



Universitatea
Ștefan cel Mare
Suceava

Quiz Millionaire

Proiectare Interfete Utilizator

- documentatie -

DOCUMENTATIE INCOMPLETA

Pricob Mihai Alexandru
3124A – Calculatoare

<https://github.com/pricob/QuizMillionaire-FINALE->

CUPRINS

1	Introducere	
2	Conditii pentru rulara aplicatiei	
3	Scopul Aplicatiei	
4	Clase si metode folosite.	
5	Librarii si metode abordate	
.		
6	Bibliografie	
.		
7	Concluzii proiect	
.		
X		

I. Introducere

Quiz Milionaire este un concurs de cultura generala foarte popular, unde participantii isi testeaza cunostintele pentru a castiga un premiu foarte mare in bani.

Aplicatia este o simulare a concursului cu intrebari mult mai usoare disponibila oricui doreste sa isi incerce cunostintele.

A fost construita in **Microsoft Visual Studio** folosind Windows Forms pentru interfata grafica **C#**.

II. Conditii pentru rularea aplicatiei

Aplicatia a fost testata in **Microsoft Visual 2019** si **Microsoft Visual 2020**. In fisierele sursa nu au fost incluse executabile

Este necesara instalarea **.Net Framework** pentru a permite aplicatiei de a fi testata sau construita.

III. Scopul Aplicatiei

Dupa cum a fost prezentata anterior, aplicatia va afisa utilizatorului o serie de intrebari, intr-o ordine aleatorie. Acestea sunt stocate intr-un fisier care se afla in locatia: **".\QuizMilionaire\bin\Debug\"** numit **"Intrebari.txt"**.

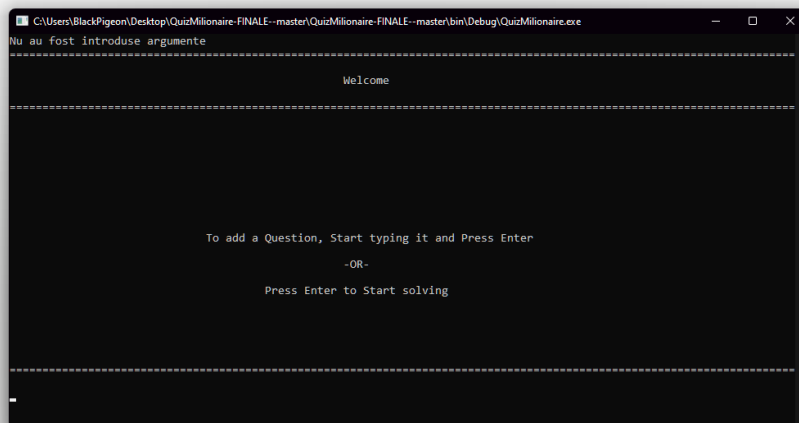
In acest fisier intrebarile sunt stocate sub forma

“intrebare; varianta A; varianta B; varianta C; varianta D; litera corecta”

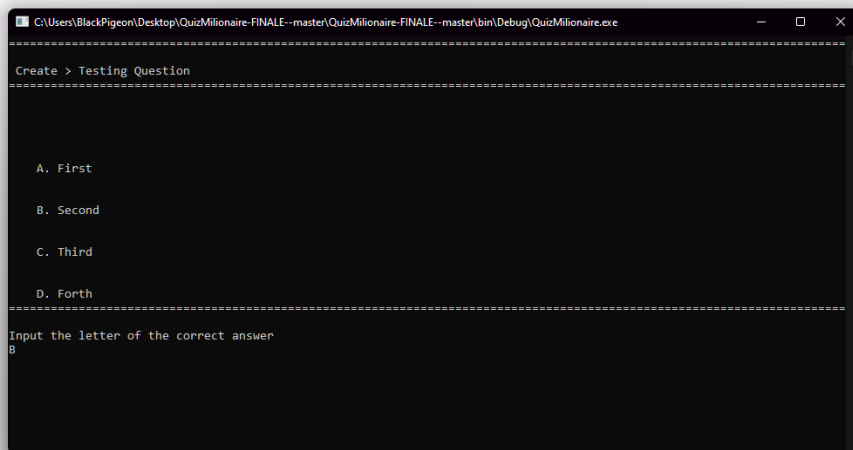
Aplicatia include in functia **main** din **Program.cs** o variabila de tip boolean care permite afisarea unei console de administrare pentru adaugarea intrebarelor noi intr-un format mai usor.

```
8      internal class Program
9      {
10         - references
11         static void Main(string[] args)
12         {
13             bool use_console = true;
14             Application.EnableVisualStyles();
15             Application.SetCompatibleTextRenderingDefault(false);
```

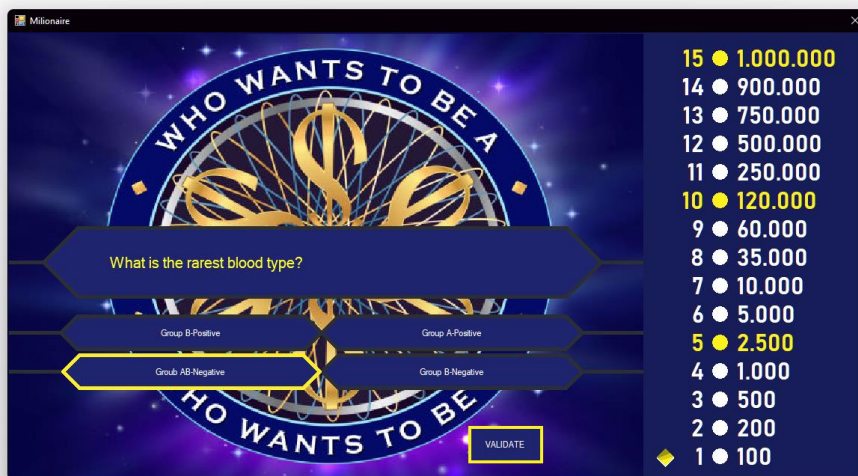
Atunci cand este setata pe **true** programul va deschide consola de administrare.



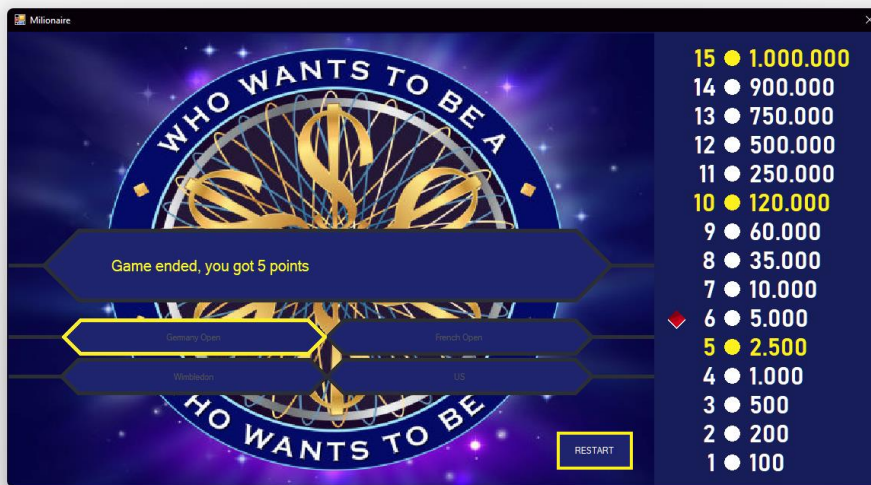
Dupa introducerea datelor, intrebarea, optiunile si raspunsul vor fi salvate in fisier.



Atunci cand variabila este setata pe **false** programul se va deschide fara consola de administrare



Fiecare intrebare raspusa corect, va incremeta scorul utilizatorului de la 0 pana la 15, caz in care utilizatorul a castigat punctaj maxim. Daca o intrebare este gresita pe parcurs, jocul se va opri, si va oferi o optiune de resetare.



IV. Clase si metode folosite

In acest proiect sunt prezente 3 clase, clasa **Program**, **Question** si **Menu**.

Clasa Program

In clasa Program, avem rulara windows forms Milionaire.cs, administrarea parametrilor introdusi, si administrarea consolei pentru adaugarea intrebarilor.

Clasa Question

In clasa Question, exista un constructor public prin care se vor crea obiecte de tip 'Question' atribuind datele primite ca parametru, si un ID care va incrementa automat.

```

22 // Constructor
23 3 references
24 public Question(
25     string _question, string _answerA, string _answerB,
26     string _answerC, string _answerD, string _solution)
27 {
28     question = _question;
29     answerA = _answerA;
30     answerB = _answerB;
31     answerC = _answerC;
32     answerD = _answerD;
33     solution = _solution;
34     ID = ID++;
35 }
36

```

O metoda de initializare pentru citirea datelor din fisier

```

38 // Initialization - method to read the questions from file
39 2 references
40 public List<Question> Init()
41 {
42     List<Question> questions = new List<Question>();
43
44     string fileName = ConfigurationManager.AppSettings.Get("FileName");
45     //Stream streamFisierText = File.Open(fileName, FileMode.Create);
46     Stream streamFisierText = File.Open(fileName, FileMode.Open);
47     using (StreamReader reader = new StreamReader(streamFisierText))
48     {
49         string fileLine;
50         while ((fileLine = reader.ReadLine()) != null)
51         {
52             string[] temp = fileLine.Split(';');
53             questions.Add(new Question(temp[0], temp[1], temp[2], temp[3], temp[4], temp[5]));
54             counter++;
55         }
56     }
57     return questions;
58 }

```

Si cateva metode pentru a obtine variantele si raspunsul in functie de parametrul ID.

```

60 // Obtain the question on the current list index
61 // 2 references
62 public string GetQuestion()...
65
66 // Obtain the first option
67 // 2 references
68 public string GetA()...
71
72 // Obtain the second option
73 // 2 references
74 public string GetB()...
77
78 // Obtain the third option
79 // 2 references
80 public string GetC()...
83
84 // Obtain the forth option
85 // 2 references
86 public string GetD()...
89
90 // Otain the correct nswer
91 // 3 references
92 public string GetSolution()...
95
96 // 1 reference
97 public static int GetCounter()...

```

Clasa Menu

In clasa Menu, sunt 4 metode, Init() se ocupa cu initializarea la inceputul rularii programului, daca utilizatorul apasa tasta ENTER, programul va rula in continuare sub forma de terminal fara interfata grafica. Daca utilizatorul incepe sa scrie, va fi luata ca intrebare, se va cere introducerea optiunilor si selectarea unei optiuni adevarate. La final programul va adauga setul de date in fisier folosind metoda **AddQuestion(string input)**, si se va intoarce.

INCOMPLET