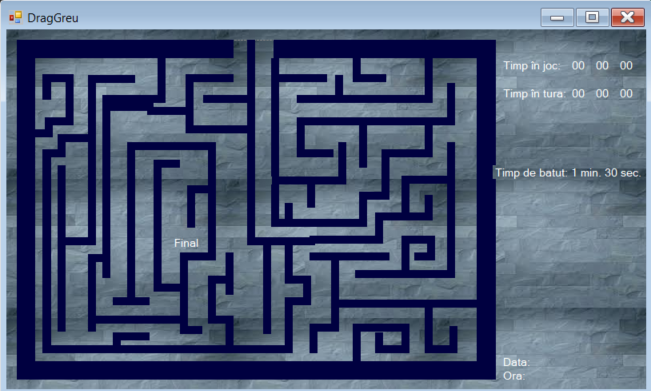


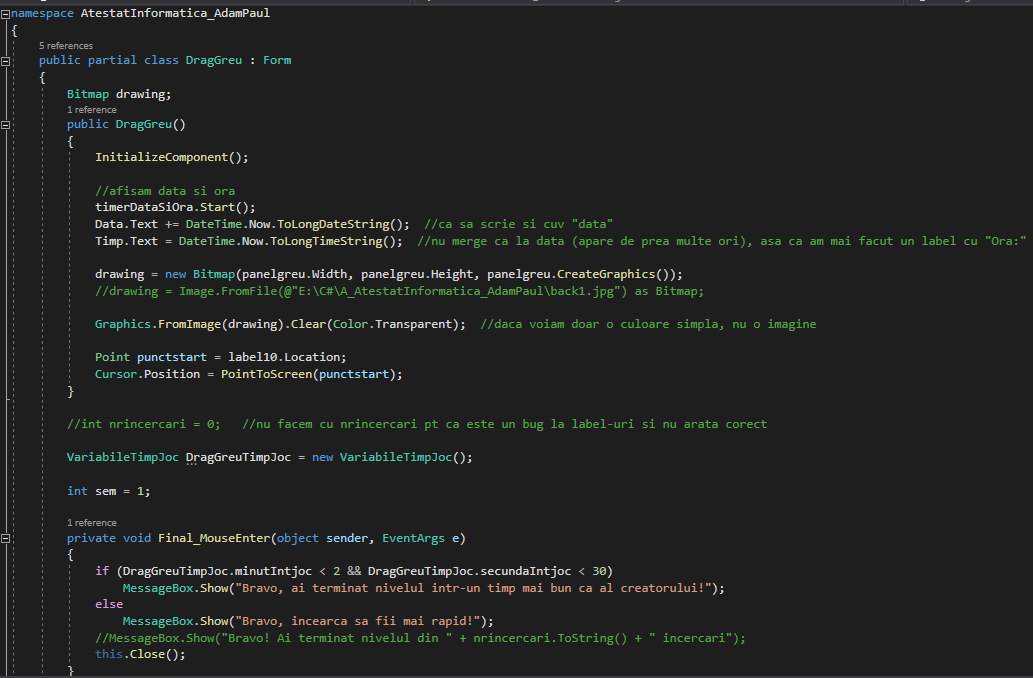
 Subprogramul Panelrandom\_DoubleClick() determină linia și coloana butonului apăsat, îi verifică textul și în funcție de acesta execută comenzile necesare, apoi face butoanele vizibile din nou.

 Primul subprogram preia coordonatele X și Y ale mouse-ului și le pune în două label-uri diferite, iar a doua funcție este cea a butonului “Resetează”.

5.7 Formurile „DragUsor”, „DragMediu”, „DragGreu”

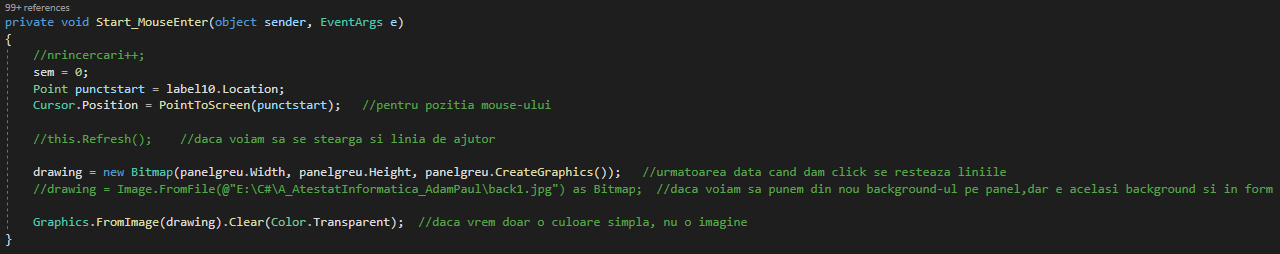
Cele 3 formuri sunt în funcția dificultății: ușor, mediu și greu. Ele conțin un *panel* pe care sunt puse mai multe *label-uri,* iar jucătorul trebuie să ajungă de la Start la Final fără să atingă label-urile, încercând să “bată” timpul creatorului. De asemenea, formurile arată și timpul petrecut în joc, timpul unei ture, iar jucătorul are posibilitatea să marcheze traseul pe unde a trecut, făcând click cu mouse-ul, cu ajutorul unui *bitmap* și a unui *rectangle*.

 Exemplu de cod pentru „DragGreu”: (și celelalte două conțin aproximativ același cod)

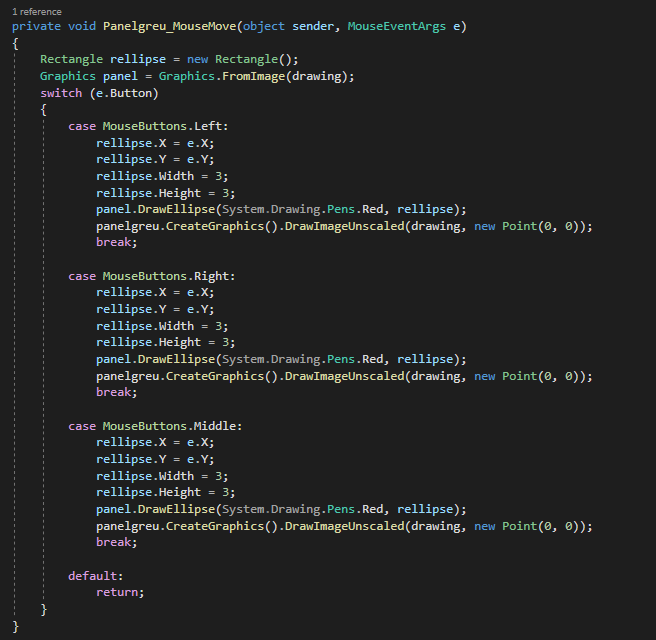


În subprogramul DragGreu() afișăm data și ora, creăm bitmap-ul și poziționăm mouse-ul.

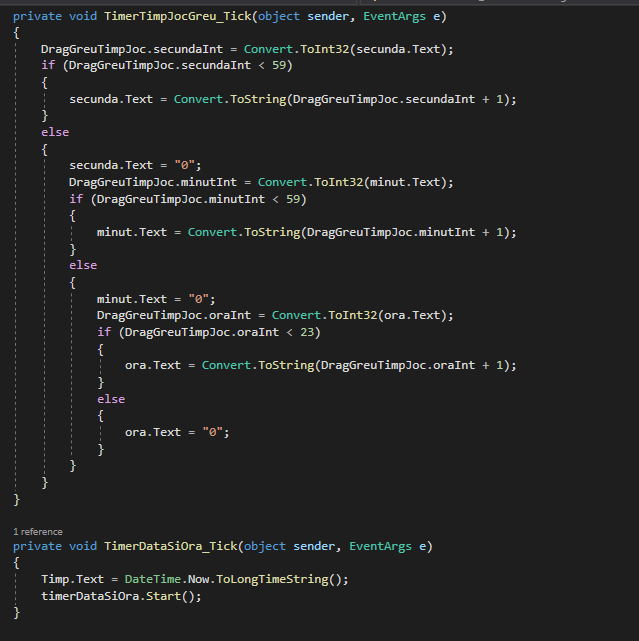
Subprogramul Final\_MouseEnter() este folosit atunci când ajungem la final.

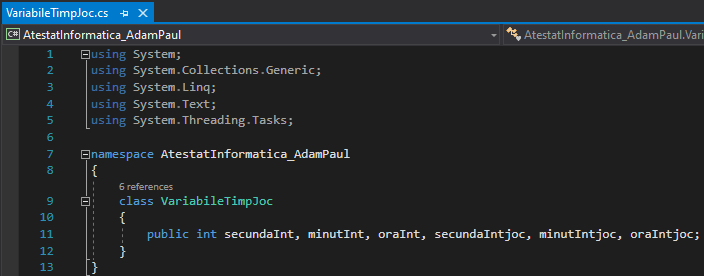


Cu ajutorul acestui subprogram resetăm labirintul atunci când intrăm într-un perete.

 În acest subprogram creăm forma geometrică lăsată atunci când apăsăm click stânga.

Subprogramul de mai sus arată pe ecran de cât timp am început tura și se resetează de fiecare dată când lovim un perete.

 Acest subprogram cronometrează de cât timp suntem în joc.



Pentru a putea refolosi variabilele destinate timpului de joc în mai multe formuri am folosit clasa „VariabileTimpJoc”.

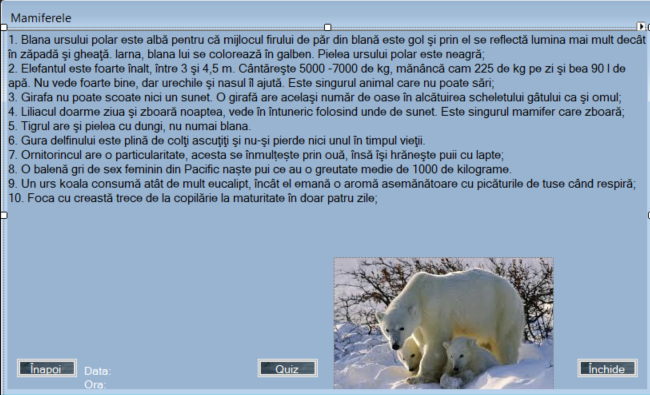
5.8 Formul „Teorie”

 Acest form are rolul de a deschide unul dintre cele 3 capitole destinate învățării: ”Corpul uman”, “Sistemul solar” și “Mamiferele”.

5.9 Formurile „Cap1”, „Cap2”, „Cap3”

Aceste trei formuri conțin informații generale despre temele celor trei capitole, scrise intr-un *richtextbox.* Toate aceste informații vor putea fi verificate cu ajutorul quiz-urilor.

De asemenea, formurile conțin și un buton care ne direcționază direct la quiz.

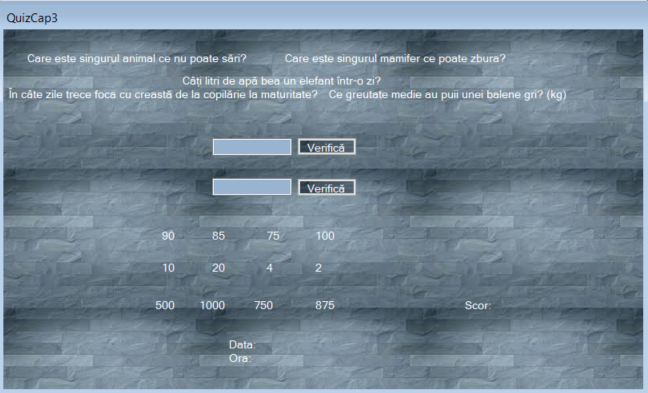
 Exemplu de interfață pentru “Cap3”: (interfața e la fel la toate cele trei formuri)

5.10 Formul „QuizVerificare”

Formul acesta ne direcționează către quiz-urile destinate verificării cunoștiințelor dobândite în urma studierii teoriei.

5.11 Formurile „QuizCap1”, „QuizCap2”, „QuizCap3”

Cu ajutorul acestor formuri verificăm cunoștințele obținute după citirea teoriei. Fiecare quiz conține 5 întrebări de tip grilă sau de completat in căsuțe, iar la sfârșit se va afișa la câte întrebări s-a răspuns corect.

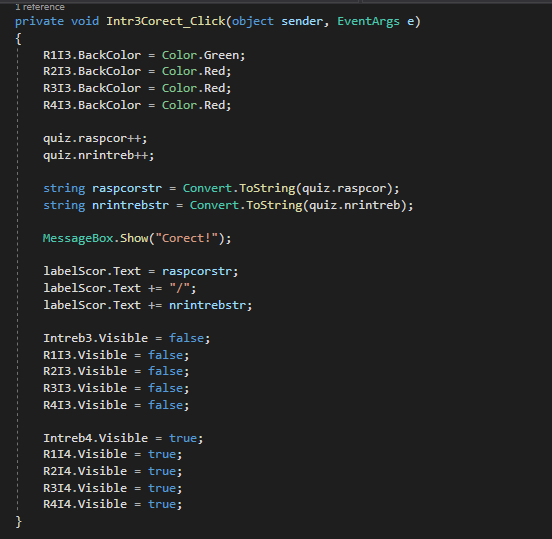
Exemplu interfață și cod pentru „QuizCap1”: (cele trei formuri sunt aproximativ la fel)

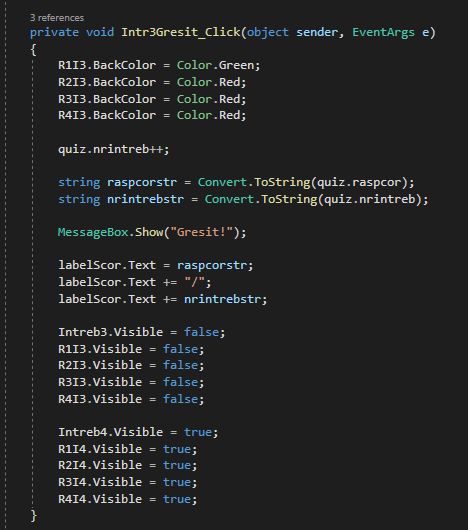
Aceste întrebări și răspunsuri vor fi suprapuse și vor apărea pe rând pe ecran.

 Acest subprogram este destinat întrebărilor de tipul “completat în căsuță”. Codul de mai sus este dependent de mouse și de un buton pentru a verifica răspunsul.

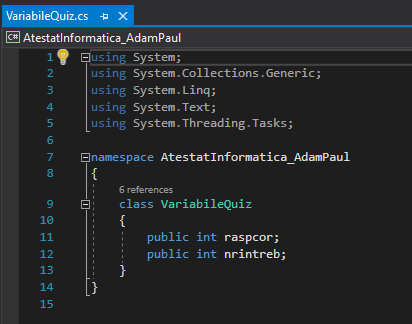
 Dacă răspunsul este corect, numărul întrebărilor corecte se mărește, altfel rămâne același. Apoi, “ștergem” întrebarea de pe ecran și o afișăm pe următoarea.

Acest subprogram este la fel ca cel prezentat mai sus, doar că pentru a verifica răspunsul trebuie să apăsăm tasta „Enter”.

 Subprogramul de mai sus este destinat întrebărilor de tip grilă.

 După darea răspunsului, cel corect se va face verde, iar celelalte roșii. Dacă răspunsul este corect, numărul întrebărilor corecte se mărește, altfel rămâne același. Apoi, “ștergem” întrebarea de pe ecran și o afișăm pe următoarea.

Subprogramul acesta este același cu cel anterior și este folosit atunci când nu răspundem corect la întrebare, deoarece nu mărește numărul întrebărilor răspunse corect.

 Pentru a reține numărul de întrebări corecte și numărul de întrebări totale am folosit clasa „VariabileQuiz”.

1. Webografie
2. [www.stackoverflow.com](http://www.stackoverflow.com)
3. [www.msdn.microsoft.com](http://www.msdn.microsoft.com)
4. [www.howtogeek.com](http://www.howtogeek.com)
5. [www.codeproject.com](http://www.codeproject.com)
6. www.youtube.com