



Šolski center Celje
Splošna in strokovna gimnazija Lava

Kviz
(raziskovalna naloga)

Mentor:
Borut SLEMENŠEK, univ. dipl. inž.

Avtor:
Boris ŠPOLJAR, GL 4.f

Marec, 2005

1. Kazalo vsebine

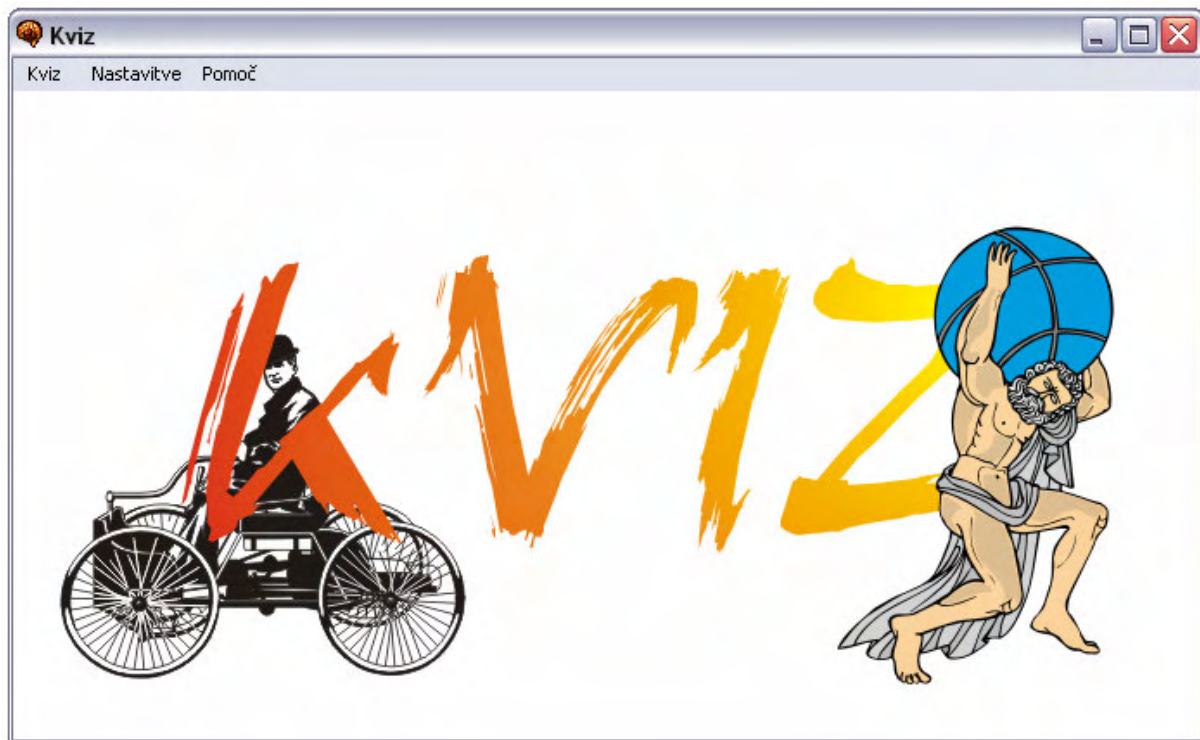
1. Kazalo vsebine	1
2. Kazalo slik.....	1
3. Povzetek	2
4. Uvod.....	3
4.1. Predstavitev raziskovalnega problema.....	3
4.2. Cilj.....	3
4.3. Opis raziskovalnih metod.....	3
5. Postavitev podatkovne baze	4
6. Povezava programa s podatkovno bazo	4
7. Priprava testa	5
7.1. Razvrstitev odgovorov	5
8. Povezava programa s podatkovno bazo	5
9. Spreminjanje privzetih nastavitev	6
10. Grafični dodatki.....	6
11. Vnašanje zapisov v podatkovno bazo	8
12. Statistika	9
13. Zaključek.....	11
14. Viri in literatura	12
15. Priloge	13
16. Zahvala	14

2. Kazalo slik

Slika 1:Program ob zagonu	2
Slika 2:Oblika glavnega okna	4
Slika 3:V tem oknu je mogoče spreminjati privzete nastavitve programa.....	6
Slika 4:Pred vprašanjem je zaporedna številka trenutnega vprašanja.....	7
Slika 5:Grafični prikaz pravilnosti odgovorov	7
Slika 6:Prikaz pravilnega odgovora	7
Slika 7:Vnašanje zapisov v tabelo.....	8
Slika 8:Okence za dodajanje novih tem	8
Slika 9:Ob koncu testa program prikaže statistiko.....	9
Slika 10:Če želimo shraniti svoj rezultat, nas program povpraša po imenu	9
Slika 11:Izpis rezultatov	10

3. Povzetek

Za svojo raziskovalno nalogo sem izdelal program Kviz, ki uporabniku postavlja vprašanja iz poljubne teme. Poleg vsakega vprašanja sodi tudi določena slika. Uporabnik se odloča med štirimi odgovori. Pravilnih odgovorov je lahko tudi več kot le eden. Uporabnik ima možnost spreminjanja privzetih nastavitvev, dodajanja vprašanj v tabelo ter shranjevanja rezultatov. Pri svojem delu sem presegel zadane cilje, ter izdelal uporabniku prijazno aplikacijo.



Slika 1: Program ob zagonu

4. Uvod

4.1. Predstavitev raziskovalnega problema

Za raziskovalno nalogo sem naredil program Kviz. Pobudo zanj sem dobil pri mentorju. Programiranje sem spoznal že v osnovni šoli. Takrat sem spoznal, da me to zelo veseli in sem se odločil, da v tej smeri nadaljujem svoj študij. Če želim biti nekoč v tem delu uspešen, moram izkoristiti vsako priložnost, da spoznam nova obzorja tega področja. Zato glede ponudbe mentorja nisem okleval.

4.2. Cilj

Na začetku sem si zadal cilj, da bom naredil program, ki bo uporabniku postavljaj vprašanja iz izbrane teme, uporabniku bo pa v pomoč slika, na katero se določeno vprašanje nanaša. Uporabniku bi naj bili podani štirje odgovori, on bi pa med njimi moral izbrati pravega. Vsi podatki bi naj bili shranjeni v bazi podatkov, ki jo lahko uporabnik dopolnjuje izven ali znotraj samega programa. Uporabnik bi naj imel možnost izbiranja med temami, dvema težavnostnima stopnjama, ter med številom postavljenih vprašanj.

4.3. Opis raziskovalnih metod

Pri delu sem uporabljal programski paket Borland C++ Builder 6, v katerem sem programiral v programskem jeziku C++. Podatkovno bazo sem izdelal v programu Microsoft Access 2002. Pri iskanju podatkov v tabeli sem uporabljal še povpraševalni jezik SQL.

5. Postavitev podatkovne baze

Odločil sem se, da bo najbolje, če bodo vprašanja shranjena v tabeli. Tako bo možno enostavno naknadno dodajanje zapisov. Naredil sem tabelo z relacijsko shemo:

Tabela(ID#,VPRASANJE,A,B,C,D,RESITEV,SLIKA,ZAHTEVNOST,TEMATIKA)

ID - je primarni ključ, ki predstavlja zaporedno številko vprašanja

VPRASANJE - je tekstovni atribut, omejen z stotimi znaki, v njem je zapisano vprašanje

A,B,C,D - so tekstovni atributi, omejeni z 100 znaki. V njih so zapisani odgovori na določeno vprašanje

RESITEV - je število, ki pove, kateri odgovor je pravilni

SLIKA - je tekstovni atribut, v katerem je zapisana pot do slike, ki pripada določenemu vprašanju

ZAHTEVNOST - je število, ki predstavlja težavnostno stopnjo vprašanja. Vrednost 1 predstavlja lahko, 2 pa težko vprašanje

TEMATIKA - je število, ki predstavlja tematiko posameznega vprašanja

6. Povezava programa s podatkovno bazo

Moje delo se je začelo tako, da sem na list papirja skiciral okno, kjer bi naj uporabnik med štirimi podanimi odgovori izbral pravega. Po skici sem oblikoval naslednje okno:



Slika 2:Oblika glavnega okna

Na oknu lahko vidite sliko, štiri gumbke za izbiro, drsnik, izpisano vprašanje ter dva gumbka. Vsak objekt je povezan s smiselnim atributom iz tabele. Vse slike, ki jih program kliče posredno preko tabele so velikosti 240x180 ter shranjene v formatu bmp. Slike sem dobil večinoma na internetu, nekaj pa sem jih dobil tudi iz raznih učbenikov za zgodovino.

7. Priprava testa

Ob začetku programa se ustvari dvodimenzionalna tabela:

```
int polje[30][3];
```

S to tabelo sem omejil maksimalno število vprašanj na trideset. Ta tabela omogoča, da program za vsako vprašanje zapiše tri podatke. Ko uporabnik klikne na gumbek »Začni« v meniju »Kviz«, se izvrši zanka tolikokrat, kot je število vprašanj v posameznem testu. Med izvajanjem zanke program vsakič izbere naključno število, ki predstavlja zaporedni zapis v tabeli. Če se ta zapis ujema z izbrano težavnostno stopnjo in tematiko, število zapisa shrani v tabelo na naslednji položaj:

```
polje[x][1];          x predstavlja številko vprašanja
```

7.1. Razvrstitev odgovorov

Ko sem oblikoval prvo okno, sem dejal, da bo za vsako vprašanje pravi en odgovor. S časom sem razmišljal, da bi bilo bolje, če bi dodal možnost, da je lahko pravih odgovorov več. Tabela sem preoblikoval tako, da sem kot rešitev zapisal število pravih odgovorov. Pravilni odgovori pa morajo biti nujno zapisani pred napačnimi. Pri prepisu odgovorov iz tabele se je seveda dogajalo, da so bili pravilni odgovori vedno na vrhu, napačni pa na dnu okna. Ta bi odpravil to težavo, sem pri pripravi testa dodal dodatno zanko, ki je za vsako vprašanje našla naključno število od vključno 1 do vključno 4. To število se zapiše v omenjeno dvodimenzionalno polje *polje[x][1]* in predstavlja razvrstitev odgovorov na oknu. Razvrstitev odgovorov sem spreminjal z lastnostjo *Top*, ki predstavlja oddaljenost objekta od zgornjega roba okna. To v bistvu pomeni, da se pri vsakem vprašanju gumbki za izbiro naključno pomešajo. Preverjanje pravilnosti odgovora s tem ni oteženo, saj je prva izbira še vedno zapisana v prvi prostorček za besedilo, druga izbira v drugi prostorček in tako naprej. Če je kot rešitev zapisano število dve, kar pomeni, da sta pravilna odgovora ena in dve, moramo preveriti ali sta prvi in drugi odgovor označena ter tretji in četrti odgovor neoznačena.

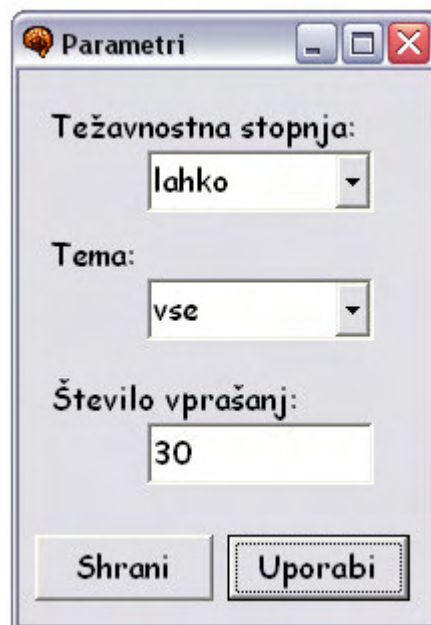
8. Povezava programa s podatkovno bazo

Podatkovno tabelo sem povezal s programom preko objekta *Query*. Ko uporabnik odgovori na vprašanje ter klikne na gumb »Naprej«, se uporabnikov odgovor na trenutno vprašanje zapiše v tabelo *polje[x][2]*, pozicija drsnika pa se poveča za ena. Program je narejen tako, da vsakič, ko se pozicija drsnika spremeni program pogleda v tabelo *polje[x][0]*, kjer je zapisana zaporedna številka zapisa vprašanja v tabeli. Program pogleda ta zapis v tabeli, ter posamezne attribute poveže s smiselnimi objekti v oknu programa. Nato še prebere *polje[x][1]*, ter glede

na prebrano število ustrezno razporedi gumbke za izbiro. Na začetku je drsnik zaklenjen, ter lahko njegovo pozicijo spreminjamo le tako, da kliknemo na gumb »Preveri«, ko pa pridemo na konec testa, lahko uporabnik poljubno spreminja pozicijo drsnika, vendar pa ne more več spreminjati odgovora, ampak lahko le opazuje, kako je prej odgovoril.

9. Spreminjanje privzetih nastavitev

Današnje aplikacije morajo biti zasnovane tako, da omogočajo uporabniku, da si lahko nastavi program po svojih željah. Na začetku sem kot privzete nastavitve v programu nastavljal da je število vprašanj 20, tematika nedoločena, težavnostna stopnja pa je lažja. Da bi si lahko uporabnik nastavljal svoje privzete nastavitve sem dodal mapo data, v njej pa tekstovno datoteko prizeto.txt. V prvi vrstici te datoteke je zapisana težavnostna stopnja, v drugi vrstici privzeta tematika, v tretji vrstici pa privzeto število vprašanj. Program je naučen, da ob vsakem zagonu to datoteko prebere in uporabi prebrane nastavitve. K programu sem dodal novo okno, v katerem lahko uporabnik spreminja to datoteko in s tem tudi privzete nastavitve.



Slika 3: V tem oknu je mogoče spreminjati privzete nastavitve programa

Ob kliku na gumb »Shrani« se izbrane nastavitve shranijo v datoteko prizeto.txt, ob pritisku na gumb »Uporabi«, pa se izbrane nastavitve uporabijo ne da bi se shranile v datoteko. Kot število vprašanj lahko uporabnik vpiše števila od 5 do 30. Težavnostni stopnji sta dve: lahka in težka. Kot temo lahko uporabnik izbere vse teme, ki se nahajajo v tabeli oziroma v datoteki tematika.txt (Glej pod naslov 11. Vnašanje zapisov v datoteko).

10. Grafični dodatki

Ko je osnova programa delovala pravilno, sem se lahko posvetil bolj prijaznemu grafičnemu vmesniku. Uporabnik je lahko glede na pozicijo drsnika videl pri katerem vprašanju je. Vendar je pri večjem številu vprašanj preštevanje črtic na drsniku zoprn. Zato sem z naslednjo vrstico dosegel, da se številka vprašanja izpiše pred vprašanjem.

`Label2->Caption=" "+Label37->Caption+". "+(QueryIVPRASANJE->AsString);`

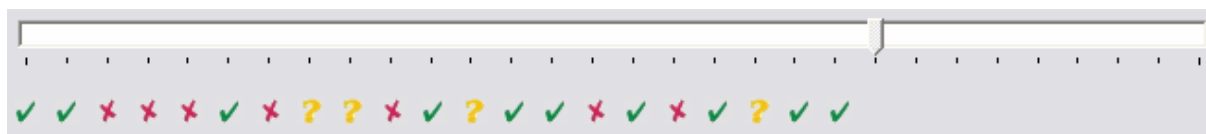
Label2 – je tekstovni prostorček, v katerega se izpiše vprašanje

Label37 – je tekstovni prostorček, v katerem je zapisana pozicija drsnika

5. V kolikšnem času se je potopil Titanic?

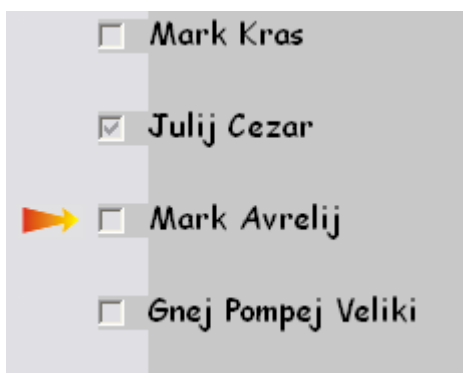
Slika 4:Pred vprašanjem je zaporedna številka trenutnega vprašanja

Da bi uporabnik lažje videl na katero vprašanje je odgovoril pravilno, na katero je odgovoril napačno in na katero ni odgovoril sem pod drsnik narisal štirideset slikic. Te slikice se postavijo pod drsnik glede na to, kakšno je število vprašanj v posameznem testu. Odvečne slike dobijo prazno sliko, tako da jih uporabnik ne opazi. Število vprašanj sem omejil na najmanj pet ter največ trideset. Ko uporabnik klikne na gumb »Pripravi test«, program preveri število vprašanj ter vedno na isto pozicijo postavi prvo sliko, nato pripravi dve števili, ki sta odvisni od števila vprašanj. Ti dve števili nato program prišteva vsaki naslednji sliki k lastnosti Left, torej vsako sliko glede na prejšnjo sliko za določeno število prestavi bolj na desno stran. Dve števili prištevam tako, da k vsaki sodi sliki prištejem prvo število, vsaki lihi pa drugo število. S tem dosežem večjo natančnost, ter da je vsaka slika pod ustrezno črtico na drsniku. Pripravil sem tri majhne slike, križec, vprašaj in kljukica. Kljukica označuje pravilni odgovor, križec napačen odgovor, vprašaj pa označuje, da uporabnik na to vprašanje ni odgovoril. Slikice se prikazujejo za vsako vprašanje posebej, ko kliknemo na gumb »Preveri«.



Slika 5:Grafični prikaz pravilnosti odgovorov

Če je uporabnik neučakan, sem dodal gumb »Pravilni odgovor«, ki uporabniku onemogoči spreminjanje odgovora trenutnega vprašanje ter prikaže puščico pred pravilnim odgovorom.



Slika 6:Prikaz pravilnega odgovora

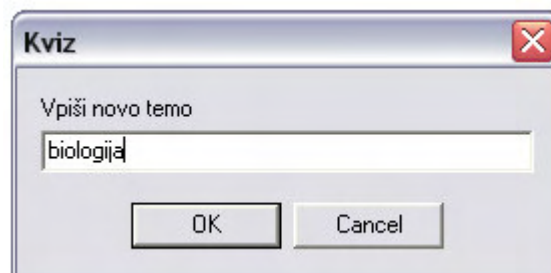
11. Vnašanje zapisov v podatkovno bazo

Da bi program omogočal enostavno razširitev podatkovne baze sem dodal novo okno, v katerem lahko uporabnik sam vpisuje svoja vprašanja na svojo temo.



Slika 7: Vnašanje zapisov v tabelo

Zgoraj se nahaja okno, v katerem lahko uporabnik vpiše in izbere vse lastnosti novega vprašanja. Zgoraj desno je drsnik, s katerim se lahko uporabnik spreha po zapisih v tabeli. Levo zgoraj se nahaja tekstovni prostor, kamor uporabnik vpiše vprašanje. Pod njim je prostor za sliko, pod sliko je še en tekstovni okvir, kamor uporabnik vpiše pot do svoje slike. Pravilnost zapisa poti lahko preverimo s pritiskom na gumb »Preizkusi sliko«. Poleg slike lahko uporabnik vpiše vse štiri možne odgovore, ter označi kateri odgovor oziroma odgovori so pravilni. Na desni strani še uporabnik izbere zahtevnost in tematiko novega vprašanja. Tukaj sem dodal tudi možnost, da uporabnik doda svojo temo. Vse teme sem shranil v tekstovni datoteki tematika.txt v mapi data. V tej datoteki se nahaja tema v vsaki vrstici. Iz te datoteke se preberejo možne teme tudi pri oknu za nastavljanje privzetih lastnosti programa. Če kliknemo na gumb »Dodaj temo« se pojavi novo okno, v katerega vpišemo naziv nove teme. Prebrana tema se shrani v datoteko tematika.txt.



Slika 8: Okence za dodajanje novih tem

Ko kliknemo na gumb »Shrani« program vse nastavitve shrani v tabelo. Posebej pazljiv mora biti program pri zapisu odgovorov, saj morajo biti pravilni odgovori nujno zapisani pred napačnimi. Ob zapisu v tabelo se ta zapre in ponovno odpre, tako da lahko uporabnik nemudoma vidi rezultat. Nov zapis se zapiše na konec tabele z ustrežno zaporedno številko.

Možnost vnašanja novih zapisov v tabelo in spreminjanja privzetih lastnosti je možna le, če trenutno nismo sredi reševanja testa. Zato sem dodal v meniju tudi gumb, ki zaključi trenutni test.

12. Statistika

Program je veliko bolj zanimiv, če omogoča hranjenje rezultatov ter poveča tekmovalnost med uporabniki. Ko pridemo s posameznim testom do konca, se gumbek »Pravilni odgovor« preimenuje v »Statistika«. Ta gumb nam prikaže kako smo odgovarjali na vprašanja v zadnjem testu.



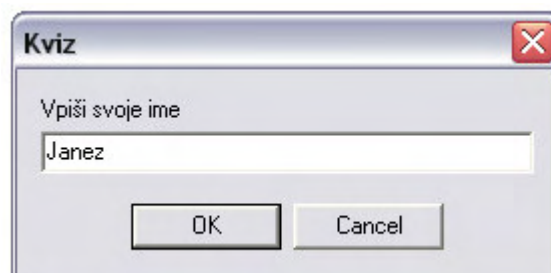
The screenshot shows a dialog box with the following data:

Pravilni odgovori	19
Neppravilni odgovori	5
Neodgovorjeni odgovori	1
Uspešnost v procentih	76

At the bottom of the dialog box are two buttons: "Shrani" and "Izhod". Below the dialog box is a larger button labeled "Statistika".

Slika 9:Ob koncu testa program prikaže statistiko

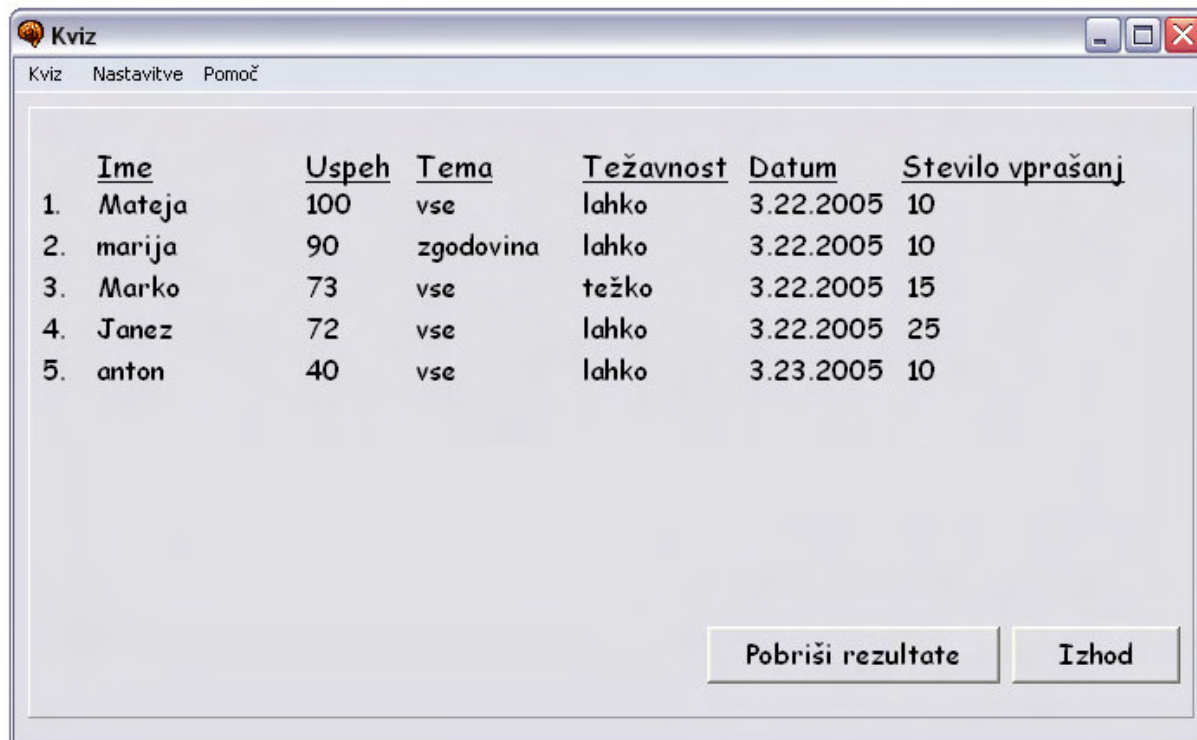
Uporabniku se izpišejo pravilni, nepravilni ter nedogovorjeni odgovori. Poleg tega program še izračuna uspešnost v procentih. Vse te podatke je možno shraniti v tabelo z rezultati rezultat.DBF. Ob kliku na gumb »Shrani« se pojavi okence, ki nas povpraša po imenu.



The screenshot shows a dialog box titled "Kviz" with a close button (X) in the top right corner. It contains a text input field with the label "Vpiši svoje ime" and the text "Janez" entered. At the bottom are "OK" and "Cancel" buttons.

Slika 10:Če želimo shraniti svoj rezultat, nas program povpraša po imenu

Poleg imena ter zgoraj naštetih podatkov se v tabelo shrani tudi težavnostna stopnja, tematika, število vprašanj ter datum. En rezultat lahko zapišemo le enkrat. Če želimo videti vse dosedanje rezultate moramo v meniju v glavnem oknu izbrati podmeni »Rezultati«. Ta nam prikaže prvih deset najboljših rezultatov v tabeli po uspešnosti. Seveda če je v tabeli sploh kakšen zapis. Če je tabela prazna, nas program na to opozori.



	<u>Ime</u>	<u>Uspeh</u>	<u>Tema</u>	<u>Težavnost</u>	<u>Datum</u>	<u>Število vprašanj</u>
1.	Mateja	100	vse	lahko	3.22.2005	10
2.	marija	90	zgodovina	lahko	3.22.2005	10
3.	Marko	73	vse	težko	3.22.2005	15
4.	Janez	72	vse	lahko	3.22.2005	25
5.	anton	40	vse	lahko	3.23.2005	10

Pobriši rezultate Izhod

Slika 11:Izpis rezultatov

Če kliknemo na gumb »Pobriši rezultate« spraznimo tabelo z rezultati.

13. Zaključek

Ob začetku izdelave sem si zadal cilj izdelati kviz, ki bo uporabniku postavljaj poljubno število vprašanj iz poljubne tematike. V pomoč bi pa mu naj bila slika, ki se nanaša na postavljeno vprašanje. Vse te cilje mi je uspelo realizirati. Dodal sem še možnost vnašanja novih vprašanj v tabelo, dodajanje novih tem, spreminjanje privzetih lastnosti, ter hranjenja rezultatov. Mislim da je moja raziskovalna naloga dovolj obsežna. Največja odlika te naloge je, da je uporabniku prijazna ter da si jo lahko uporabnik prilagodi po svojem okusu.

14. Viri in literatura

- B. Swart, M. Cashman, P. Gustavson in J. Hollingworth.: Borland C++ Builder Developer's Guide. (e-Knjiga). Indianapolis, Indiana, USA. SAMS. December 2002. ISBN: 0-672-32480-6.
- K. Reisdorph in K. Henderson.: Teach Yourself Borland C++ Builder in 21 Days. (e-Knjiga). Indianapolis, Indiana, USA. SAMS. Februar 1997. ISBN: 0-672-31020-1.
- A. Vugrinec.:Preveri svoje znanje, Ateja d.o.o. Ljubljana 2003.
- V. Brodnik, R. A. Jernejčič, Z. Radonjič, T. Urankar-Dornik,Zgodovina 1, DZS d.d., Ljubljana 2001.
- Prevedel J. Plešej, 1000 vprašanj in odgovorov, Editions Caramel 2000, za slovensko izdajo Mladinska knjiga Založba, Ljubljana 2003.
- P. Mrhar, Uvod v SQL, Flamingo Založba, Nova Gorica 2002.

15. Priloge

Na cd-ju:

- program Kviz

- Raziskovalna naloga

- e-knjiga: Bob Swart, Mark Cashman, Paul Gustavson in Jarrod Hollingworth:
Borland C++ Builder Developer's Guide

- e-knjiga: Kent Reisdorph in Ken Henderson: Teach Yourself Borland C++ Builder
in 21 Days

16. Zahvala

Iskreno se zahvaljujem mentorju, ki me je z napotki usmerjal pri raziskovanju.

Zahvaljujem se tudi vsem, ki so program testirali ter mi z nasveti oziroma opozorili pomagali, da sem pripeljal raziskovalno nalogo do končne oblike.