

Gymnázium, Praha 6, Arabská 14

Obor programování



Ročníková práce

Tomáš Vondra

Procvičování matematických operací

Květen 2021

Prohlašuji, že jsem jediným autorem tohoto projektu, všechny citace jsou řádně označené a všechna použitá literatura a další zdroje jsou v práci uvedené. Tímto dle zákona 121/2000 Sb. (tzv. Autorský zákon) ve znění pozdějších předpisů uděluji bezúplatně škole Gymnázium, Praha 6, Arabská¹⁴ oprávnění k výkonu práva na rozmnožování díla (§ 13) a práva na sdělování díla veřejnosti (§ 18) na dobu časově neomezenou a bez omezení územního rozsahu.

Název ročníkového projektu: Procvičování matematických operací

Autor: Tomáš Vondra

Anotace: Cílem projektu bylo vytvořit program, díky kterému je možno se zlepšit v základních matematických operacích (násobení, dělení, sčítání, odečítání). Program je převážně určen pro žáky prvního stupně základní školy. Úlohy jsou vybrány tak, že žák páté třídy základní školy, by měl zvládnout spočítat každou zadanou úlohu, co mu program poskytne. Ostatní ročníky prvního stupně mohou přeskočit úlohy, které ještě nezvládnou spočítat. Žáci mají na výběr, jestli chtějí lehčí nebo těžší úlohy. Tyto dvě verze se liší v zadávání odpovědí.

Obsah

1	Úvod	5
2	Orientace v programu	6
2.1	Úvodní okno	6
2.2	Okno s úlohami	7
3	Verze odpovídání	9
3.1	Lehčí verze	9
3.2	Těžší verze	10
4	Jak funguje program	12
4.1	První formulář	12
4.2	Druhý formulář	13
5	Závěr	15
	Seznam obrázků	16
	Bibliografie	17

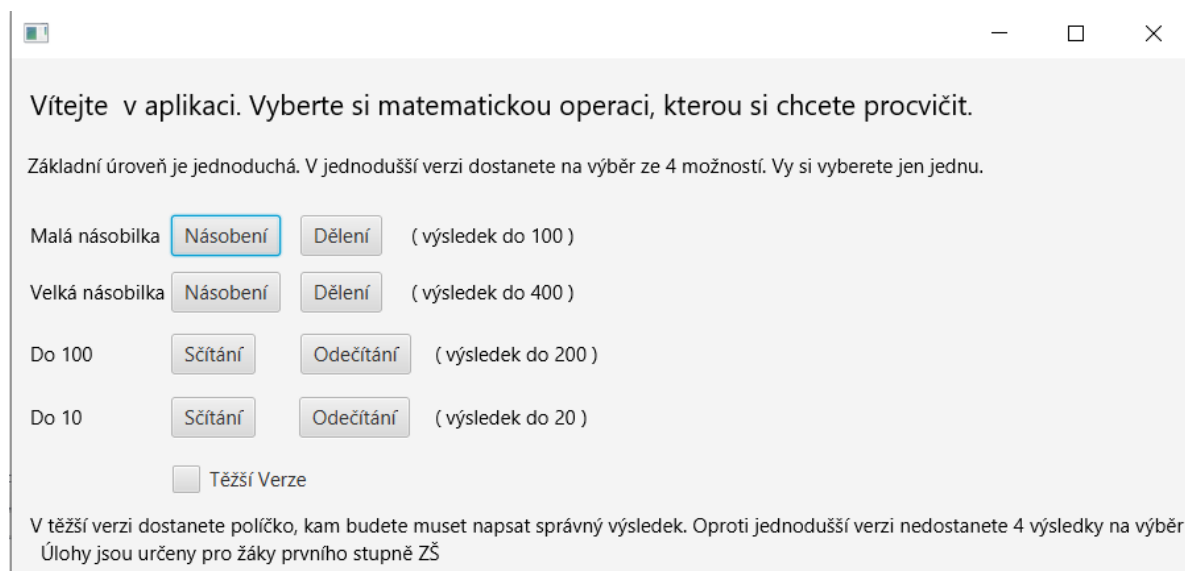
1. Úvod

Tento program by měl být cílem ke zlepšení početních operací žáků, ve kterých mají menší nebo větší nedostatky. Sami si volí typy příkladů. Žák se tedy konkretizuje na to, co mu dělá potíže. Pokud bude žák program pravidelně používat, tak si na základě předešlých výsledků může zjistit, jak moc se zlepšil (např.: jak moc se zlepšil v malé násobilce).

2. Orientace v programu

2.1 Úvodní okno

Při spuštění programu se objeví *úvodní okno*, které vás přivítá a seznámí s programem.



Obrázek 2.1: Ukázka úvodního okna

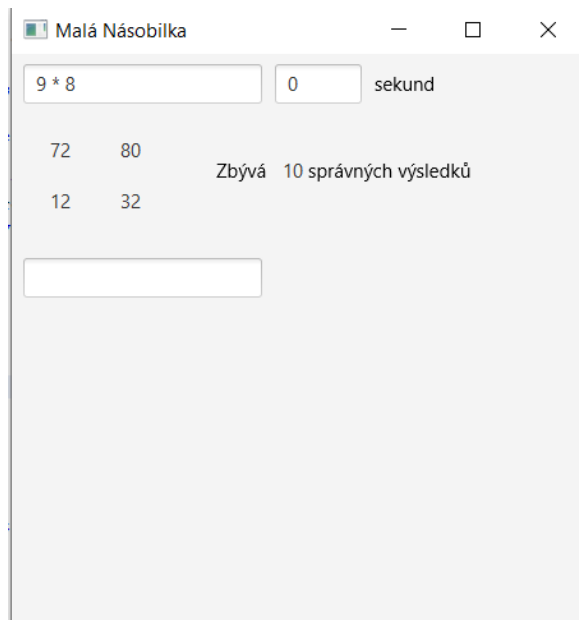
V tomto okně vidíme výběr z několika možností, které si žák může procvičit. Pokud zvolíme *Násobení* v sekci *Malá násobilka*, dostaneme úlohy, které budou mít v zadání čísla od 1 do 10, tudíž maximální možný výsledek bude 100. Maximální možný výsledek je vždy uveden vedle tlačítek.

Napravo od tlačítka *Sčítání* je uvedeno, jak velká mohou být čísla v zadání. Pro *Odečítání* jsou maximální možná čísla v zadání úlohy rovna maximálnímu možnému výsledku.

Když si žák vybere, co konkrétně chce zlepšit a je připravený spustit další okno, měl by ještě zvážit, jestli chce lehčí verzi nebo těžší verzi zadávání příkladů. Ze začátku doporučuji lehčí verzi. Když už žák lehčí verzi zvládá bez chyby, může zvolit verzi těžší. Tu nastavíme tak, že klikneme na čtvereček vedle nápisu *Těžší verze* (viz obrázek 2.1). Druhým kliknutím systém nastavíme opět na lehčí verzi. Pokud zvládá

žák i těžší verzi bez chyby, může si vyzkoušet i ostatní matematické operace, které jsou zde připravené.

2.2 Okno s úlohami



Obrázek 2.2: Ukázka okna s úlohami

Okno s úlohami se otevře, když zvolíte nějaké tlačítko z úvodního okna (např.: *Malá násobilka*).

Nahoře můžeme vidět rámeček, který poskytuje žákovi úlohy (např.: $9 * 8$). Nutno podotknout, že úlohy nikdy neobsahují nulu. Z důvodu toho, že dělit nulou se nesmí a v ostatních operacích se úloha stává příliš snadnou.

Pod rámečkem s úlohami se nachází možnost odpovědi. Prostor pro odpovědi závisí na zvolené obtížnosti (verzi). Více se o možnostech odpovídání dozvíte v kapitole 3. Nejspodnější rámeček ukáže, jestli zadaný výsledek je správně nebo ne. Pro základní potřebu je to vše.

Pak jsou zde i funkce navíc. První je *Odpočet správných výsledků*. *Odpočet* je vždy nastaven na 10. Pokud žák zadá odpověď správně, z odpočtu se odečte 1. Pokud žák odpoví špatně, číslo se nemění. Žák musí desetkrát správně odpovědět, aby "hra" skončila.

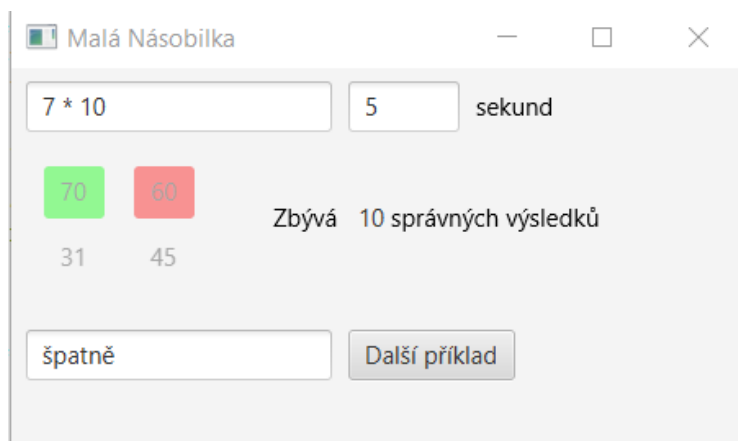
Druhá funkce navíc je *Časomíra*, která žákovi po 10 správných odpovědích ukáže, jak rychle dokázal desetkrát správně odpovědět. Spustí se hned, jakmile se otevře okno. Po každé odpovědi se *Časomíra* aktualizuje a na konci se zastaví. Díky časovači může žák porovnávat své časy, zlepšovat je a dělat "rekordy". Pro větší motivaci, jak zlepšit své výsledky, může žák porovnávat své časy se svými spolužáky, s přáteli nebo i matematickými experty. *Časomíru* najdeme nahoře vpravo vedle rámečku se zadáváním úloh. Čas se měří v sekundách.

3 Verze odpovídání

3.1 Lehčí verze

Již v minulé kapitole je zmíněno, že Lehčí a Těžší verze určuje možnosti odpovídání. V této kapitole si ukážeme, jak se tyto dvě verze liší.

Základní verze je *lehčí*. V úvodním okně můžeme verzi změnit. Pro začátek je pro žáky lepší si *těžší verzi* zvolit až poté, co budou dobře zvládat tu *lehčí*. V *lehčí verzi* má žák na výběr ze čtyř odpovědí (viz obrázek 2.2), kde jedna je správná a zbylé tři jsou špatně. Po kliknutí na nějakou z nabízených odpovědí se v dolním rámečku napíše, jestli je odpověď správně nebo špatně. Zároveň se tlačítko, na které žák klikl, rozsvítí červeně nebo zeleně (zeleně když je odpověď správná a červeně, když je odpověď špatná). Pokud se zmáčknuté tlačítko rozsvítilo červeně, tak tlačítko se správnou odpovědí zezelená. Žák se tak může poučit z chyby. Při správné odpovědi žádné políčko nezčervená.

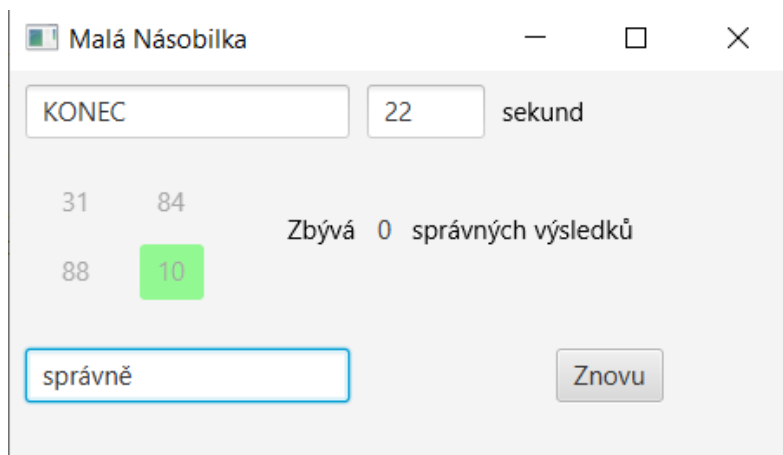


Obrázek 3.1: Špatná odpověď

Když bude žák v takové situaci, že zvolí špatnou odpověď, nebude už mít možnost ji změnit. Žák má pouze jeden pokus. Jakmile odpoví, musí pro pokračování zmáčknout tlačítko *Další příklad*. *Odpočet* se nemění. Tlačítko *Další příklad* se objevuje pouze po odeslání odpovědi. Nelze tedy úlohy přeskakovat!

Jakmile žák odpoví desetkrát správně, *Časomíra* se zastaví a v rámečku se zadáváním úloh se objeví: *KONEC*. Žákovi se nyní neobjeví tlačítko *Další příklad*, ale tlačítko *Znovu* (viz obrázek dole).

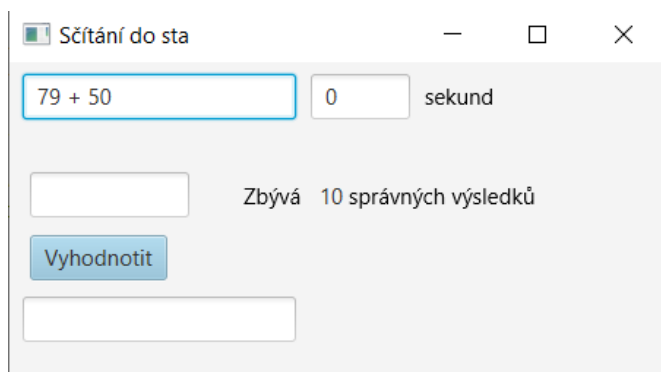
Při jeho zmáčknutí se *Časomíra* resetuje na 0 a *Odpočet správných odpovědí* se resetuje na 10.



Obrázek 3.2: Konec

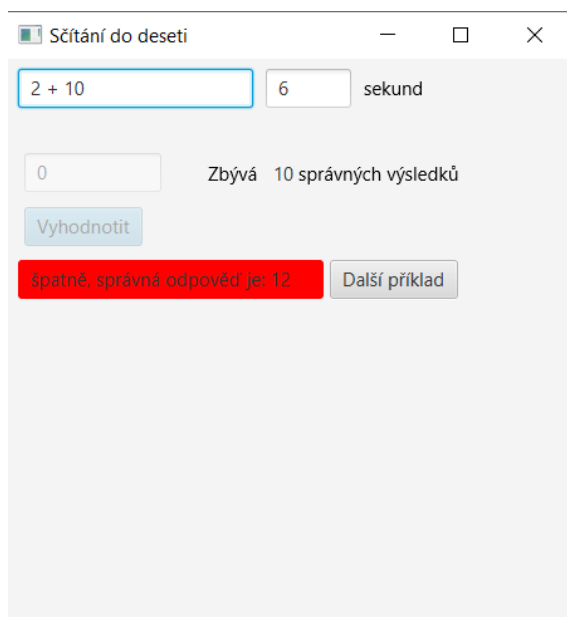
3.2 Těžší verze

Těžší verze je podobná té *lehčí*. Místo tlačítek s odpověďmi se žákovi objeví rámeček, do kterého musí napsat výsledek. Pokud žák napíše do rámečku písmena či jiné znaky než číslice, automaticky se odpověď počítá jako chybná.



Obrázek 3.3: Zadání těžší verze

Jakmile žák zadá odpověď do rámečku, musí zmáčknout tlačítko *Vyhodnotit*. Program je naprogramovaný tak, že stačí zmáčknout *enter*. Poté, co žák odpověď odešle, políčko dole mu ukáže, jestli je odpověď správná nebo špatná. Pokud je odpověď špatná, tak políčko s textem zčervená a napíše vedle správný výsledek. Pokud je odpověď správně, tak zezelená.



Sčítání do deseti

2 + 10

6 sekund

0 Zbývá 10 správných výsledků

Vyhodnotit

špatně, správná odpověď je: 12

Další příklad

Obrázek 3.4: Špatně zadaná odpověď

4 Jak program funguje

4.1 První formulář

Program je naprogramován v projektu **NetBeans** v programovacím jazyce *Java*.

Program je naprogramován na jednom formuláři, který zobrazí *úvodní okno* a na druhém, ve kterém se rozhoduje, jaká jeho část bude fungovat podle výběru matematické operace.

První formulář jsem programoval jako program, co umí pouze malou násobilku a pak jsem k němu přidal další možnosti. Malá násobilka byla původní verze programu. Program nejprve načte políčko se zadanou úlohou. Do políčka načte náhodné číslo od 1 až po hodnotu *maximalniCislo*. *MaximalniCislo* se u každé operace liší. U malé násobilky je toto číslo rovno 10. Pak udělí program název čtyř tlačítek, která se označují jako odpovědi. Program nejprve zvolí čtyři náhodná čísla od 1 až po největší možný výsledek, který je uveden v *úvodním okně* v závorce. U malé násobilky je to například číslo 100. Těmito čtyřmi čísly program sám očísluje tlačítka. Jedno tlačítko přepíše na hodnotu *správného výsledku*. Když žák zmáčkne nějaké z těchto tlačítek, program pomocí metody *equals* zjistí, jestli tlačítko obsahuje správný výsledek a pak podle toho ve spodním poli zobrazí, zda je výsledek špatně nebo správně. Následně také tlačítko obarví pomocí metody *setStyle*.

Po kliknutí na odpověď se aktualizuje čas. Proto, aby se *Časomíra* měřila v sekundách, tak jsem *duration* (doba trvání) musel vydělit 1000. *Časomíra*, spodní i dolní políčko jsou v programu nastavené jako needitovatelné, aby nikdo nemohl ovlivnit svůj výsledek.

4.2 Druhý formulář

Druhý formulář s názvem *Úvod* otevírá *úvodní okno*. Ostatním operacím je nastaveno do *Action Event*, že se otevřou v tom samém jako *Malá násobilka*. Bylo tedy nutné první formulář upravit. Z toho vyplývá, že se bude muset větvit pomocí *if*.

Nejprve jsem pro třídu *Velká násobilka násobení* definoval maximální číslo. Místo 10 to bylo 20. Tím jsem měl hotové oboje operace.

Následovně jsem definoval metodu *jdeONasobeni*, která, jak jde poznat z názvu, určila, jestli si žák zvolil větev s násobením nebo sčítáním. Rozvětvil jsem si program na to, co má dělat, když vybraná operace je násobení nebo sčítání. Pokud bylo zvoleno sčítání, tak program změnil zadání úlohy a odpovědi byly v jiném rozmezí produkovány náhodně. Část programu s násobením se ani nespustila. Sčítání tedy dostalo hodnotu *false* v metodě *jdeONasobeni* a *maximální číslo* bylo nastaveno na 100 nebo na 10, podle toho, jestli žák chtěl získávat odpovědi do 20 nebo 200.

Poté jsem ještě potřeboval získat větev, která bude používat dělení a odečítání. Nastavil jsem metodu *jdeOOpacnouOperaci*. Měl jsem tedy najednou 8 různých variant, které se žákovi mohou poskytnout (viz obrázek 2.1). Takže jsem nastavil nyní 3 různé metody, které nasměrují program správným směrem.

```
if(jdeOOpacnouOperaci){
    if (jdeONasobeni) {
        priklad.setText(randomNum * randomNum2 + " : " + randomNum);
        spravnyVysledek = randomNum2;
    } else {
        priklad.setText(randomNum + randomNum2 + " - " + randomNum);
        spravnyVysledek = randomNum2;
    }
} else {
    if (jdeONasobeni) {
        priklad.setText(randomNum + " * " + randomNum2);
        spravnyVysledek = randomNum * randomNum2;
    } else {
        priklad.setText(randomNum + " + " + randomNum2);
        spravnyVysledek = randomNum + randomNum2;
    }
}
```

Obrázek 4.1: Všechny verze

Poslední věcí bylo naprogramovat *těžší verzi*. Pro naprogramování byla jednodušší než ta *lehčí*. Do programu jsem vložil další metodu, tentokrát s názvem *jdeOTezsiVerzi*. Tato metoda program rozšířila na 16 různých možností, jak program navolit. Vše, co se mělo dít s původními 8 větvemi jsem vložil do *if* a do *else* jsem naprogramoval tuto těžší verzi. Pak jen na základě toho, jestli je to *těžší* nebo *lehčí verze* se zviditelní nebo zneviditelní různá pole a tlačítka. Do fxml souboru bylo už jen potřeba přidat *CheckBox*. Nyní byl program celý hotový.

```
NasobilkaFormController controller = loader.getController();
controller.setMaximalniCislo(100);
controller.setJdeOOpacnouOperaci(true);
controller.setJdeOTezsiVerzi(tezsiVerze.isSelected());
controller.setJdeONasobeni(false);
-
```

Obrázek 4.2: Nastavení pro odečítání do sta

5 Závěr

V závěru bych chtěl sdělit, že program splnil všechna má očekávání. V budoucnu by program mohl mít plno dalších funkcí, které by program zlepšilo nebo aby byl program pro žáky prvního stupně větší zábavou. Pro děti by tak mohlo být procvičování násobilky příjemnější. To je mým cílem.

Doufám, že program alespoň trochulepší všem žákům jejich dovednosti v oblasti matematiky. Tento program by po větších updatech mohl zlepšit celkovou úroveň matematiky žáků hravou formou. Byl bych rád, kdyby se můj program využíval na prvním stupni základní školy při procvičování příkladů v matematice.

Seznam obrázků

2.1 Ukázka úvodního okna	6
2.2 Ukázka okna s úlohami	6
3.1 Špatná odpověď	9
3.2 Konec	10
3.3 Zadání těžší verze	10
3.4 Špatně zadaná odpověď	11
4.1 Všechny verze	13
4.2 Nastavení pro odečítání do sta	14

Bibliografie

Žádná citace či literatura nebyla v dokumentu použita.