



DOMAĆA ZADAĆA br. 1

– RAZVOJ PROGRAMSKIH RJEŠENJA –

Prezime i ime: **Ahmed Popović**

Broj indeksa: **16963**

Grupa: **6**

14.11.2015.

Sarajevo



NAPOMENA

Ovu zadaću sam predao prije 13.11., međutim zbog interfejs (korisnički interfejs) dijela aplikacije jako sam kasno primjetio da je ona imala problem u Meni klasi (call stack se povećavao i povećavao, ponašala se klasa kao neka beskonačna rekurzija). Problem sam riješio, te u .rar fajlu u kojem dolazi ova dokumentacija šaljem staru (prije 13.11 poslanu) verziju i novu verziju sa *minornim* izmjenama u Meni klasi koje popravljaju problem koji je postojao u prethodnoj verziji. Funkcionalnost je apsolutno ista u obje verzije i nije dodano ništa što već ranije od zahtjeva nije bilo implementirano.

Uvod

Rješenje koje je razvijeno je nazvano „Exams“. Sami kod rješenja vezan za zahtjeve zadatke nije puno komentaran jer je relativno shvatljiv samim čitanjem koda (to je projekat **rpr1**), te će poneki dijelovi koda biti komentarisani ovdje. Međutim, jedan dio koda je relativno težak za koristiti bez dokumentacije – pa je ta klasa (zapravo to je više klasa koje su unutar projekta **UI**) komentarisana sa XML stilom komentiranja, pa samim tipkanjem raznih funkcija i atributa u Visual Studio, svaka povratna varijabla, svaki parametar i svrha funkcije će biti lijepo pojašnjeni.

Budući da je naporno smišljati pitanja i unositi ispite (a tek polagati ispite!), program dolazi sa dva „demo“ fajla students.bin i exams.bin koji se mogu učitati u program kroz interfejs voditelja/profesora. Demo fajlovi imaju 3 testna ispita (IM1, IM2, TP) i 3 testna studenta, koji su već polagali neke ispite.

Sistemske zahtjevi

Iako je samo rješenje funkcionalno jednostavno i malo je čudno što je potrebno spomenuti sistemske zahtjeve, interfejs je ipak malo „ambiciozniji“ i zauzima malo više sistemskih resursa nego inače. „Redraw“ čitavog console screena se često radi kada se kreće po menijima, pa je *poželjno* imati malo jači CPU kako bi biranje opcija po meniju glatko radilo.

Program je testiran na 3 računara i uredno je radio na svakom (AMD X4 750K, drugi je i7 [tačan model nepoznat], te jos jedan laptop sa dvojezgrenim procesorom), međutim, nažalost, ne isključuje se mogućnost da je meni malo „štekav“ na slabijim računarima.

Da bi se program pokrenuo, potrebno je da sistem ima instaliran .NET Framework 4.



Analiza zahtjeva

Odlučeno je sljedeće:

- Postojeće samo jedan voditelj u programu i neće biti potrebna odvojena klasa za njega
- Postojeće mogućnost unošenja više ispita tipa A i tipa B
- Klase za ispis po konzoli će biti odvojen projekat i time DLL koji će biti referenciran u glavnom projektu
- Biće podržana lahka izmjena raznih validacijskih kriterija za korisnički unos
- Program će biti lokalizovan (imaće podršku za više jezika)
- Voditelj ne bi trebao da zna koji to studenti imenom i prezimenom polažu ispit, već samo da ispite pregleda po *šifri*

Realizacija zahtjeva

Klasa Exam predstavlja baznu klasu iz koje je naslijeđen ExamA i ExamB. Postoji i bazna klasa Question iz koje su naslijeđene klase RegularQuestion i FillInQuestion. ExamA i ExamB koriste interfejs IExamQuestionAdder. Postoji i klasa TakenExam koja pamti tačne i netačne odgovore studenta. Uz ovaj .pdf fajl dolaze i dvije slike **RPR Dijagram** i **UI Dijagram** gdje su prikazane sve klase, njihova hijerarhija. Logičke cjeline (ukoliko nema *prave* hijerarhije) su zajedno grupirane.

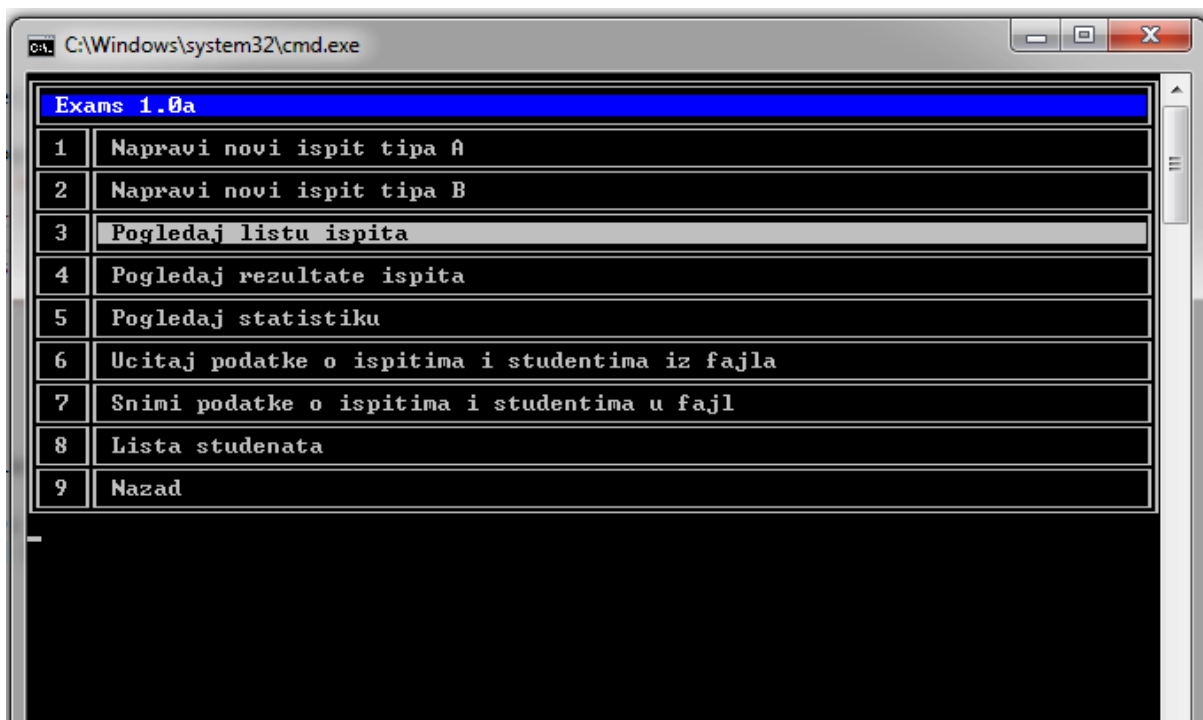
Slika **UI Dijagram** prikazuje klase koje su razvijene za prikaz informacija u konzoli. Klase unutar UI projekta su relativno kompleksne i jako pomažu pri unosu informacija, pored tabelarnog (konzoli prilagodljivog) prikaza.

Polimorfizam se koristi kod Exam i Question klase. Program klasa je parcijalna klasa i sadrži svo kreiranje menija, formi za unos pitanja, odgovaranja itd. Naravno, svo to kreiranje je izvedeno sa UI klasama.

Dekompozicija programa je ispoštovano tako što su veće logičke cjeline odvojene u svoje zasebne fajlove.

Struktura programa

Pri ulasku u program, prvo je potrebno izabrati željeni jezik. Potom se bira interfejs voditelja/profesora ili studenta.



Meniji poput ovih se u kodu kreiraju na sljedeći način:

```
Menu m = new Menu(ProgramTitle + " - Welcome / Dobro dosli");
m.AddOption("English language", Menu_EnglishLanguage)
  .AddOption("Bosanski jezik", Menu_BosnianLanguage)
  .AddNote("")
  .AddNote("Use arrows up and down, and the Enter key to make a choice.")
  .AddNote("Koristite strelice gore i dolje, te tipku Enter da nacinite
  izbor.")
  .Draw();
```

U slučaju izabira *interfejsa voditelja*, otvaraju se mogućnosti kreiranja novih ispita, pregledanja pitanja već kreiranih ispita, pregledanje rezultata polaganih ispita, učitavanje i snimanje podataka u fajl, te lista studenata. U slučaju izabira *interfejsa studenta*, otvaraju se mogućnosti polaganja ispita i pregledavanja ranije polaganih ispita. Svaki student ima svoj ključ (moglo se napraviti da svaki student ima username i password, ali je za ovaj program sasvim dovoljno da svaki student ima samo šifru za polaganje ispita koja će ujedno služiti i kao identifikator tog studenta).

Ukoliko student nema ključ od ranije, ukucaće bilo šta kao ključ i bit će mu ponuđena forma za registraciju. Pritiskom na Y ili N (yes ili no), forma se pokazuje i ispunjava se sa Enterom poslije svakog ispunjenog polja.

Exams 1.0a	
Unesite ključ za prijavu	f
Ime	Ahmed
Prezime	Po_
Datum rođenja (u dd.mm.yyyy formatu)	
Naziv fakulteta	
Godina studija	

Ove forme za registraciju **odmah validiraju** unesene podatke za ispravnost po datim kriterijima na sljedeći način:

```
List<object> studentData = ui.MultipleInputString(new string[] {
    strings.StudentFirstName,
    strings.StudentLastName,
    strings.StudentBirthDay,
    strings.StudentFaculty,
    strings.StudentStudyYear
}, new string[] {
    "string",
    "string",
    "date",
    "string",
    "int"
}, new Func<object, bool>[] {
    x => Utility.NumberOfWords(x.ToString()) == 1, // Maximum one word in first
    x => Utility.NumberOfWords(x.ToString()) == 1, // and last name
    x => true, // no validation here
    x => Utility.NumberOfWords(x.ToString()) < 10, // max 10 words in faculty
    x => (int)x < 11 && (int)x > 0 // study year in [1,10]
});
```

```
Student s = new Student((string)studentData[0], (string)studentData[1],
    (DateTime)studentData[2], (string)studentData[3], (int)studentData[4]);
```

Uspješno registrovani student se snima u

```
private static List<Student> Students = new List<Student>();
```

Na sličan način, dodani ispiti se snimaju u

```
private static List<Exam> Exams = new List<Exam>();
```

što se sve nalazi u klasi Program.



Na mnogo mjesta su iskorištene lambda funkcije i Func klase, recimo pri traženju tačnog odgovora

```
answers.FindIndex(x => x == q.CorrectAnswer)
```

Ili traženju da li je student već polagao određeni ispit

```
if (Exams.Find(x => x == e && x.TakenExams.Find(y => y.Item1 == studentKey) != null) != null)
```

Neke od klasa razvijenih kao dio ovog rješenja imaju „fluent interface“, gdje većina funkcija vraća *sebe*, pa je moguće napraviti konstrukcije poput one već prikazane za menije ili za kreiranje pregleda ispita i tačnih odgovora i sl.

Prikaz polaganih ispita je izveden tako što se funkciji prikaza proslijede Exam e i za njega vezani TakenExam te koja u lijepom formatu ispisuje rezultate, tačne i korisnički izabrane odgovore:

Exams 1.0a

--- IM1 - B / Polozen ---

1. Urijednost treceg korijena iz 8 je

1	2	63
---	---	----

2. Profesor ovog predmeta je

prof. Narcis Behlilovic	prof. Dzenana Djonko	prof. Huse Fatkic
-------------------------	----------------------	-------------------

3. Rolle-ova teorema kaze da svaka realna funkcija

koja je neprekidna, diferencijabilna na intervalu $\langle a, b \rangle$ i postoje tačke a i b da je $f(a) = f(b)$, onda postoji c koje pripada $\langle a, b \rangle$ takvo da je $f'(c) = 0$	koja je prekidna je diferencijabilna na mjestima prekida i u njima je $f'(a) = a$	ima a takvo da je $f(a) = -1$
---	---	---------------------------------

4. Gauss-Ostrogradski teorema se tice

diskretnih funkcija	samo kvadratnih realnih funkcija	vektorskih polja
---------------------	----------------------------------	------------------

5. Stacionarna tacka funkcije $f(x)$ je tacka a za koju vazi

$f(x) * f'(x) = f(x^2)$	$f'(a) = 0$	$f'(f(a)) = 0$
-------------------------	-------------	----------------

6. Gauss-Ostrogradski teorema je specijalni slucaj

Stokesove teoreme	Lagranzove teoreme	Pitagorine teoreme
-------------------	--------------------	--------------------

7. Jedno od rjesenja $x^2=16$ je

4	-4	2	-2	16
---	----	---	----	----

8. Pravougaonik cija je površina 20 može imati stranice

$a=5, b=5$	$a=5, b=4$	$a=4, b=5$	$a=6, b=3.14$	$a=20, b=0$
------------	------------	------------	---------------	-------------

Izaberite opciju

1	Nazad
---	-------



Demo podaci

Uz program dolaze students.bin i exams.bin fajlovi koji kada se učitaju u program iz voditelj/profesor interfejsa, daju tri već unešena ispita i studenta, te par već polaganih ispita koje je moguće gledati (IM1, IM2, TP). Korisnici i njihove šifre su:

Ahmed Popović
Šifra: **ZUSKI17LMG**

Kerim Popović
Šifra: **FB0W7F50FZ**

Muharem Serbezovski
Šifra: **UUI1ARW68N**

Nedostaci

Iako je aplikacija po mom mišljenju jako funkcionalna, mislim da svaki program ima mjesta za poboljšanje, ispravljanje, pa tako baš i ovaj. Jedna od mogućnosti kada bi se pravilo rješenje u pravom svijetu, bilo bi da ima više voditelja. Međutim, u ovom zadatku nije bilo poente za više voditelja budući da svaki voditelj ima jednaka, odnosno, nepostojeća ograničenja. Dakle, eventualno poboljšanje bi bilo da postoji više voditelja, ali sa različitim nivoima pristupa.

Još jedno unaprijeđenje bi bilo da se pitanja, ispiti i rezultati snimaju u SQL databazu, odnosno da je program samo frontend za tu databazu i služi za unos, izmjenu i brisanje podataka iz iste. Prvo sam planirao ovako uraditi zadatak, međutim vjerujem da je ipak poenta zadatke bila da nas nauči dekompoziciji, uočavanju klasa i sl.

Mislim da su se određene klase mogle još bolje (strateški) promisliti, mada je i ovako poprilično „modularno“, i lahko je dodati nove tipove ispita ili promijeniti kriterije.