

Jak připravit žákům trenažer pro cvičení jednoduchých dovedností

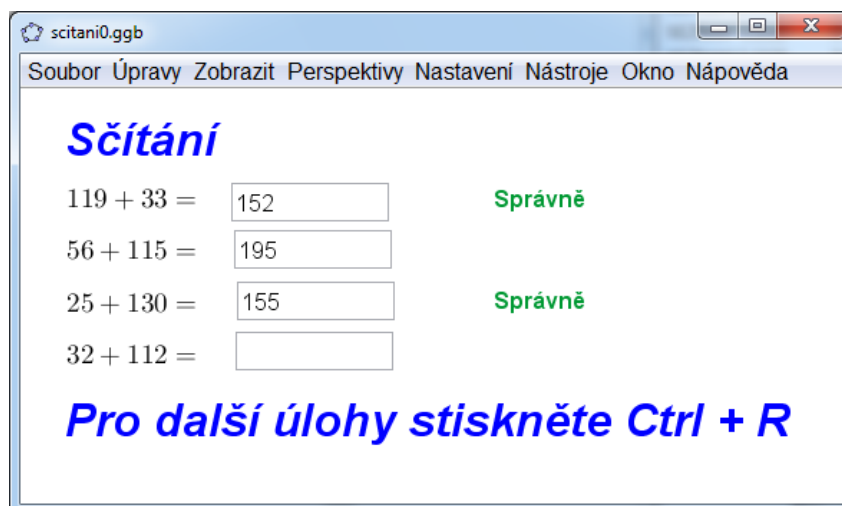
Ukázka 17 – Trenažery – Aktivní nástroje – Pole pro vkládání textu, tlačítko

Modely určené k procvičování – model první – bez skriptování

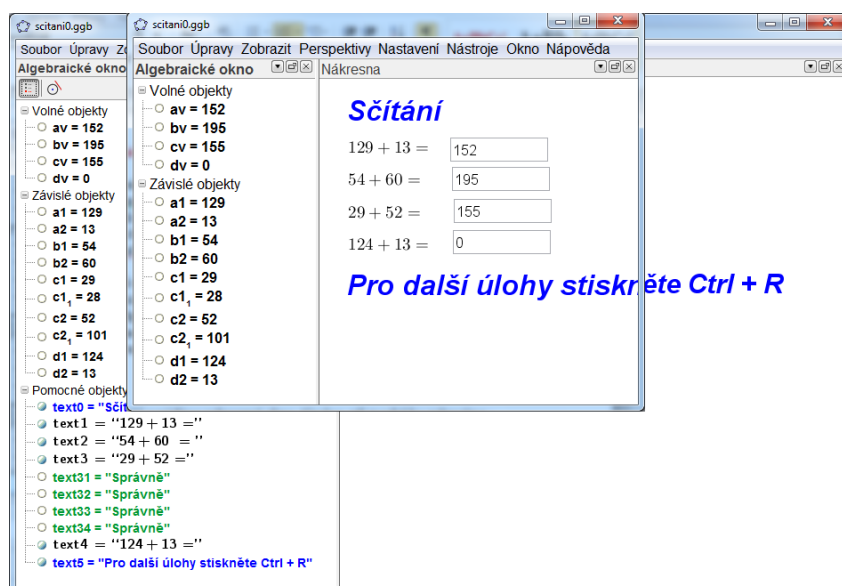
Modely, které automatizují procvičování nějaké dílčí dovednosti, mohou být užitečnou pomůckou, která nahradí sbírku úloh a šetří čas učitele, zejména pokud je model interaktivní a zobrazuje i hodnocení žákova řešení. Takové modely většinou vyžadují alespoň trochu programování, díky GeoGebře však zvládneme sestavit úvodní (velmi) jednoduchý model i zcela bez programování.

V poslední trojici ukázek vytvoříme základní model, posléze model zdokonalíme pomocí jednoduchého kódu, a nakonec vytvoříme dynamickou webovou stránku (pracovní list), v němž bude tento model vnořen. Připravený model je záměrně svým tématem velmi jednoduchý – procvičování sčítání celých čísel – a soustředíme se na nástroje GeoGebry použité k jeho sestavení.

Základní model [scitani0.ggb](#) vidíme na obrázku 17.1.



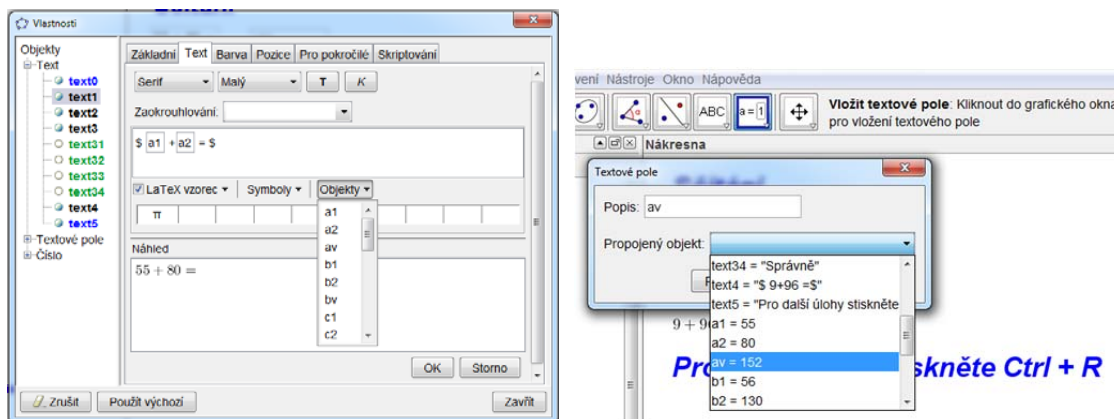
Obr. 17.1



Obr. 17.2

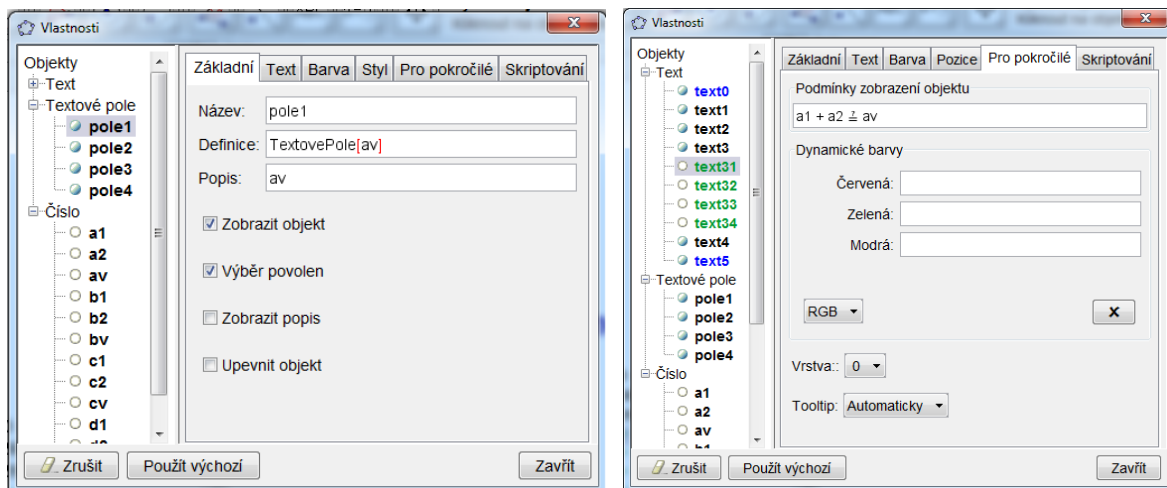
1. Ačkoliv na obrázku 17.1 nevidíme zobrazené okno *Algebra*, při tvorbě modelu ho zobrazíme, usnadní nám sledování naší práce, výběr objektů a kontrolu jejich hodnot. Skryjeme ho až před dokončením závěrečných úprav vzhledu modelu. Obrázek 17.2 ukazuje objekty v modelu, jednak bez zobrazení pomocných objektů, jednak i s nimi. Bude se nám hodit i *Vstupní pole* a *Panel Nástrojů*.

2. Do modelu vložíme číselné proměnné, nikoliv ale konstanty či posuvníky, ale výsledky příkazu, který vrací náhodné číslo:
 - $a1 = \text{NahodneMezi}[5, 150]$
3. Takových čísel potřebujeme více, opakujeme tedy příkaz postupně pro proměnné $a2, b1, b2, c1, c2, d1, d2$. To můžeme udělat různými postupy, popíšeme si dva nejrychlejší:
 - **Využití historie příkazového řádku:** Po potvrzení prvního příkazu stiskneme klávesu *šipka* (nahoru), klávesou **Enter** přijmeme nabízený poslední provedený příkaz (je-li příkazů více, můžeme šipkami listovat seznamem), změníme v něm název proměnné a potvrdíme – opět klávesou **Enter**.
 - **Kopírování:** Copy–Paste (**Ctrl + C**, **Ctrl + V** ve Windows) funguje i zde, jen musíme upravit název proměnné.
 - Uvedené postupy pak zopakujeme i při tvorbě dalších typů prvků.
4. Do *Nákresny* vložíme textové pole se zadáním prvního násobení. Do textu vložíme obsah $a1 + a2 =$, přičemž $a1$ a $a2$ do textu vložíme jako objekty, tj. vybereme je v okně *Algebra* či v seznamu objektů – viz obrázek 17.3. (Všimněme si, že pole s dynamickým textem má vlastní nastavení zaokrouhlování – při zobrazení celých čísel to však nevyužijeme.)
5. Připravíme si číselné proměnné av, bv, cv, dv pro vstup výsledků výpočtů – odpovědi žáka. Na jejich počátečních hodnotách nezáleží, takže třeba $av = 0$.
6. Do *Nákresny* vložíme aktivní prvek, který umožní zadat v textové podobě hodnotu nějakého prvku přímo z *nákresny* – *Vstupní textové pole*. Nástroj vyvolá ikona *Vložit textové pole* ze sady pro vkládání aktivních objektů (druhá sada zprava). Nastavíme jeho popis a dále propojený objekt – tím bude objekt, jehož hodnotu chceme prostřednictvím tohoto pole měnit, v našem případě tedy proměnná av .
Vložení textového pole a nastavení jeho parametrů ukazuje obrázek 17.4.
 - Popisek je důležitý, za něj lze textové pole uchopit a přesunout. V hotovém modelu jej však skryjeme – viz obrázek 17.5.



Obr. 17.3, 17.4

7. Do *Nákresny* vložíme text „Správně“, který se zobrazí pouze tehdy, pokud do vstupního textového pole zadáme správnou hodnotu – proto na jeho kartě *Pro pokročilé* okna *Vlastnosti* nastavíme *Podmínku zobrazení objektu* – do zmíněného pole zapíšeme výraz $a1 + a2 \stackrel{?}{=} av$. Vstup vidíme na obrázku 17.6. Speciální znak porovnání vložíme buď z tabulky znaků dostupné (podobně jako ve vstupním řádku) pod ikonou α vpravo v editačním poli, nebo znak alternativně zapíšeme přímo z klávesnice dvojnázkem == (dvě rovnítka).
8. Analogicky připravíme prvky pro další tři součiny.



Obr. 17.5, 17.6

9. Model doplníme o texty s nadpisy a pokyny. Protože zadané hodnoty sčítanců nejsou pevné, ale generované jako náhodné číslo, změni – přepočtou se – při přepočítání výrazů v modelu. To vyvoláme volbou *Přepočítat všechny objekty* z menu *Zobrazit* – rychleji ovšem klávesovou zkratkou **Ctrl + R**. Tento příkaz tedy v našem trenažeru způsobí zadání nové sady úloh. Zapišeme ho jako pokyn pro řešitele do *Nákresny*.
10. Upravíme prostředí modelu tak, aby byla zobrazena pouze potřebná část *Nákresny*, a soubor uložíme.

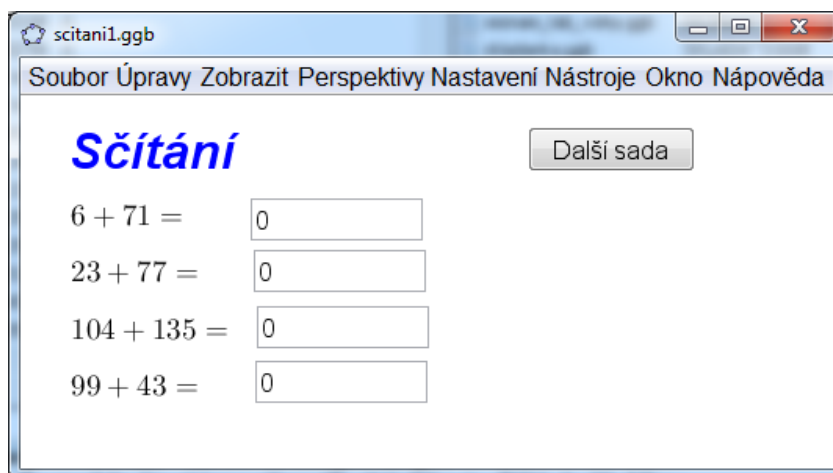
Psát žákům „stiskněte Ctrl + R“ je podivné, tlačítko by bylo lepší... Přidáme ho v dalším modelu.

Trocha skriptování

Modely určené k procvičování – model druhý – model obsahující tlačítko (a skript)

Vyjdeme z modelu, který jsme si připravili výše. Místo klávesové zkratky, kterou bychom museli žáky „naučit“, v něm budou žáci volit další sadu úloh intuitivně pomocí vloženého tlačítka.

Základní model [scitani1.ggb](#) vidíme na obrázku 17.7.



Obr. 17.7

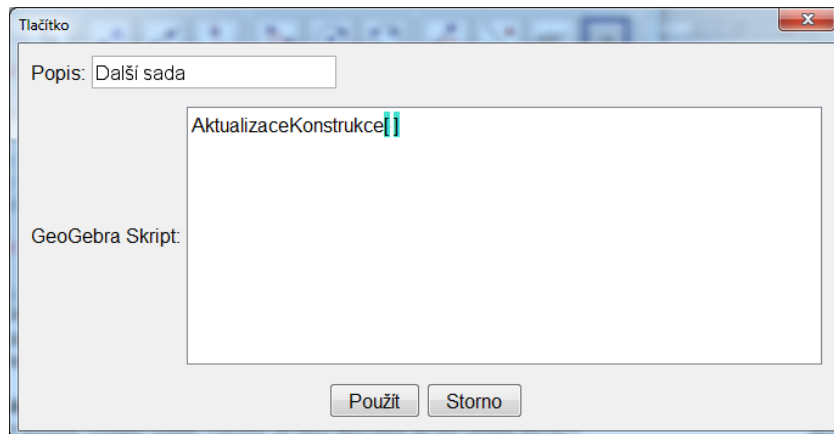
1. Volbou *Uložit jako* z menu *Soubor* uložíme model vytvořený v minulém příkladu pod novým názvem. Soubor budeme dále upravovat.
2. Odstraníme z něho text s výzvou ke stisku klávesové zkratky pro nové zadání. Dočasně zobrazíme *Panel Nástrojů* a okno *Algebra*.

Základní úprava – Tlačítko

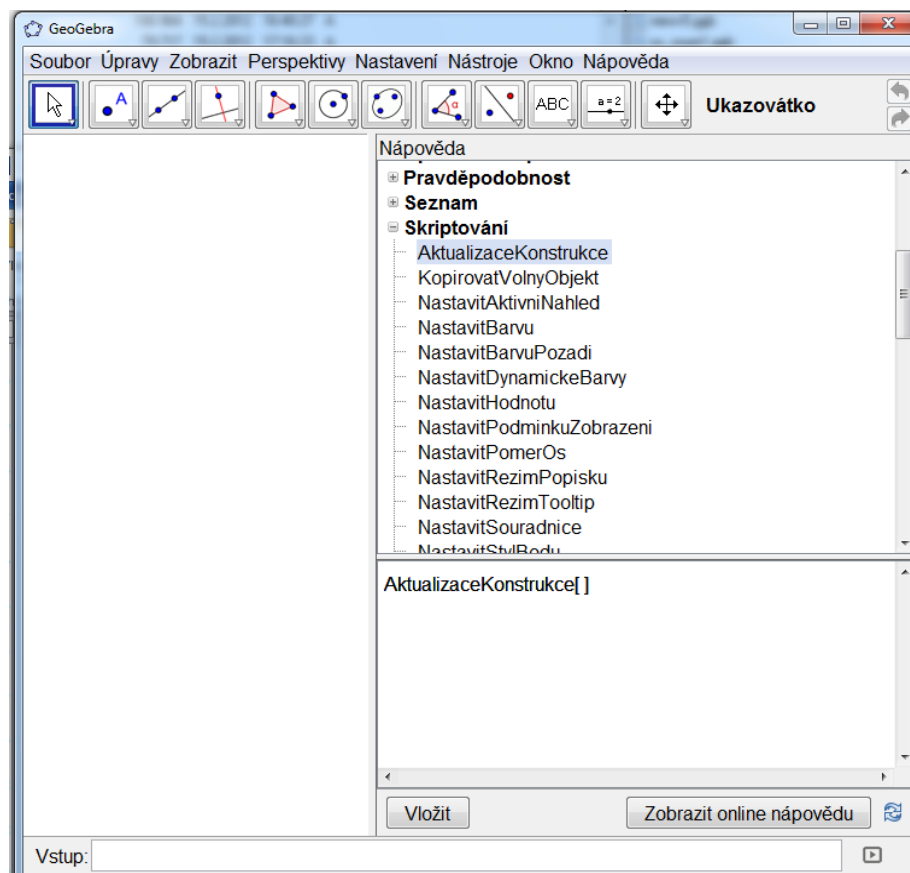
3. Do modelu vložíme objekt *Tlačítko* – ikona *Vložit tlačítko* ze sady aktivních prvků (druhá zleva). Po kliknutí do *Nápovědy* se aktivuje okno s parametry příkazu (viz obrázek 17.8).

- Do jeho pole *Popis*: napíšeme text „Další sada“.
- Do jeho pole *GeoGebra Skript*: vložíme příkaz AktualizaceKonstrukce[].

Tento příkaz – stejně jako další příkazy GeoGebra Skriptu – najdeme v seznamu příkazů (ten je přístupný pod ikonkou (šipkou) *Nápověda* vpravo dole v okně programu, jak ukazuje obrázek 17.9. Příkazy, které můžeme ve skriptu používat, jsou v seznamu příkazů vypsány ve skupině *Skriptování*.



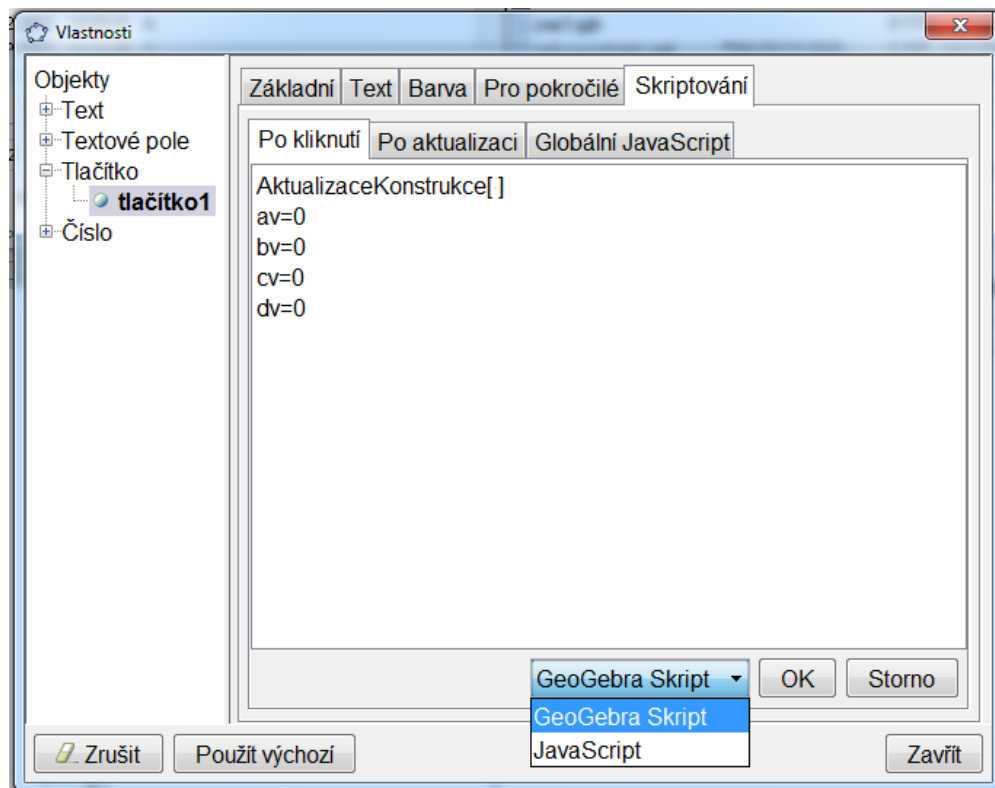
Obr. 17.8



Obr 17.9

Kód GeoGebra Skriptu může – kromě těchto speciálních příkazů – obsahovat také příkazy, které bychom mohli zapsat do příkazového pole. V naší ukázce uvedeme jen jednoduchý příklad:

Pokud po zadání další sady úloh zůstanou v textových polích hodnoty, které byly řešením minulé úlohy, nevypadá to hezky. Navíc – jakkoliv to není příliš pravděpodobné – by se mohlo stát, že nové zadání bude mít stejný součet, jako to minulé, a tudíž bude jedna úloha rovnou správně vyřešena. Proto při zadání nové sady zároveň vynulujeme pole odpovědí.



Obr. 17.10

4. Vyvoláme pole *Vlastnosti Tlačítka* a přejdeme na záložku *Skriptování*. Zde přejdeme na záložku *Po kliknutí*. Otevře se editační okno pro zápis kódu skriptu. Na obrázku 17.10 vidíme, že můžeme zvolit GeoGebra Skript nebo JavaScript. My máme zvolen GeoGebra Skript.

Pod dříve zadaný příkaz `AktualizaceKonstrukce[]` zapíšeme další čtyři:

- `av=0`
- `bv=0`
- `cv=0`
- `dv=0`

Příkazy potvrdíme (a zároveň zkontrolujeme) tlačítkem **OK**.

Zatímco příkazy GeoGebra Skriptu se vykonávají postupně příkaz po příkazu bez větvení a cyklů, JavaScript je plnohodnotný skriptovací jazyk, zde doplněný o speciální příkazy GeoGebry. Programátoři ho jistě znají. Zjednodušeně a velmi nepřesně je možné říci, že kód v GeoGebra Skriptu zapíšeme snáze, dává nám ale méně možností, než kód v JavaScriptu.

Další možnosti skriptování

Skriptování není vyhrazeno jen pro tlačítka. Skript můžeme prostřednictvím okna *Skriptování* přiřadit každému objektu. U našeho *tlačítka* máme zadán kód ošetřující událost *kliknutí na objekt*. Další záložky okna – *Po aktualizaci* a *Globální JavaScript* slouží k programování dalších situací spojených se změnou vlastností objektu, zobrazením modelu a průběhem jeho zpracování. Kód uvedený v části *Po aktualizaci* se provádí vždy hned po změně nějaké vlastnosti objektu. Kód *Po kliknutí* se hodí pro tlačítka, body, které fungují jako tlačítka, a podobně. Programování však není náplní tohoto textu a lze se o něm více dozvědět v jiných materiálech, a také v manuálech ke GeoGebře na webu [věnovaných skriptování](#).