

Vysoká škola Báňská – Technická univerzita
Ostrava

Fakulta elektrotechniky a informatiky



Tvorba aplikací pro m. z. II

Semestrální projekt

Den: Pondělí

Hodina: 16:00

Filip Šabacký SAB0067

Obsah:

1.	Základní Popis mobilní aplikace.....	3
2.	Obecný popis aplikace	3
3.	Grafický návrh aplikace.....	3
4.	Návrh struktury aplikace.....	4
5.	Schéma databáze.....	5
6.	Ovládání aplikace	7
7.	Seznam otázek	10

1. ZÁKLADNÍ POPIS MOBILNÍ APLIKACE

V práci je popisovaná již výše zmíněná aplikace Kvíz pro jazyk C. Aplikace slouží pouze pro operační systém Android. Jedná se o nativní aplikaci.

2. OBECNÝ POPIS APLIKACE

Aplikace Kvíz pro jazyk C je vytvořena formou kvízu. Uživatel odpovídá na náhodně vybrané otázky z databáze. Vše probíhá tak, že se uživateli zobrazí otázka a čtyři různé odpovědi. Vždy jen jedna odpověď je správná. Počet otázek si uživatel zvolí před začátkem jednotlivé hry. Může si vybrat jednu ze čtyř úrovní (lehká - střední - těžká - legendární). Úrovně se liší počtem otázek, které dostane uživatel v jedné hře (kvízu). Lehká úroveň obsahuje pět otázek. Každá další úroveň má o pět otázek více. Legendární úroveň bude tedy obsahovat dvacet otázek.

Během hry uživatel vidí, kolik má aktuálně bodů. Body začínají vždy na hodnotě nula, za každou správnou odpověď uživatel získá deset bodů. Za špatně zodpovězenou otázku se body neodčítají. V každé hře je také vidět počet otázek, které má ještě uživatel před sebou a na kolik otázek už odpověděl. Po každé hře následuje vyhodnocení, které obsahuje počet získaných bodů a počet otázek, na které uživatel odpověděl správně.

Z vyhodnocení je možné spustit rovnou další hru, aniž by bylo nutné navštívit hlavní menu aplikace. Nová hra se pak spustí s tou úrovní, která byla nastavena v předchozím kvízu. Po každé hře se získané body (skóre) zapíšou do SQLite databáze. Body se pak dají zobrazit v žebříčku. Ten je pro každý chytrý telefon jedinečný. Díky toho funguje celá aplikace v režimu offline.

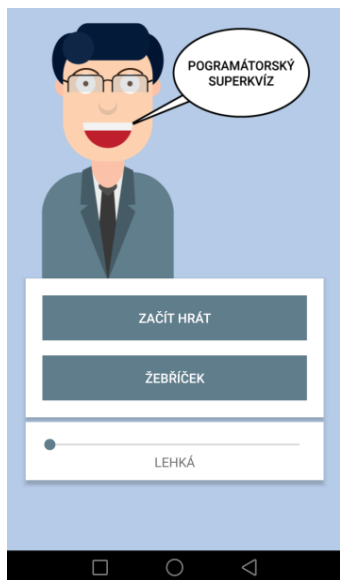
3. GRAFICKÝ NÁVRH APLIKACE

Aplikace je založena na kombinaci tří barev – šedé, bílé a modré. Tyto barvy jsou použity z důvodu, aby bylo možné nerušeně pracovat s aplikací jak v noci, tak i ve dne. Dále by pak tyto barvy, oproti jiným barevným kombinacím, měly být dobře viditelné na slunečním svitu.

V aplikaci je použit jeden hlavní obrázek. Na obrázku je muž, který pomocí komiksové bubliny říká: „Programátorský superkvíz“. Tento text je zvolen jako nadsázka, protože uživatel získává v aplikaci teoretické znalosti, které neslouží pro laiky.

Hlavní menu, které se zobrazí po spuštění aplikace, obsahuje:

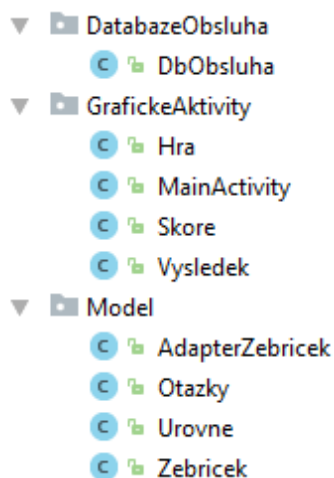
- Obrázek (logo aplikace).
- Posuvnou lištu.
 - Uživatel si pomocí posunutí může změnit úroveň hry.
- Tlačítka
 - Pro spuštění hry.
 - Pro zobrazení žebříčku.



Obr. 1 - Hlavní menu aplikace Kvíz pro jazyk C (vlastní)

4. NÁVRH STRUKTURY APLIKACE

Po grafickém návrhu aplikace bylo důležité si dobře navrhnout strukturu aplikace. Ta obsahuje celkově devět tříd.



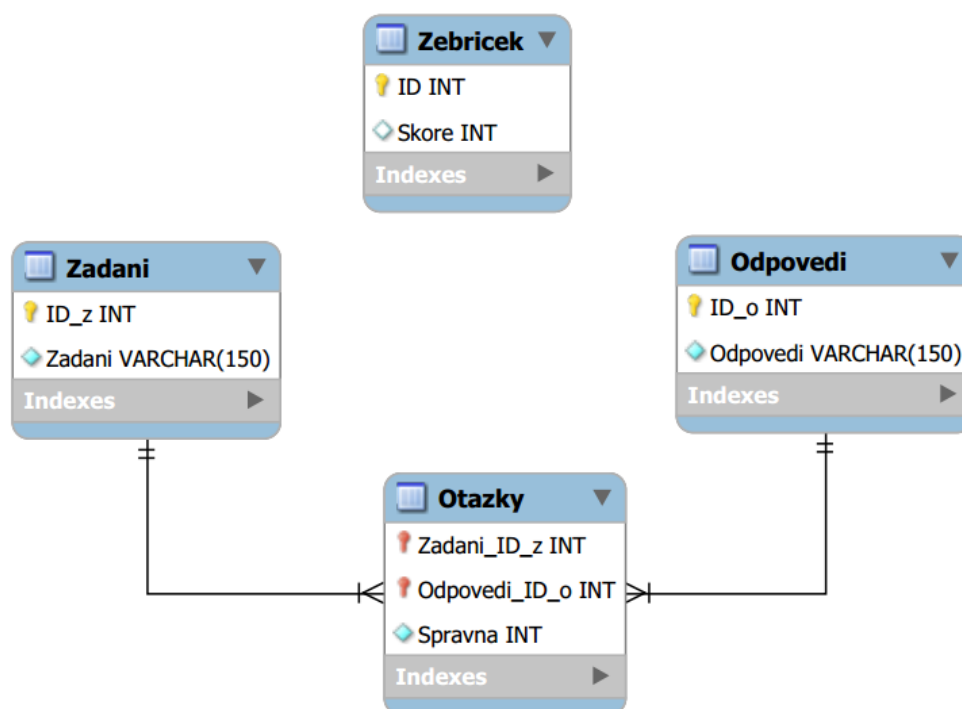
Obr. 2 - Struktura aplikace Kvíz pro jazyk C (vlastní)

Aplikace je rozdělena do tří základních částí:

1. DatabazeObsluha
 - Pomocí této třídy obsluhují SQLite databázi.
2. GrafickeAktivity
 - Patří zde čtyři třídy.
 - Každá třída se stará o jednu aktivitu.
 - Slouží nám tedy pro grafické zobrazení.
3. Model
 - Kromě třídy AdapterZebricek nemá žádná třída vliv na grafický vzhled aplikace, ale starají se pouze o funkcionalitu.

5. SCHÉMA DATABÁZE

Databáze, která je použita v aplikaci Kvíz pro jazyk C byla nejdříve vymodelována v programu MySQL Workbench. Jedná se pouze o E-R diagram, jelikož některé datové typy se v SQLite jmenují jinak.



Obr. 3 - E-R diagram databáze pro aplikaci Kvíz pro jazyk C (vlastní)

Každé zadání (ID_z) se vyskytuje v tabulce Otazky celkem čtyřikrát. Každý záznam má přiřazenou také odpověď (ID_o) z tabulky Odpovedi. To slouží k tomu, aby měl

uživatel vždy na výběr ze čtyř odpovědí. Všechny záznamy v tabulce Otazky obsahují také správnou odpověď (ID_spravna).

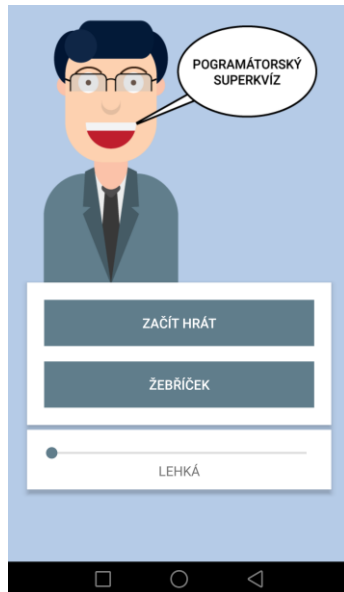
ID_z	ID_o	Spravna
Filtr	Filtr	Filtr
1	1	3
1	2	3
1	3	3
1	4	3

Obr. 4- Příklad záznamů v tabulce Otazky (vlastní)

Do tabulky Zebricek se ukládá pouze počet nasbíraných bodů v odehraných kvízech.

6. OVLÁDÁNÍ APLIKACE

A. Hlavní nabídka aplikace a žebříček



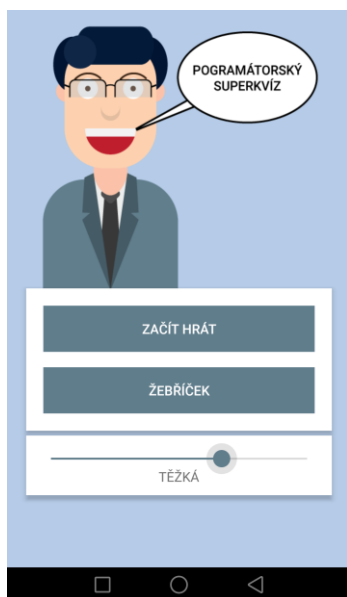
Obr. 5- Hlavní menu aplikace (vlastní)

NEJLEPŠÍ SKÓRE	
1	60,0
2	50,0
3	30,0
3	30,0
	20,0
	20,0
	20,0

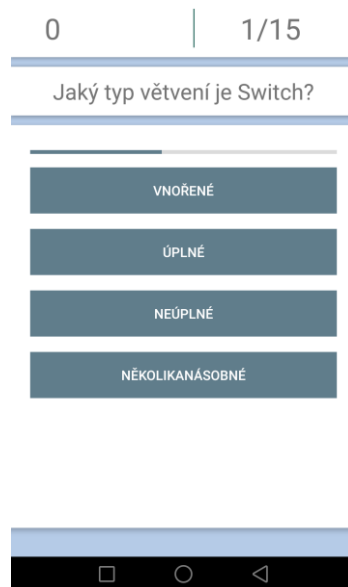
Obr. 6- Žebříček odehraných her (vlastní)

Tlačítko Žebříček nám po stisknutí zobrazí body ze všech odehraných her.

B. Hlavní nabídka aplikace a konkrétní hra (kvíz)



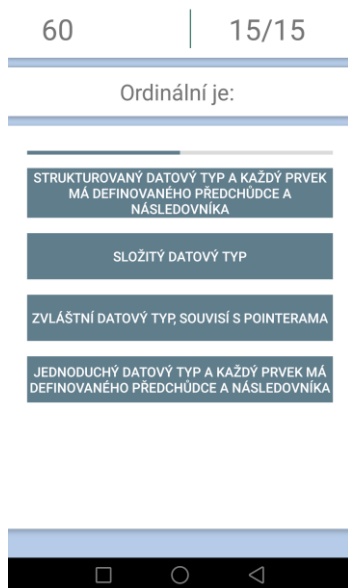
Obr. 7- Hlavní menu aplikace - výběr úrovně (vlastní)



Obr. 8- Samotná hra (vlastní)

Pomocí Seek baru si uživatel zvolí obtížnost hry. Na výběr má ze čtyř úrovní - lehká, střední, těžká a legendární. Jednotlivé úrovně se od sebe liší pouze počtem otázek, které bude muset uživatel řešit v jednom kvízu. Pak už pouze stačí kliknout na tlačítko Začít hrát a hra se spustí. U každé otázky vybírá uživatel jednu správnou odpověď. Vždy je správná pouze jedna odpověď.

C. Kvíz aplikace a vyhodnocení kvízu



Obr. 9- Samotná hra s ukazatelem bodů (vlastní)



Obr. 10- Vyhodnocení kvízu (vlastní)

Po dokončení aktuálně odehrané hry se uživateli zobrazí vyhodnocení. To obsahuje počet získaných bodů (za správnou odpověď získá uživatel 10 bodů, za špatnou se body neodečítají), počet správně zodpovězených otázek a celkový počet položených otázek. Poté si může zvolit, jestli bude hrát další hru se stejnou úrovní nebo se vrátí zpět do hlavního menu aplikace.

7. SEZNAM OTÁZEK

1. Mezi vlastnosti algoritmu nepatří:			
Elementárnost	Konečnost	Složitost	Hromadnost
2. Ukazatel se nazývá jiným slovem:			
Pointer	Makro	Struktura	Dynamická proměnná
3. Datový typ struktury:			
Heterogenní	Jednoduchý	Zvláštní	Složený
4. Mezi relační operátory patří:			
>	&&	*a	+a
5. Reálný datový typ:			
float	int	void	short
6. Elementárnost:			
Se skládá z konečného počtu kroků srozumitelné pro realizátora	Má určený další postup	Skončí v reálném čase	Je použitelná pro stejný typ úloh
7. Determinovanost:			
Se skládá z konečného počtu kroků srozumitelné pro realizátora	Musí být vždy určen další postup a každý krok musí být jednoznačně definován	Skončí v reálném čase	Je použitelná pro stejný typ úloh
8. Konečnost:			
Má přesně stanovený postup řešení	Má určený další postup	Skončí v reálném čase	Je použitelná pro stejný typ úloh

9. Hromadnost:			
Má přesně stanovený postup řešení	Má určený další postup	Skončí v reálném čase	Je použitelná pro stejný typ úloh
10. Rezultativnost:			
Má přesně stanovený postup řešení	Dává pro stejné vstupní údaje stejný výsledek	Se uskuteční v co nejkratším čase	Je použitelná pro stejný typ úloh
11. Efektivnost:			
Má přesně stanovený postup řešení	Dává pro stejné vstupní údaje stejný výsledek	Se uskuteční v co nejkratším čase	Je použitelná pro stejný typ úloh
12. Mezi možnosti zápisu algoritmu nepatří:			
Slovní	Grafické	Počítačový program	ER model
13. Algoritmus je:			
Je to postup řešení, jehož realizací získáme správný výsledek	Je to problém, který musíme vyřešit	Je to výsledek, jehož realizací získáme správný postup	Je to určité řešení, které je správné
14. Spojka se značí ve vývojovém diagramu:			
Kosočtvercem	Kruhem	Obdélníkem	Čtvercem
15. Značka přiřazení se značí ve vývojovém diagramu:			
Obdélníkem	Kruhem	Kosočtvercem	Kosodélníkem
16. Jakou příponu má zdrojový (textový) soubor v jazyce C?			
*.c	*.cc	*.cp	*.h
17. Jaký typ větvení neexistuje?			
Úplné	Neúplné	Celé	Několikanásobné

18. Jaký typ větvení je Switch?			
Úplné	Neúplné	Vnořené	Několikanásobné
19. Selektor v několikanásobném dělení musí být datového typu:			
Ordinálního	Strukturovaného	Zvláštního	Složitého
20. Mezi jaké větvení patří ternární operátor?			
Úplné	Neúplné	Vnořené	Několikanásobné
21. Jaké jsou 4 základní typy větvení?			
Úplné, neúplné, několikanásobné, vnořené	Celé, neúplné, několikanásobné, vnořené	Neúplné, úplné, složité, nesložité	Meganásobné, neúplné, celonásobné, vnořené
22. Jaké jsou 3 typy cyklů?			
For, break, while	Continue, do-while, while	For, do-while, while	If, break, while
23. Cyklus for patří mezi cykly:			
Obyčejný s podmínkou na začátku	S podmínkou na konci	Zvláštní případ s podmínkou na začátku	S podmínkou uprostřed
24. Cyklus while musí proběhnout minimálně:			
Jednou	Dvakrát	Stokrát	Nemusí proběhnout vůbec
25. Kdy použijeme cyklus for?			
Pokud známe počet opakování předem	Pokud neznáme počet opakování předem	Kdykoliv	Pokud známe hodnotu, od které má začít
26. Kdy použijeme cyklus do-while?			
Pokud známe počet opakování předem	Pokud neznáme počet opakování předem	Pokud známe hodnotu, od které má začít	Vždy, když máme plnou paměť

27. Příkaz break slouží k:			
Ukončení cyklu, i kdyby podmínka platila	Ke skoku na konec cyklu a následného opakování (cyklus se neukončí)	Resetování cyklu	Ke změně na jiný typ cyklus
28. Příkaz continue slouží k:			
Ukončení cyklu, i kdyby podmínka platila	Ke skoku na konec cyklu a následného opakování (cyklus se neukončí)	Resetování cyklu	Ke změně na jiný typ cyklus
29. Ordinální je:			
Jednoduchý datový typ a každý prvek má definovaného předchůdce a následovníka	Strukturovaný datový typ a každý prvek má definovaného předchůdce a následovníka	Zvláštní datový typ, souvisí s pointerama	Složitý datový typ
30. Mezi celočíselné datové typy nepatří:			
Short	Int	Float	Bool
31. Mezi celočíselné datové typy patří:			
Char	Float	Void	Long double
32. Výčtový datový typ:			
Struktura	Enum	Float	Int
33. Mezi reálné datové typy nepatří:			
Long double	Long	Float	Bool
34. Homogenní prvek (stejného typu) je:			
Pole	Struktura	Unie	Enum

35. Heterogenní prvek (stejného typu) je			
Unie	Pole	Řetězec	Char
36. Unie je datový typ:			
Strukturovaný	Jednoduchý	Zvláštní	Lehký
37. Soubor je datový typ:			
Strukturovaný	Jednoduchý	Zvláštní	Složený
38. U struktury se vytvoří paměťové místo pro:			
Pro všechny položky	Jen tolik, kolik zabírá největší prvek	Pro první tři položky	Pro náhodný počet prvků
39. U unie se vytvoří paměťové místo:			
Pro všechny položky	Jen tolik, kolik zabírá největší prvek	Pro první tři položky	Pro náhodný počet prvků
40. Pointer slouží pro:			
Ukládání adres proměnných a funkcí	Ukládání hodnot proměnných	Vytvoření cyklu	Adresy proměnných, nelze ho použít pro adresy funkcí
41. Globální proměnné jsou deklarovány:			
Ve funkcích	Mimo funkce (kromě funkce main, tam být můžou)	Mimo funkce (tzn. i mimo funkci main)	Pouze v konstrukturu
42. Globální proměnné:			
Jsou implicitně nulové a můžeme je použít kdekoliv v programu	Nejsou nikdy implicitně nulové	Můžeme je použít pouze ve funkci main	Jsou stejné jako lokální proměnné
43. Lokální proměnné:			
Vždy přebijí	Přebijí globální	Nikdy nepřebijí	Přebijí globální

globální proměnné	proměnné pouze ve funkci main	globální proměnné	proměnné, pokud jsou deklarovány velkými písmeny
44. V jazyce C musí být lokální proměnné deklarovány (definovány):			
Na začátku těla funkce, k níž patří	Kdekoliv ve funkci	Ve funkci nebo mimo funkci	Vždy na konci programu
45. Lokální proměnné jsou:			
Nenulové, deklarovány ve funkci nebo uvnitř bloku	Nulové, deklarovány pouze ve funkci	Nulové, deklarovány pouze uvnitř bloku	Deklarovány vždy mimo funkce a mimo bloky
46. Paměťová třída:			
Určuje, kde bude proměnná v paměti uložena, viditelnost a životnost	Určuje velikost proměnné	Odkazuje na zásobník	Kontroluje datový typ