UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" IAŞI, FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE, SPECIALIZAREA CALCULATOARE ŞI TEHNOLOGIA INFORMAŢIEI

Ingineria programării - proiect
Joc întrebări de cultură generală

Coordonator:

Prof. Corina Cîmpanu

Studenti:

Boldureanu G. Gelu-Florin, grupa 1307B

Boțic F.G. Teodora, grupa 1306A

Cănuci M.C Răzvan-Andrei, grupa 1307B

Vecliuc Draia-Teodora, grupa 1307B

1. Introduction

1.1 Scopul

Smarty Pants este un joc de cultură generală care poate fi jucat de orice persoană dornică să-și testeze cunoștințele acumulate în diferite domenii precum: biologie, fizică, geografie sau istorie. Jocul se poate juca doar de o singură persoană sau mai multe dacă se dorește o întrecere între toate persoanele prezente în fața unui calculator, iar jocul să fie câștigat de cel cu punctajul final cel mai mare. Pentru mai multe detaliile și specificații aupra funcționalității jocului sunt definite în secțiunile 3 și 4. O prezentare generală a jocului este date în secțiunea 2, iar o listă de cerințe este dată în secțiunea 5.

1.2 Convențiile documentului

Acest document urmează standardul formatului IEEE pentru dezvoltare de software. Standardul IEEE alcătuit din mai multe cerințe, o cerință este o condiție sau capacitatea de care are nevoie un utilizator pentru a rezolva o problemă sau a atinge un obiectiv. Scrierea acestui document se va realiza la persoana a III-a singular, va fi actualizat mereu la fiecare modificare a aplicației, va fi organizat pe secțiuni cu folosirea de enumerări verticale, cu linii sau puncte, tabele, diagrame etc.

1.3 Audiență vizată și sugestii de lectură

Acest document nu este destinat jucătorului, utilizatorului pentru care s-a creat aplicația, deoarece este prezentat pas cu pas modul în care software-ul a fost creat și a fost implementat. Un jucător nu are nevoie de asemenea informații suplimentare, el are nevoie de instrucțiunile care trebuie aplicate pentru a îi permite să folosească jocul. Din acest motiv, acest document este orientat către testării și persoanele care au dezvoltat jocul. Documentul începe cu o prezentare generală a funcțiilor și specificațiilor pentru acest joc în secțiunea 2, apoi trece la descrierea cerințelor pentru comunicarea cu partea de hardware și software extern în secțiune 3. Secțiunea 4 descrie funcțiile jocului în detaliu, iar secțiunea 5 enumeră diverse cerințe ale jocului care trebuie să fie respectate după finalizare. Ca o sugestie de lectură, toate categoriile de public ale acestui document să înceapă cu secțiunea 2 mai întâi pentru a se prezenta o idee generală despre cerințele software. Testerii ar trebui să citească în continuare secțiunile 5.1 până la 5.4 (performanță, siguranță, cerințe de securitate și atribute de calitate software). Scopul este de a ne face o idee despre cum jocul îi va afecta pe ei și sistemul pe care îl rulează, precum și aspirațiile de calitate. Un tester ar trebui să citească secțiunea 3.1 (interfețe cu utilizatorul) urmată de toată secțiunea 4 (funcțiile sistemului). Citind documentul în această ordine îi va oferi testatorului o idee despre ce să se aștepte în interfață la prima vedere și atunci ei pot testa toate funcțiile individuale pentru a se asigura că respectă specificațiile. După ce au citit secțiunea 2, dezvoltatorii de jocuri ar trebui să citească secțiunile rămase în ordine, deoarece aceasta documentul a fost conceput special în scopul dezvoltării jocului. Dezvoltatorul trebuie să obțină o idee generală în secțiunea 2 a jocului. Apoi, cum trebuie să se interfațeze cu orice altceva din secțiunea 3 (deci au o idee despre ce instrumente să folosească și, eventual, cum ar trebui să le folosească). Secțiunea 4 este cea mai mare important

pentru un dezvoltator, deoarece descrie toate funcțiile jocului în detaliu și va ajuta cu luarea deciziilor în scrierea codului real pentru joc. Secțiunea 5 este considerată cea mai puțin importantă, dar dezvoltatorul ar trebui să-l citească în continuare pentru a se asigura că jocul lor a aderat la idealurile date.

1.4 Domeniul de aplicare al produsului

Smarty Pants este un joc de cultură generală care poate fi jucat de orice persoană dornică să-și testeze cunoștințele acumulate în diferite domenii, jocul este dezvoltat în C#. Obiectivul acestui joc ca o persoană să-și testeze cunoștiințele de cultură generală prin a răspunde la întrebări din diferite domenii. Fiecare persoană trebuie să răspundă la 20 de întrebări dintr-o categorie sau mai mule în funcție de câte categorii a ales în pagina de start a jocului. Jocul este conceput pentru a fi pe cât de ușor de utilizat, pe atât de distractiv pentru jucător. Este destinat să ruleze pe orice computer care poate folosi C# și poate fi rulat cu ușurință de un computer cu putere medie de procesare.

2. Descriere generală

2.1 Perspectiva produsului

SmartyPants este un joc de cultură generală care poate fi jucat de orice persoană dornică să-și testeze cunoștințele acumulate sau să-și împropăteze anumite noțiuni din diferite domenii precum: biologie, fizică, geografie sau istorie. Acesta este un joc captivant, care antrenează fiecare neuron dezvoltând noi circumvoluțiuni. El stimulează și antrenează memoria și flexibilitatea gândirii prin pendularea întrebărilor din diferite categorii care au fost acumulate de-a lungul întregii vieți.

De acest joc se poate bucura individul în solitudinea proprie, sau pentru cei competitivi își pot invita prietenii, colegii, rudele pentru " a se duela" în cunoștințele acumulate. Cel câștigator va simți o satisfacție emoțională deosebită, irenică și încredere în sine va fi sporită, iar dacă nu va câștiga, va învăța să piardă cu grație. Chiar dacă nu este un joc de echipă, acesta înlesnește coeziunea si conexiunea umană. De asemenea, este o manieră neinvazivă și agreabilă de a evalua propriile cunoștiințe, dar și de asimilare și reținere de noi informații.

Pe de o parte, toții oameni își doresc să fie tot mai informați și updatați la evenimentele din jurul lor, să-și formeze o părere proprie, pentru a nu fi duși în eroare, din "nebunia de a gândi cu mintea lor",GL. Pe de altă parte, cunoștințele acumulate de mai mult timp, prăfuite râmân tot mai adâncite în Groapa Marianelor. Persoana care se joacă își antreanează atât memoria de scurtă durată cât și cea de lungă durată, dezvoltându-si abilitatea de a trece spontan de la marile războaie mondiale la formula vitezei luminii în vid, și apoi poate, la noțiuni anatomie și fiziologie umană. Totodată, capacitatea de concentrare și focusare a jucătorului este solicitată de-a lungul, fiecărei întrebări și a întregului joc.

Concentrare, focus, logică, flexibilitate mintală sunt însușirile pe care consolidează și intensifică acest joc prin necesitatea actualizării unor noțiuni, precum și prin atenția sporită pe

care o cere pe toată durata de timp, fac ca acest joc să fie o ocazie de relaxare deosebită împletită perfect cu îmbunătățirea abilitaților individuale. Astfel, acesta este mai mult decât o opțiune efemeră de a petrece timpul, este șansa unei dezvoltării personale, o gimnastică a creierului ce ce solicita anumite procese congnitive.

2.2 Funcțiile produsului

Funcțiile majore pe care SmartyPants trebuie să le îndeplinească pentru utilizatorul său final sunt următoarele:

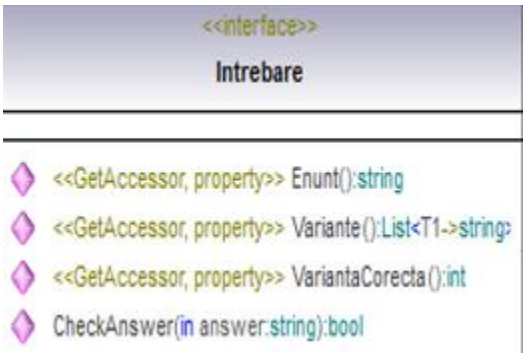
- Să permită alegerea unei categorii și generarea de întrebări pentru respectiva categorie
- Actualizarea puntajului intermediar în funcție de câte întrebări au fost generate și de câte răspunsuri corecte a oferit utilizatorul
- Afișarea puntajului final și obțiunea de a reîncepe sau de a închide jocul

2.3 Clasele și caracteristicile utilizatorilor

SmartyPants ar trebui să fie conceput pentru a fi ușor de utilizat, astfel încât orice clasă de utilizatori ar trebui să fie capabil să se așeze și să se joace. Interfețele și opțiunile ar trebui să fie simple, descriptive și ușor de navigat. Simplitatea setului de caracteristici SmartyPants va permite oricărui utilizator să joace jocul, dar ar trebui să ofere a sistem de joc semnificativ provocator pentru jucătorii avansați.

Clasele utilizate în implementare:

- Interfata Intrebare

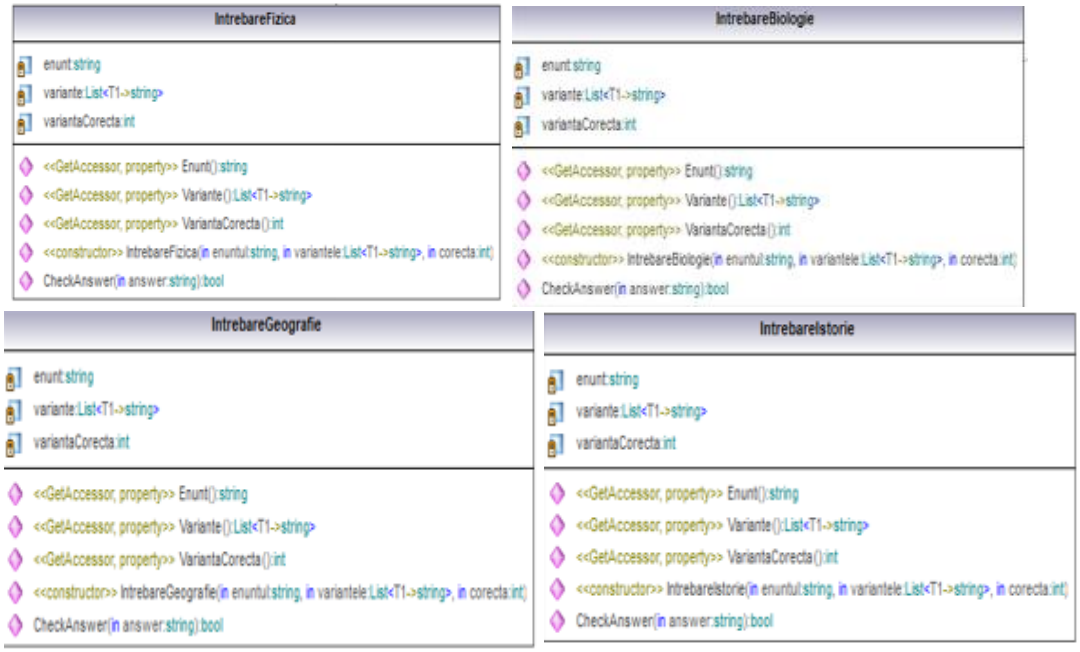


Această interfață a fost creată deoarece reprezintă punctul de plecare pentru fiecare categorie de întrebare. Orice întrebare, indiferent din ce domeniu provine, este alcătuită dintr-un enunț, conține câteva variante de răspuns și varianta de răspuns corectă pentru a verifica dacă utilizatorul a răspuns corect la întrebare

- Enunt() reprezintă o metodă de get care permite returnarea unui string ce conține enunțul întrebării
- Variante() reprezintă o metodă de get care permite returnarea unei liste de stringuri ce conține variante de răspuns a întrebării, pe care utilizatorul le vede în interfață și din care trebuie să selecteze un răspuns pe care îl consideră corect

- VariantaCorecta() reprezintă o metodă de get care permite returnarea unui int ce conține varianta de răspuns corectă a întrebării. Răspunsul corect este de tip int deoarece este stocat indexul din lista de stringuri de la Variante() unde răspunsul este corect.
- CheckAnswer() este o metodă care returnează true sau false dacă răspunsul pe care l-a ales jucătorul este corect sau nu. Această metoda se folosește de metodele Variante() și VariantaCorecta() pentru a putea realiza acest lucru.

- Clasele IntrebareBiologie, IntrebareFizica, IntrebareGeografie, IntrebareIstorie



Toate aceste 4 clase implementează interfața Intrebare și au rolul de a creea întrebări specifice pentru categoria din care aparțin, de exemplu pentru IntrebareIstorie

“

În ce an a reușit Mihai Viteazul unirea celor trei mari țări medievale?

a) 1590
b) 1600
c) 1790
d) 1950

“

- în variabila enunt va fi stocat informația “ În ce an a reușit Mihai Viteazul unirea celor trei mari țări medievale? ”, aceasta va avea ca specificator de accces private, din acest motiv are nevoie de o metoda de tip get publică pentru a putea fi accesată și din exterior

- în variabila variante va fi stocat informația [“1590”, “1600”, “1790”, “1950”], aceasta va avea ca specificator de accces private, din acest motiv are nevoie de o metoda de tip get publică pentru a putea fi accesată și din exterior
- în variabila variantaCorecta va fi stocat informația 1, indexul din lista de variante de răspuns unde răspunsul este corect , aceasta variabila va avea ca specificator de accces private, din acest motiv are nevoie de o metoda de tip get publică pentru a putea fi accesată și din exterior

Constructorul pentru fiecare clasa inițializează câmpurile enunt, variante și variantaCorecta cu valorile corespunzătoare date ca parametrii în funcție.

- Clasa CautInfo

Această clasă a fost creată deoarece este folosită pentru a prelua toate informațiile necesare din fișiere corespunzătoare pentru fiecare categorie.

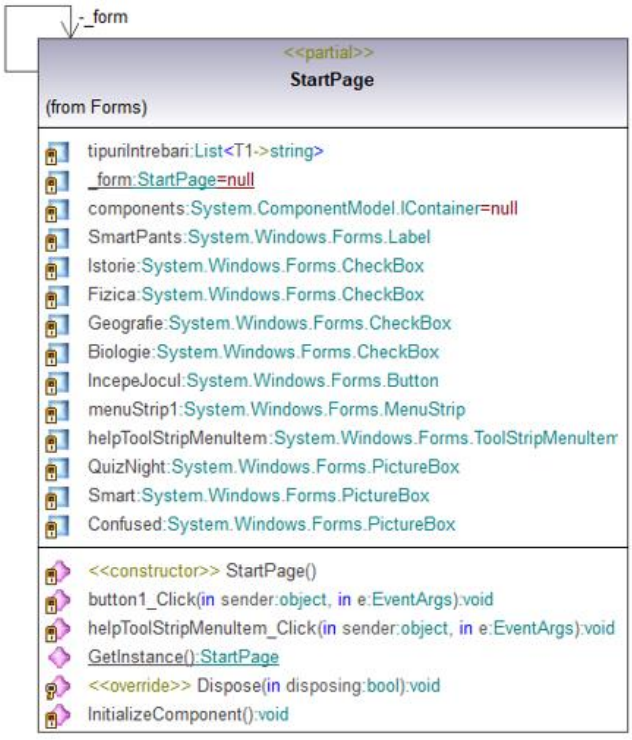
- Variabila membra a acestei clase este path, care este de tip string și reține path-ul către folderul unde sunt toate fișierele informațiile necesare pentru intrebare
- Metoda Enunturi() primește ca parametru numele fișierului din care trebuie să citească toate enunțurile pentru o anumită întrebare, concatenând path-ul de la folder cu numele fișierului și creând path-ul complet pentru a putea realiza acest lucru.
- Metoda Variante() primește ca parametru numele fișierului din care trebuie să citească toate variantele de răspuns pentru o anumită întrebare, concatenând path-ul de la folder cu numele fișierului și creând path-ul complet pentru a putea realiza acest lucru.
- Metoda RaspunsCorect() primește ca parametru numele fișierului din care trebuie să citească toate răspunsurile corecte pentru o anumită întrebare, concatenând path-ul de la folder cu numele fișierului și creând path-ul complet pentru a putea realiza acest lucru.

- Clasele IntrebariIstorie, IntrebariFizica, IntrebariGeografie, IntrebariBiologie

Aceste 4 Clase folosesc clasa CautInfo, fiecare clasa este alcătuită din 3 variabile, care vor fi inițializate corespunzător cu valorile returnate metodele din clasa CautInfo. De exemplu, pentru clasa IntrebariIstorie

- Variabila enunturi va fi inițializată cu o listă de enunțuri pentru Istorie, apelând metoda Enunturi() din clasa CautInfo dând ca paramtru "istorie.txt"
- Variabila variante va fi inițializată cu o listă de liste de variante de răspuns pentru Istorie, apelând metoda Variante() din clasa CautInfo dând ca paramtru "variante_istorie.txt"
- Variabila enunturi va fi inițializată cu o listă de răspunsuri corecte pentru Istorie, apelând metoda RaspunsCorect () din clasa CautInfo dând ca paramtru "raspunsuri_istorie.txt"

Prima fereastră, pagina de start, a fost implementat sablonul creațional Singleton. Metoda utilizată pentru implementarea șablonului este GetInstance().



2.4 Sistemul de operare

Deoarece SmartyPants este dezvoltat în mediul de rulare C#, este foarte compatibil între platforme. Utilizatori pe un PC, Mac sau un sistem de operare UNIX nu ar trebui să aibă probleme la rularea software-ului.

2.5 Constrângeri de proiectare și implementare

Deoarece SmartyPants este proiectat și implementat într-un singur semestru ca proiect pentru disciplana Ingineria Programării, UNIVERSITATEA TEHNICĂ "GHEORGHE ASACHI" IAŞI, este posibil ca timpul să fie cel mai limitativ factor în acest sens ciclu de dezvoltare. Această constrângere de timp poate necesita ca echipa să reducă proiectul față de momentul prezentării proiectului.

2.6 Documentația utilizatorului

SmartyPants va conține toată documentația utilizatorului intr-un fișier de tip chm, care este integrat în butonul de Help din pagina de start a jocului, in acest fișier sunt prezentate in mai multe pagini funcționalitatea jocului dar și pași pe care utlizatorul trebuie să-i urmeze pentru a se bucura de joc.

3. Cerințele pentru interfață

Această secțiune prezintă o descriere detaliată a datelor de input și output ale aplicației. De asemenea, se prezintă detalii hardware, software și funcționalitățile de bază ale interfeței.

3.1 Interfața cu utilizatorul

În momentul deschiderii aplicației utilizatorul va vedea o fereastră de început precum cea din Figura 1. După selectarea măcar unei categorii din care utilizatorul vrea să primească întrebări se va deschide a doua fereastră, o imagine este cea din Figura 2. Dupa ce utilizatorul va termina să răspundă la cele 20 de întrebări se va deschide o nouă fereastră cu puntajul obținut precum cea din Figura 3.

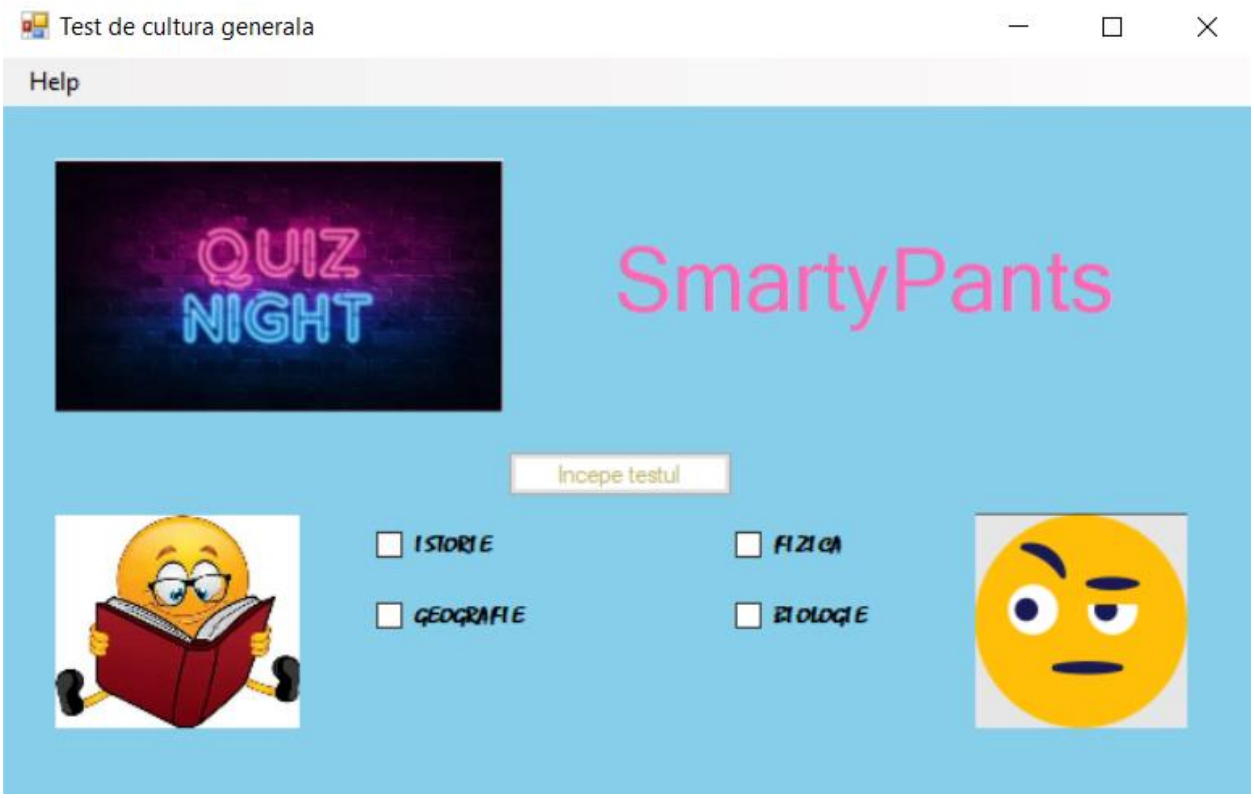


Figura 1

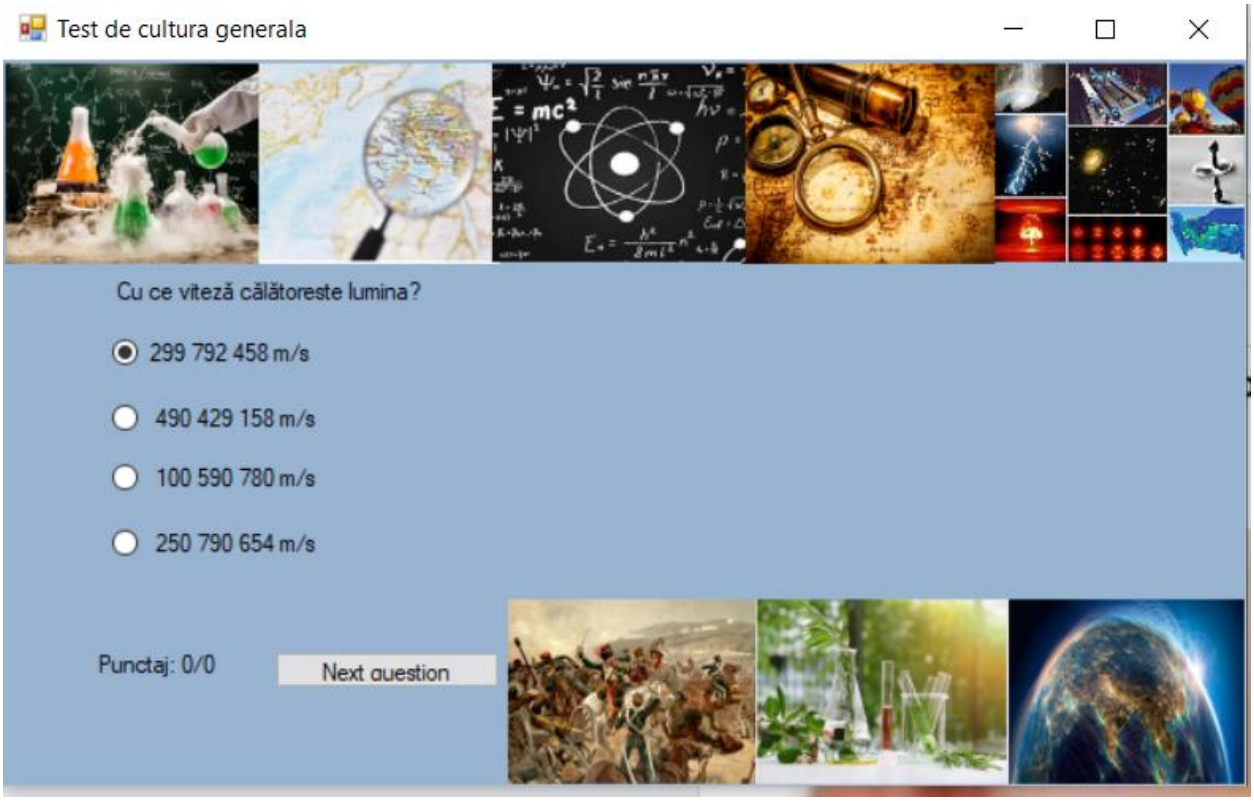


Figura 2

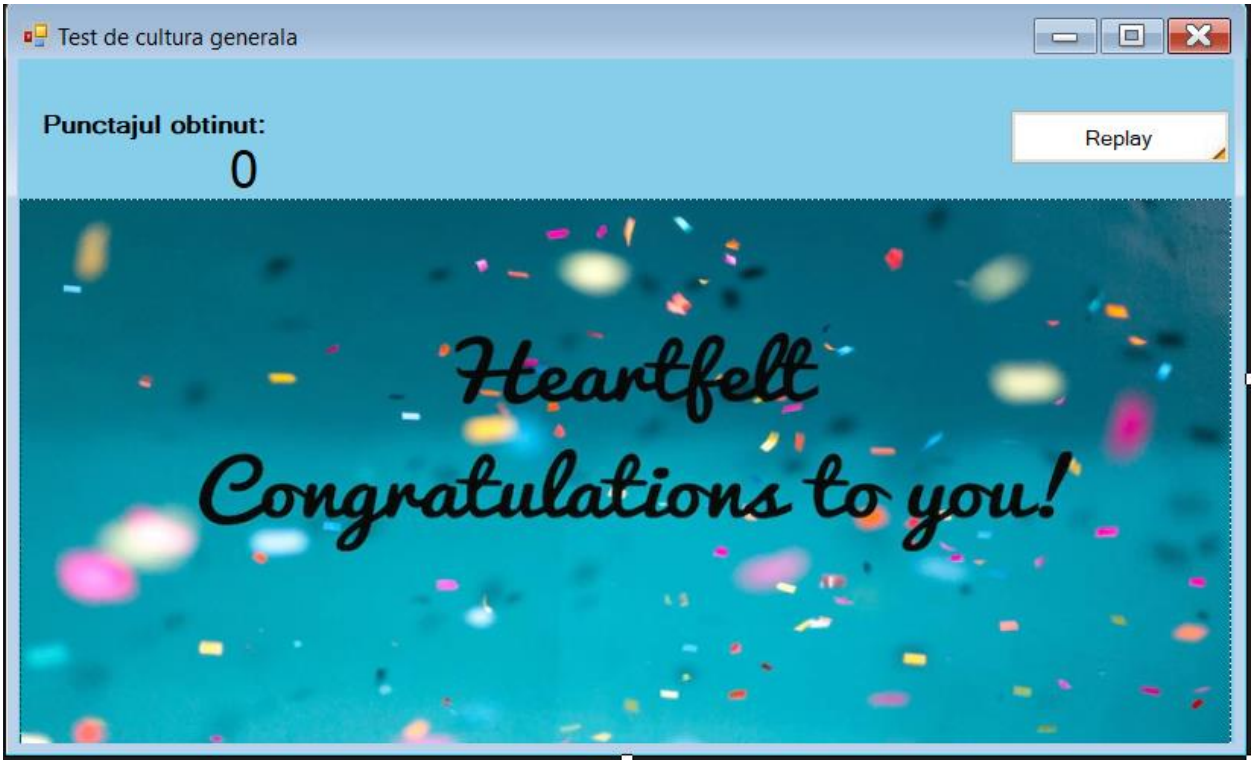


Figura 3

3.2 Interfața hardware

Din moment ce aplicația nu folosește vreo componentă hardware specifică, aceasta nu are o interfață hardware directă. Fișierele in care sunt întrebările și interfața cu utilizatorul rulează pe sistemul de operare pe care sunt instalate.

3.3 Interfața software

Comunicația dintre fișierele cu întrebări și aplicație are se realizează abia după ce utilizatorul selectează măcar o categorie din care vrea să primescă întrebări.

3.4 Interfața de comunicatie

Pentru funcționarea aplicației va fi nevoie ca utilizatorul să dețină fișierele cu întrebările pentru fiecare categorie.

3.5 Cerințe funcționale

Această secțiune include cerintele care specifică toate acțiunile din cadrul aplicației.

3.5.1 Clasa de utlizatori – Jucătorul

3.5.1.1 Cerința funcțională 1.1

Titlu: Selectarea unei categorii și generarea de intrebări

Descriere: După ce utilizatorul va deschide aplicația va trebui sa selecteze o categorie din care vrea să primească întrebări. Urmând ca după ce utilizatorul își spune preferințele să se afișeze următoarea fereastră cu prima intrebare din cele 20 la care trebuie să răspundă jucătorul.

3.5.1.2 Cerința funcțională 1.2

Titlu: Actualizarea punctajului intermediar

Descriere: In timp ce jucătorul se bucură de joc, acesta va putea vedea la care întrebare a ajuns și din intrebările la care a răspuns la câte dintre ele răspunsul era corect sau nu.

3.5.1.3 Cerința funcțională 1.3

Titlu:Afișare punctaj și reînceperea jocului

Descriere: După ce jucătorul a terminat de răspuns la toate cele 20 de întrebări o altă fereastră îi va afișa puntajul final obținut și posibilitatea de a reveni la pagina de start și de a reîncepe jocul.

4. Caracteristicele sistemului

Caracteristicile sistemului nostru sunt acoperite în profunzime în anexe separate pentru scenarii și o diagramă de clasă.

5. Alte cerințe nefuncționale

5.1 Cerințe de performanță

Jocul trebuie să poată rula pe minim Windows 2000 și Linux. Se recomanda deținerea a cel puțin 256 Mb RAM, deși se ar fi mai bine 512 Mb. Placa grafică ar trebui să aibă cel puțin 64 Mb memorie de bord.

5.2 Cerințe de siguranță

Ca și în cazul oricărui joc pe computer, există riscul de epilepsie. Dacă deținești o boala precum epilepsie, discutați cu medicul dumneavoastră înainte de a juca acest joc.

5.3 Cerințe de securitate

Acest joc nu va aduna nicio informație privată de la oameni.

5.4 Atribute de calitate software

Acest software trebuie să fie robust și cât mai lipsit de erori posibil pentru a se asigura că jucătorii au o experiență pozitivă. Jocul ar trebui să fie ușor de preluat și de început pentru un începător, cu o curbă minimă de învățare. Acesta ar trebui să fie suficient de flexibil pentru a permite crearea ușoară a conținutului suplimentar, păstrând în același timp ușurința de-al folosi utilizatorului. În cele din urmă, și cel mai puțin important, jocul ar trebui să fie portabil la sisteme specificat în secțiunea 5.1.

5.5 Reguli de afaceri

Politica echipei de dezvoltare este să respecte toate codurile de conduită stabilite de Universitate.

Diagrama cazurilor de utilizare

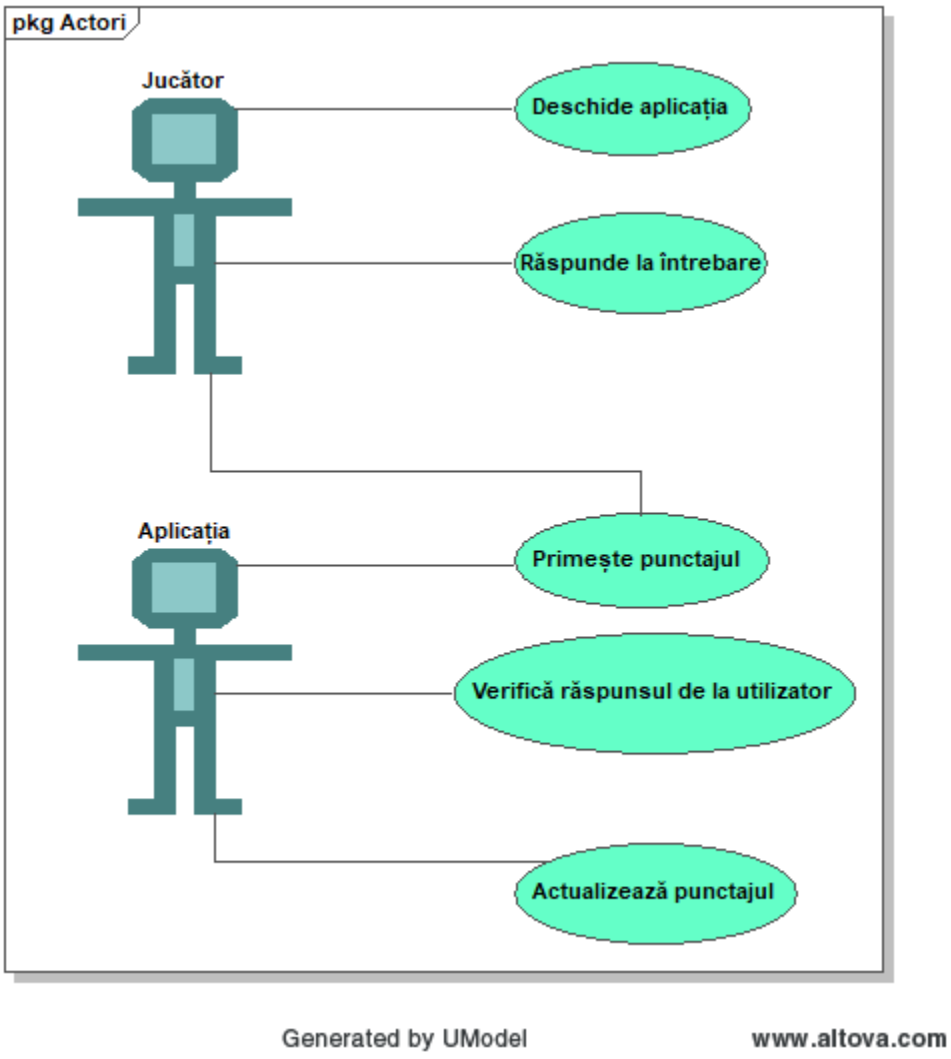
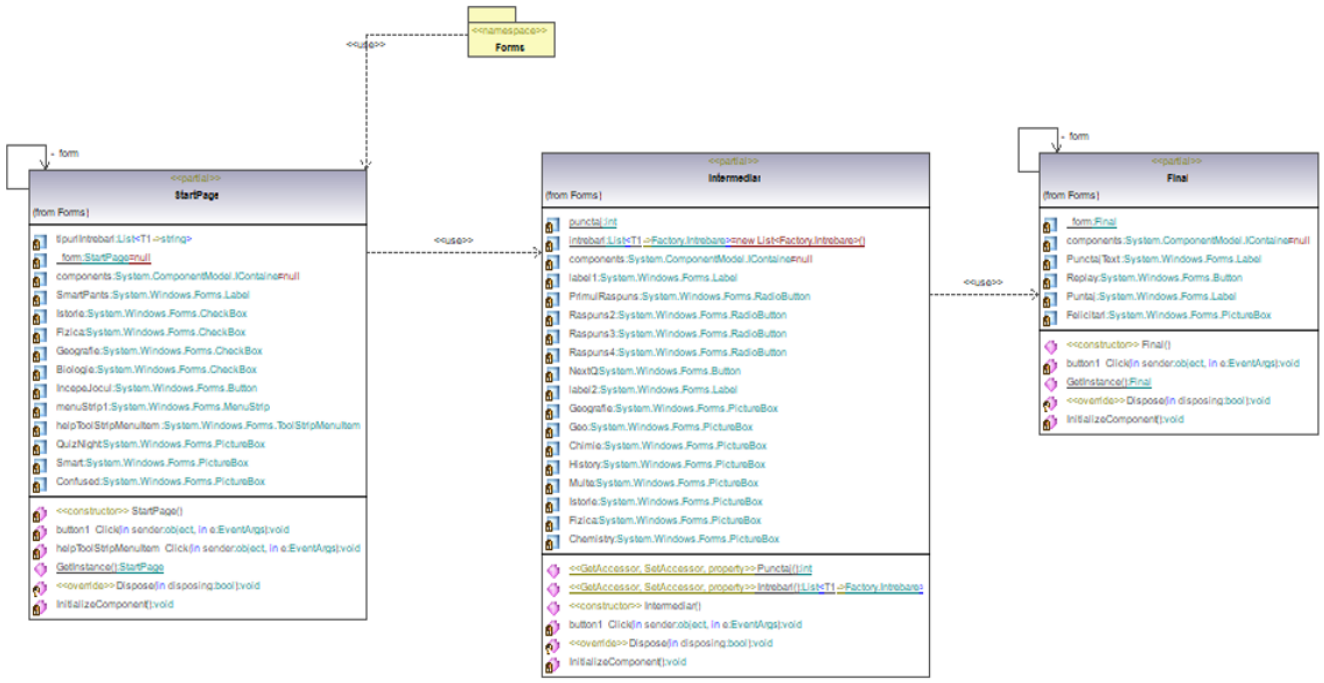
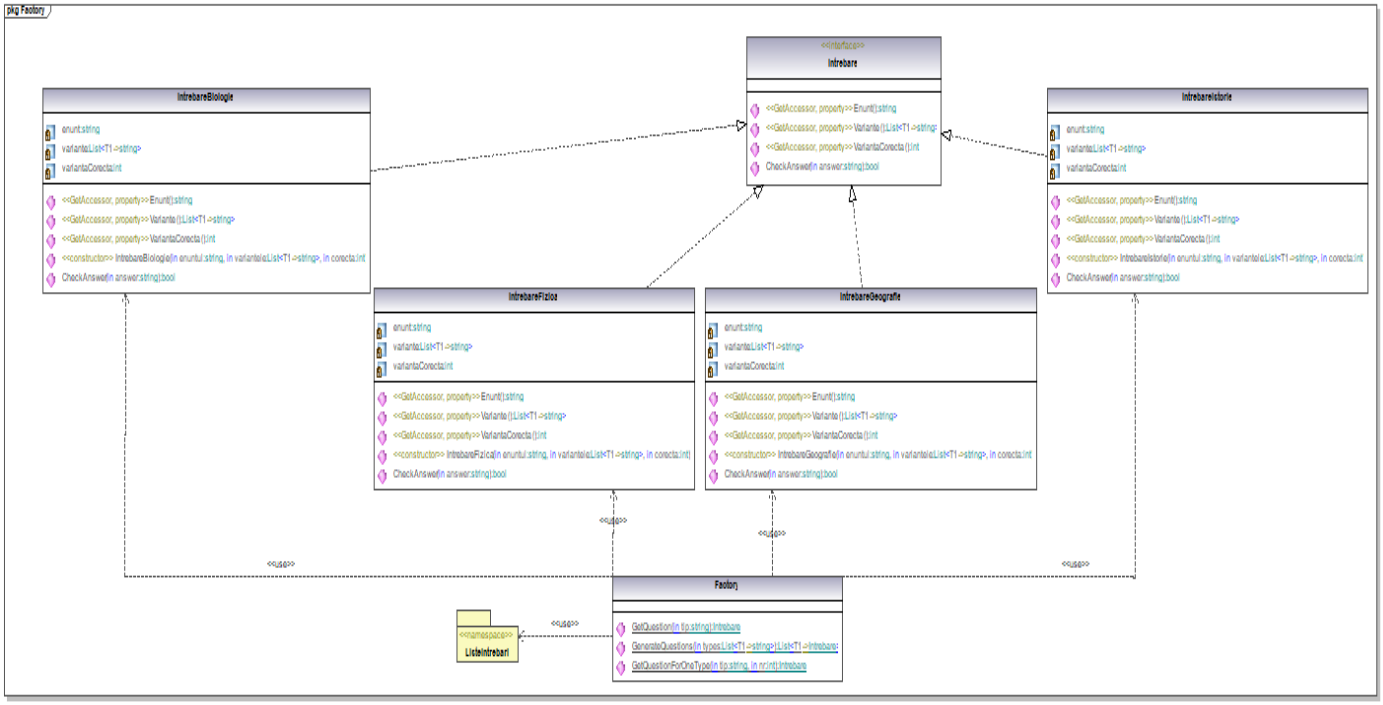
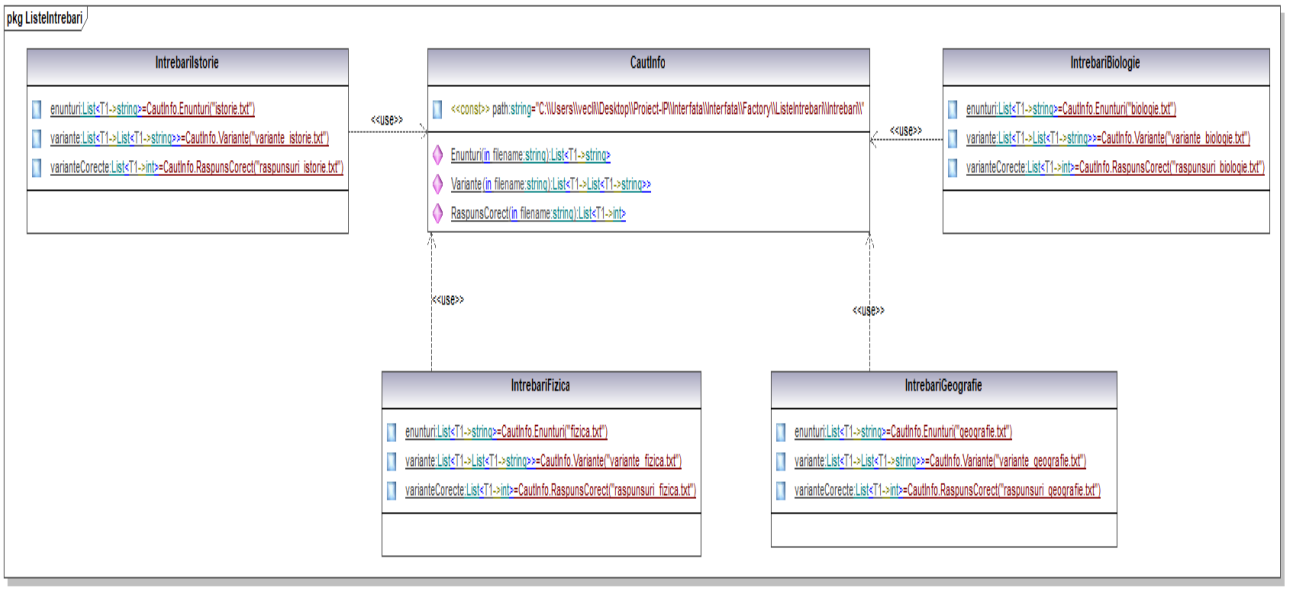


Diagrama de clase





Generated by UModel www.altova.com



Cele 20 de cazuri de testare

Nr.caz test	Metoda testata	Lista de param	Rezultat asteptat	Rezultat obtinut	Stare testului
1	FactoryGetMethodTestGeografie	Null	True	True	Passed
2	FactoryGetMethodTestIstorie	Null	True	True	Passed
3	FactoryGetMethodTestFizica	Null	True	True	Passed
4	FactoryGetMethodTestBiologie	Null	True	True	Passed
5	FactoryGenerateQuestionsTestI	Null	20	20	Passed
6	FactoryGenerateQuestionsTestG	Null	20	20	Passed
7	FactoryGenerateQuestionsTestB	null	20	20	Passed
8	FactoryGenerateQuestionsTestF	Null	20	20	Passed
9	FactoryGenerateQuestionsTestIG	Null	20	20	Passed
10	FactoryGenerateQuestionsTestIF	Null	20	20	Passed
11	FactoryGenerateQuestionsTestIB	Null	20	20	Passed
12	FactoryGenerateQuestionsTestGF	Null	20	20	Passed
13	FactoryGenerateQuestionsTestGB	Null	20	20	Passed
14	FactoryGenerateQuestionsTestFB	Null	20	20	Passed
15	FactoryGenerateQuestionsTestFBI	Null	20	20	Passed
16	FactoryGenerateQuestionsTestFBG	Null	20	20	Passed
17	FactoryGenerateQuestionsTestBIG	Null	20	20	Passed
18	FactoryGenerateQuestionsTestFIG	Null	20	20	Passed
19	FactoryGenerateQuestionsTestFIGB	Null	20	20	Passed
20	FactoryGenerateQuestionsTestIFGB	Null	20	20	Passed

Diagrama de activități

Diagrama de activități cu decizii

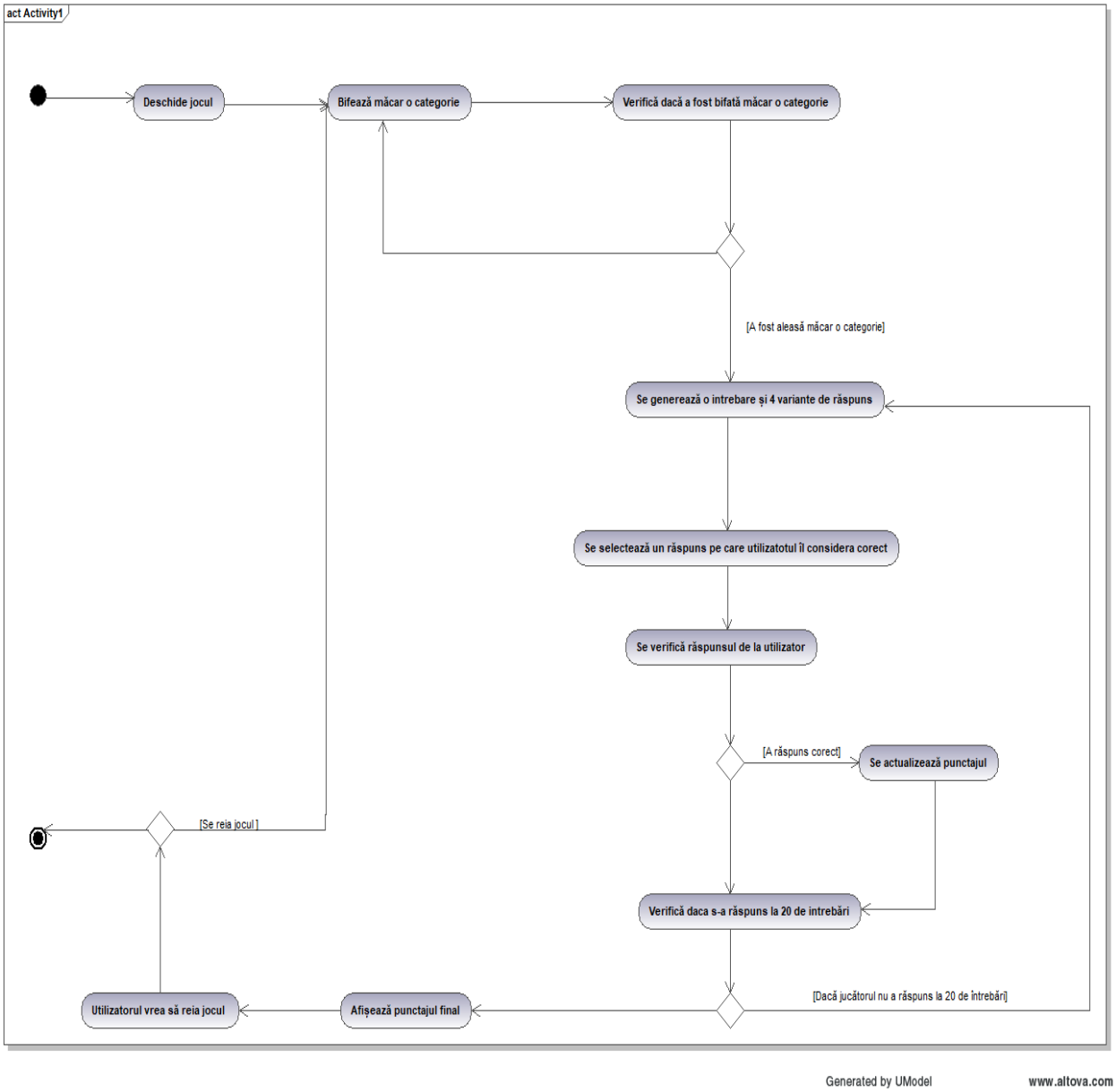


Diagrama de activități cu partiții:

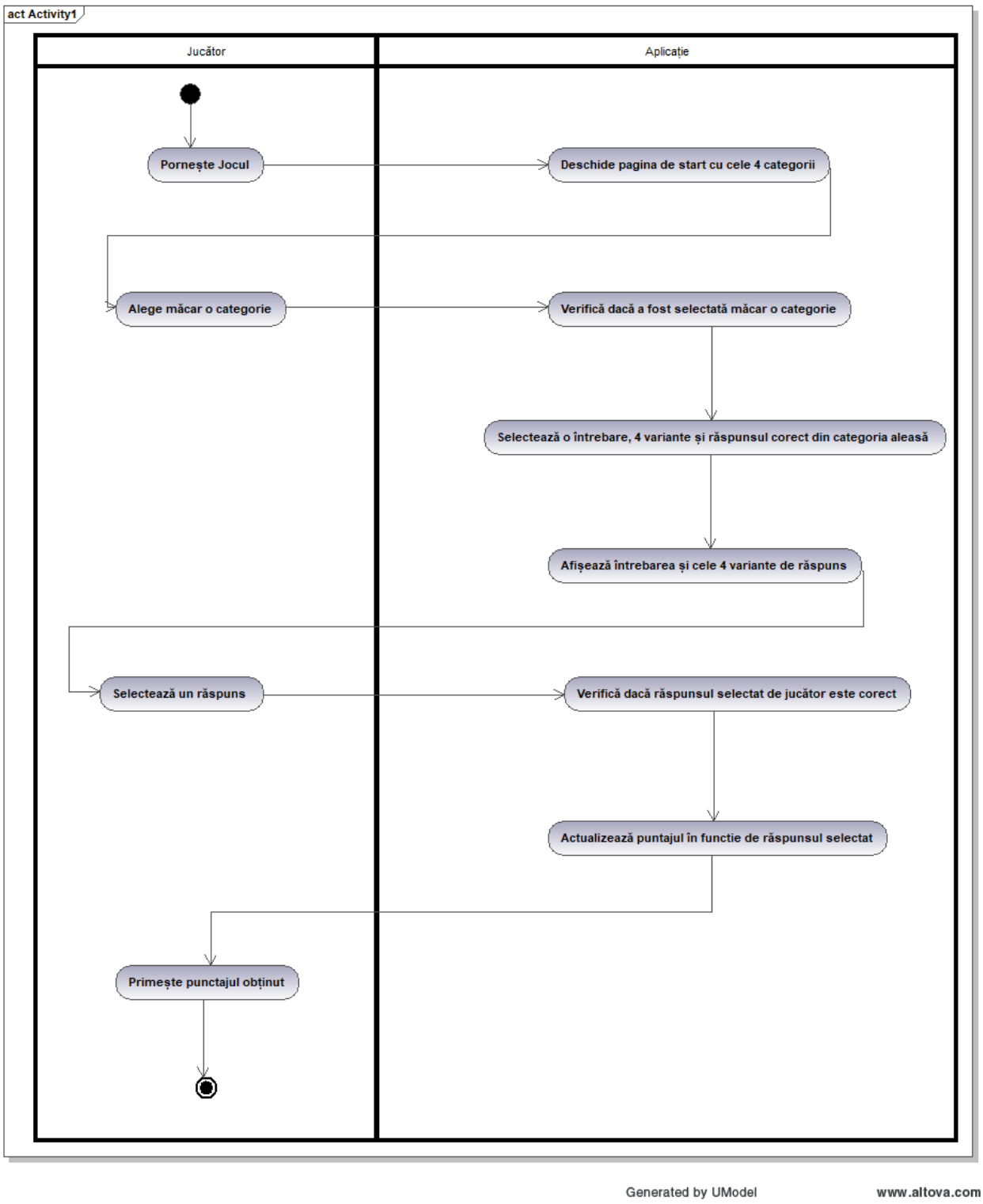
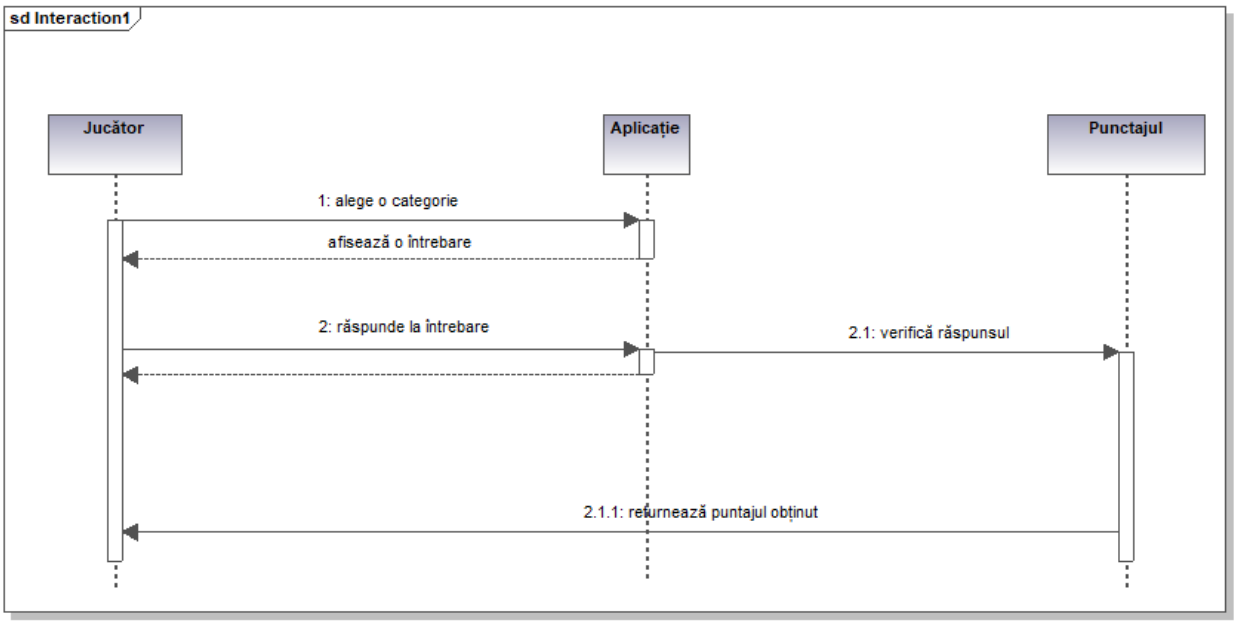
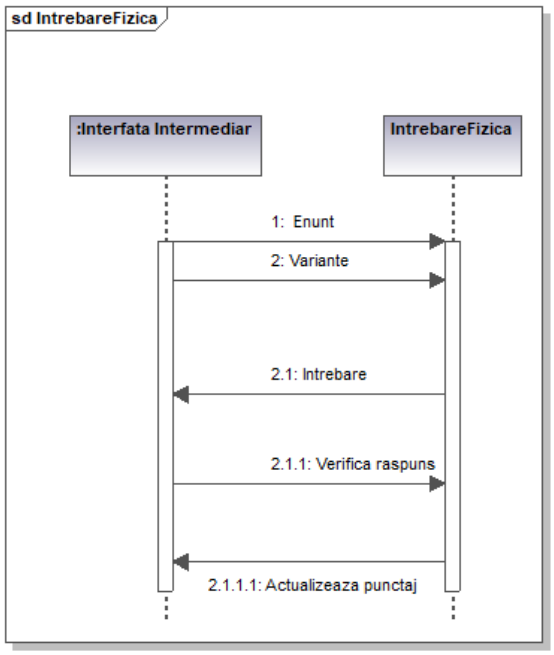


Diagrama de secvențe



Generated by UModel www.altova.com



Generated by UModel www.altova.com

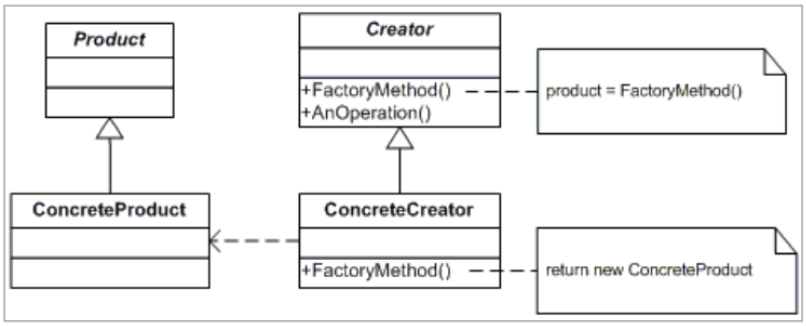
Utilizarea unui șablon de proiectare

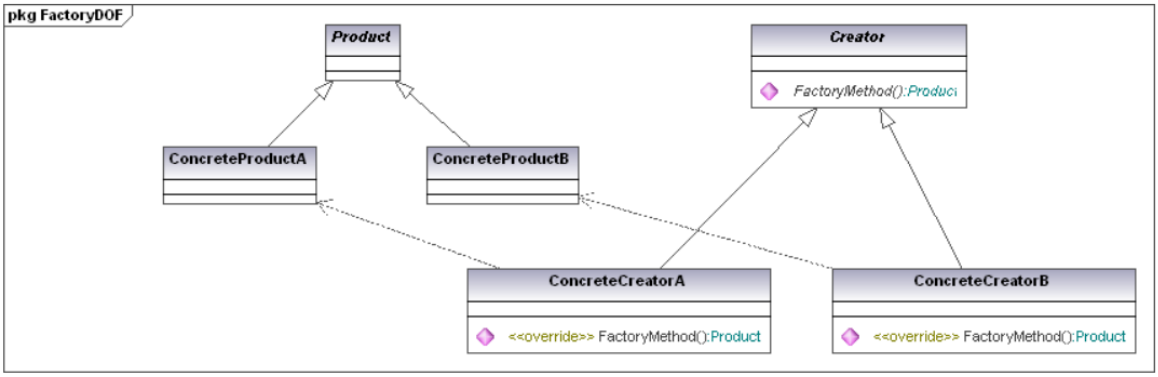
In acest proiect s-au folosit următoarele șabloane creaționale Factory Method și Singleton.

Factory Method

Șablonul Metoda fabrică definește o interfață pentru crearea unui obiect, dar lasă subclasele să decidă ce clasă să instanțieze

- Permite unei clase să cedeze subclaselor instanțierea de obiecte
- De fapt, clientul/utilizatorul decide efectiv!

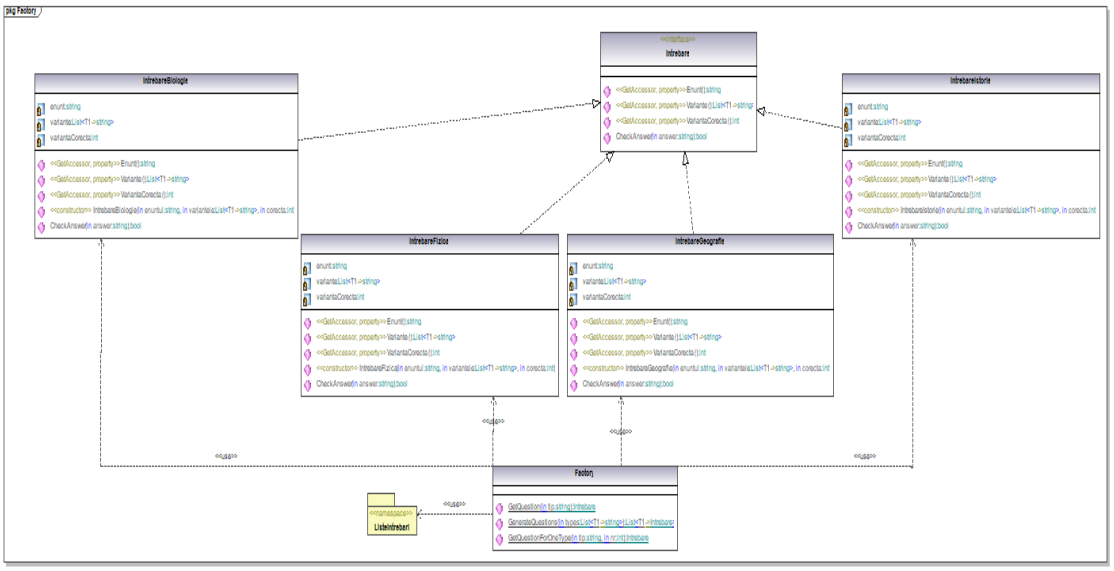




Metoda fabrică se folosește când:

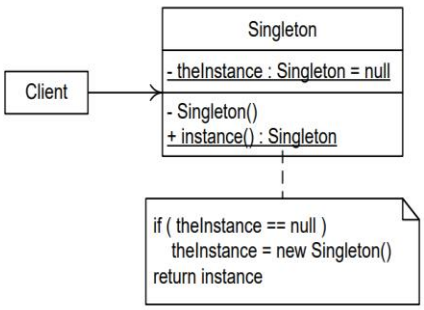
- O clasă nu poate anticipa din ce clasă trebuie să creeze un obiect
- O clasă dorește ca specificarea clasei obiectului creat să fie făcută de subclasele ei
- Informațiile despre delegare trebuie localizate

In cazul proiectului nostru șablonul a fost implementat și este reprezentat prin această diagrama UML



Singleton

Șablonul Singleton asigură faptul că o clasă are doar o singură instanță și furnizează un punct de acces global către aceasta. In proiectul nostru acesta a fost implementat pentru form deoarece trebuie să fie o singură instanță a interfeței.




```
public partial class Final : Form
{
    private static Final _form;

    public Final()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {
        this.Hide();
        StartPage.GetInstance().ShowDialog();
        this.Close();
    }
    public static Final GetInstance()
    {
        if (_form == null)
        {
            _form = new Final();
        }
        return _form;
    }
}
```

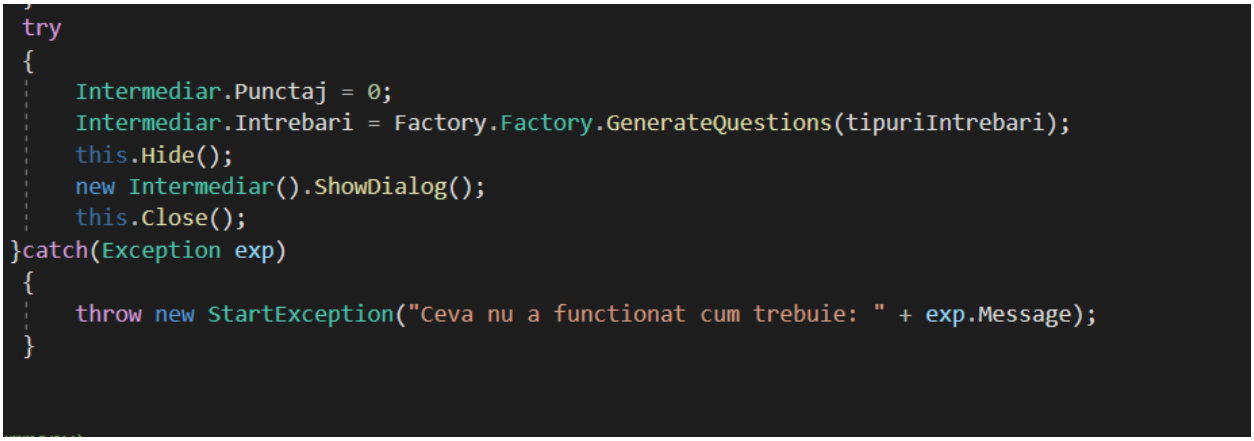
Tratarea exceptilor

S-a realizat prin blocuri de try-catch dar si prin crearea unei clase de exceptie, care mosteneste clasa Exception

```
/// <summary>
    /// Clasa care mosteneste clasa Exception si o personalizeaza pentru exceptile care
sunt la inceputul programului
```

```
    /// </summary>
        class StartException : Exception
    {
        private string _message;
        /// <summary>
        /// Metoda care realizează override la membrul message
        /// </summary>
        public override string Message
        {
            get
            {
                return _message;
            }
        }
        /// <summary>
        /// Metoda care realizeaza functionalitatea noi exceptii create
        /// </summary>
        public StartException(string message) : base()
        {
            _message = message.ToString();
        }
    }
}
```

Si care este folosita in form1.cs in cazul devaforabil in care ceva nu functioneaza cum trebuie



```
try
{
    Intermediar.Punctaj = 0;
    Intermediar.Intrebari = Factory.Factory.GenerateQuestions(tipuriIntrebari);
    this.Hide();
    new Intermediar().ShowDialog();
    this.Close();
}catch(Exception exp)
{
    throw new StartException("Ceva nu a functionat cum trebuie: " + exp.Message);
}
```

Modul de utilizare a programului abordat din perspectiva unui utilizator care nu cunoaște programul

Test de cultură generală

Smarty Pants este un joc de cultură generală pentru orice persoană dornică să-și testeze cunoștințele acumulate în diferite domenii. Jocul se poate juca doar de o singură persoană sau mai multe dacă se dorește o întrecere între toate persoanele prezente în fața unui calculator, iar jocul să fie câștigat de cel cu punctajul final cel mai mare.

Această aplicație presupune un test de cultură generală pe mai multe topicuri, topicurile sunt:

- Istorie
- Geografie
- Fizică
- Biologie

După ce răspunzi la toate întrebările, jocul se încheie și vei primi un punctaj.

Selectarea domeniului

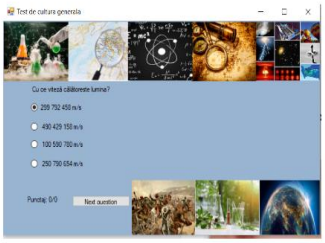
Din meniu poți să selectezi domeniul din care dorești să fie puse întrebările.

După preferință, se pot selecta mai multe domenii, întrebările for fi alese aleatoriu din domeniile selectate.
După ce te-ai decis asupra domeniilor, poți să apeși butonul de începe testul pentru a porni.

Intrebările

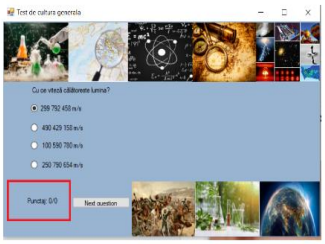
Fiecare domeniu are 20 de întrebări la care trebuie să răspunzi.

Aplicația va afișa întrebarea și 4 variante de răspuns.



Poți alege una din cele 4 variante de răspuns, după ce ai ales răspunsul, poți apăsa butonul de ,, Next Question" pentru a merge la următoarea întrebare.

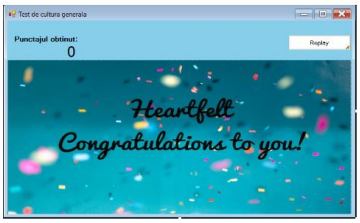
Pe parcursul testului, poți vedea punctajul obținut până acum în partea de stânga jos a ecranului.



Rezultatul

După ce răspunzi la cele 20 de întrebări, aplicația îți va arăta scorul obținut.

În acest punct poți să alegi să închizi aplicația sau să reîncepi testul apăsând butonul „Fă testul iar".



Anexa la documentație

Listingul părților semnificative din cod

Conținutul pentru o categorie de întrebare, de exemplu Biologie

biologie - Notepad

File Edit Format View Help

```
Ce este buburuza?
În ce organ se află coardele vocale la om?
Care este cel mai mare mamifer din lume?
Care este cel mai rapid animal din lume?
Care este animalul desertului?
În ce se transformă omizile când se dezvoltă?
Care este cea mai mică pasăre din lume?
Câte inimi are o caracatită?
Câte oase are organismul unui adult?
Ce sunt ribozomii?
Ce proprietati are membrana celulara?
Ce sunt incluziunile citoplasmatice?
Crustaceii sunt:
Fiecare dintre celulele-fiice formate prin diviziunea mitotică a unei celule-mamă cu 2n = 22 cromozomi are:
Fruct cărnos întâlnit la angiosperme este:
Fotosinteza:
Hormon hipofizar este:
Sângele din ventriculul drept al inimii este preluat de:
Urina se formează în:
Anexita este afecțiune a sistemului:
```

variante_biologie - Notepad

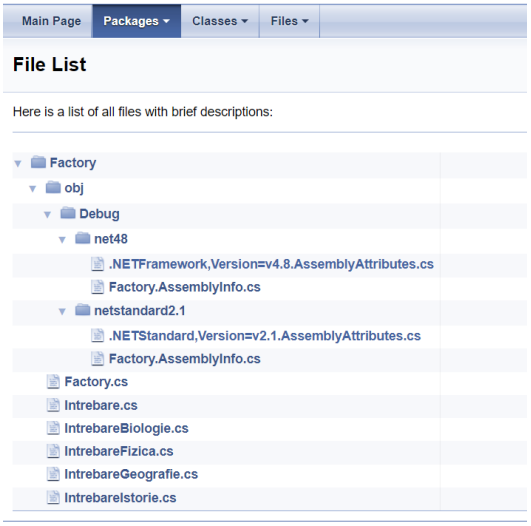
File Edit Format View Help

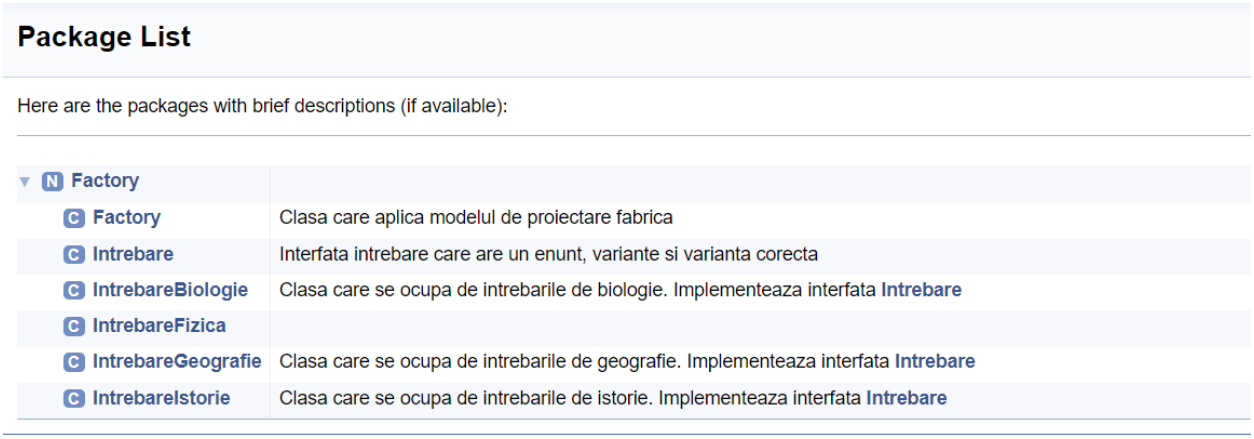
```
Arahnidă# Insectă# Reptilă# Mamifer
Limbă# Faringe# Laringe# Stomac
Urs# Balenă# Elefant# Hipopotam
Ghepardul# Iepurele de câmp# Antilopa# soimul călător
Vipera de nisip# Cămila# Gazela# Coiot
Ou# Larvă# Fluture# Libelulă
Colibri-albină# Rândunică# Privighetoare# Botgros
2# 3# 4# 5
598# 206# 376# 328
au rol în respiratia celulară# nu se găsesc în neuron# nu însotesc niciodată canaliculele reticulului endoplasmatic# au rol în sinteza proteinelor
contine materialul genetic al celulei# este impermeabilă# contine mitocondrii# este organizată după modelul mozaicului fluid
sunt reprezentate de miofibrile si neurofibrile# sunt întâlnite si în citoplasmă, dar si în nucleu# sunt  reprezentate de granule de substante de rezervă
artropode# celenterate #cordate #nematelminti
2n = 22 cromozomi# 2n = 11 cromozomi# n = 22 cromozomi# n = 11 cromozomi
achena# drupa# nuca# silicva
are loc în absența pigmenților asimilatori# constă în sinteza de substante organice# este un tip de nutriție heterotrofă# eliberează energie luminoasă
ADH-ul# insulina# STH-ul# tiroxina
artera pulmonară# artera aortă# venele cave# venele pulmonare
nefron# ureter# uretră# vezica urinară
digestiv# excretor# muscular# reproducător
```

Documentatia a fost generata si prin doxygen pe baza comentarilor care au fost adaugate in cod

Descrierea pentru fiecare clasa pe care le folosim la Factory

Factory





Pentru

Pentru a valida funcționalitatea implementarilor am creat o serie de cazuri de testare în care am încercat diferite abordări.

Cazuri de testare

```
public void FactoryGetMethodTestFizica()
{
    Factory.Intrebare a = Factory.Factory.GetQuestion("FIZICA");
    Assert.IsNotNull(a);
}
[TestMethod]
public void FactoryGetMethodTestBiologie()
{
    Factory.Intrebare a = Factory.Factory.GetQuestion("BIOLOGIE");
    Assert.IsNotNull(a);
}

[TestMethod]
public void FactoryGenerateQuestionsTestI()
{
    List<Factory.Intrebare> a = Factory.Factory.GenerateQuestions(new List<string>() { "ISTORIE" });
    Assert.AreEqual(20, a.Count);
}
[TestMethod]
public void FactoryGenerateQuestionsTestG()
{
    List<Factory.Intrebare> a = Factory.Factory.GenerateQuestions(new List<string>() { "GEOGRAFIE" });
    Assert.AreEqual(20, a.Count);
}
[TestMethod]
public void FactoryGenerateQuestionsTestB()
{
    List<Factory.Intrebare> a = Factory.Factory.GenerateQuestions(new List<string>() { "BIOLOGIE" });
    Assert.AreEqual(20, a.Count);
}
```

Implementarea de la Factory cum a fost realizata

```
/**************************************************************************
 *                                                                        *
 *  File:        Factory.cs                                               *
 *  Copyright:   (c) 2021-2022, Răzvan-Andrei Cănuci                      *
 *  E-mail:      razvan-andrei.canuci@student.tuiasi.ro                   *
 *  Description: Acest fișier este responsabil cu generarea întrebarilor  *
 *               și trimiterea lor către aplicație folosind șablonul      *
 *               creațional Factory.                                      *
 *                                                                        *
 *                                                                        *
 *                                                                        *
 *                                                                        *
 *                                                                        *
 *                                                                        *
 **************************************************************************/

using System;
using System.Collections.Generic;
using System.Threading;

namespace Factory
{
    /// <summary>
    /// Clasa care aplica modelul de proiectare fabrica
    /// </summary>
    public class Factory
    {
        /// <summary>
        /// Metoda care construieste intrebarea selectata pentru mai multe tipuri
selectate de pe UI(comentarii mai jos)
        /// </summary>
        /// <param name="tip">Tipul intrebarii</param>
        /// <returns>Un obiect de tipul intrebarii trimise ca parametru</returns>
```

```
        public static Intrebare GetQuestion(string tip)
        {

            int num;
            switch(tip)
            {
                case "ISTORIE":
                    Thread.Sleep(50);
                    num = new
Random().Next(ListeIntrebari.IntrebariIstorie.enunturi.Count);
                    return new
IntrebareIstorie(ListeIntrebari.IntrebariIstorie.enunturi[num],ListeIntrebari.IntrebariIs
torie.variante[num],ListeIntrebari.IntrebariIstorie.varianteCorecte[num]);
                case "FIZICA":
                    Thread.Sleep(50);
                    num = new
Random().Next(ListeIntrebari.IntrebariFizica.enunturi.Count);
                    return new
IntrebareFizica(ListeIntrebari.IntrebariFizica.enunturi[num],ListeIntrebari.IntrebariFizi
ca.variante[num],ListeIntrebari.IntrebariFizica.varianteCorecte[num]);
                case "GEOGRAFIE":
                    Thread.Sleep(50);
                    num = new
Random().Next(ListeIntrebari.IntrebariGeografie.enunturi.Count);
                    return new
IntrebareGeografie(ListeIntrebari.IntrebariGeografie.enunturi[num],ListeIntrebari.Intreba
riGeografie.variante[num],ListeIntrebari.IntrebariGeografie.varianteCorecte[num]);
                case "BIOLOGIE":
                    Thread.Sleep(50);
                    num = new
Random().Next(ListeIntrebari.IntrebariBiologie.enunturi.Count);
                    return new
IntrebareBiologie(ListeIntrebari.IntrebariBiologie.enunturi[num],ListeIntrebari.Intrebari
Biologie.variante[num],ListeIntrebari.IntrebariBiologie.varianteCorecte[num]);
                default:
                    return null;
            }
        }
        /// <summary>
        /// Metoda care genereaza toate cele 20 de intrebari in functie de cate tipuri au
fost selectate de UI, daca este mai mult de un tip selectat
        /// </summary>
        /// <param name="types">Tipurile selectate din UI</param>
        /// <returns>O lista cu toate intrebarile generate aleator</returns>
        public static List<Intrebare> GenerateQuestions(List<string> types)
        {
            List<Intrebare> list = new List<Intrebare>();
            switch(types.Count)
            {
                case 1:
                    for(int i=0;i<20;i++)
                    {
                        list.Add(GetQuestionForOneType(types[0],i));
                    }
                    break;
                case 2:
                    for (int i = 0; i < 10; i++)
                    {
```

```
                    list.Add(GetQuestion(types[0]));
                    list.Add(GetQuestion(types[1]));
                }
                break;
            case 3:
                for (int i = 0; i < 6; i++)
                {
                    list.Add(GetQuestion(types[0]));
                    list.Add(GetQuestion(types[1]));
                    list.Add(GetQuestion(types[2]));
                }
                list.Add(GetQuestion(types[0]));
                list.Add(GetQuestion(types[1]));
                break;
            default:
                for (int i = 0; i < 5; i++)
                {
                    list.Add(GetQuestion(types[0]));
                    list.Add(GetQuestion(types[1]));
                    list.Add(GetQuestion(types[2]));
                    list.Add(GetQuestion(types[3]));
                }
                break;
        }
        return list;

    }
    /// <summary>
    /// Metoda care este folosita pentru generarea intrebarilor daca este doar un tip
selectat pe UI
    /// </summary>
    /// <param name="tip">Tipul selectat de pe UI</param>
    /// <param name="nr">Numarul de ordine al intrebarii(maximul este 20)</param>
    /// <returns>Intrebarea in functie de tipul selectat</returns>
    public static Intrebare GetQuestionForOneType(string tip,int nr)
    {

        switch (tip)
        {
            case "ISTORIE":
                return new
IntrebareIstorie(ListeIntrebari.IntrebariIstorie.enunturi[nr],
ListeIntrebari.IntrebariIstorie.variante[nr],
ListeIntrebari.IntrebariIstorie.varianteCorecte[nr]);
            case "FIZICA":
                return new
IntrebareFizica(ListeIntrebari.IntrebariFizica.enunturi[nr],
ListeIntrebari.IntrebariFizica.variante[nr],
ListeIntrebari.IntrebariFizica.varianteCorecte[nr]);
            case "GEOGRAFIE":
                return new
IntrebareGeografie(ListeIntrebari.IntrebariGeografie.enunturi[nr],
ListeIntrebari.IntrebariGeografie.variante[nr],
ListeIntrebari.IntrebariGeografie.varianteCorecte[nr]);
            case "BIOLOGIE":
                return new
IntrebareBiologie(ListeIntrebari.IntrebariBiologie.enunturi[nr],
```

```
ListeIntrebari.IntrebariBiologie.variante[nr],
ListeIntrebari.IntrebariBiologie.varianteCorecte[nr]);
                default:
                    return null;
            }
        }

    }
       }
```

Listă ce a lucrat fiecare:

Boldureanu G. Gelu-Florin, grupa 1307B

- Crearea fișierului de help și integrarea lui
- Crearea și completarea antetului pentru fiecare fișier
- 6 cazuri de test
- Tratarea excepților
- Generarea de documentatie prin Doxygen

Boțic F.G. Teodora, grupa 1306A

- Crearea unui document word cu toate întrebările
- Crearea fișierelor pentru fiecare domeniu
- Scrierea documentului specificării cerințelor(SRS) după modelul IEEE prezentat în cursul 3
- Codul este comentat

Cănuci M.C Răzvan-Andrei, grupa 1307B

- Arhitectura jocului:
    1. Cele 3 interfețe și funcțiile de callback
    2. Factory.cs
    3. Intrebare.cs
    4. IntrebareGeografie.cs
    5. IntrebareIstorie.cs
    6. IntrebareBiologie.cs
    7. IntrebareFizica.cs
    8. UnitTest.cs , 14 din cele 20 de cazuri de testare
- Implementarea șablonului creațional Factory
- Implementarea șablonului creațional Singleton
- Implementarea fiecărui modul într-un DDL

Vecliuc Draia-Teodora, grupa 1307B

- Integrarea întrebărilor în joc, fișierul ListeIntrebari.cs
- Diagramele UML
    1. Cazuri de utilizare
    2. Diagrama de clase
    3. Diagrama activități
    4. Diagrama de secvențe
- Scrierea documentației
- Adăugarea de îmbunătățiri la partea de design a aplicației