



## Práce se seznamem

<b>Autor</b>	Václav Šimandl
<b>Oblast informatiky</b>	Algoritmizace a programování
<b>Očekávaný výstup</b>	Vytvoří přehledný program pro vyřešení konkrétního problému s ohledem na jeho možné důsledky a svou odpovědnost za ně; používá opakování, větvení programu se složenými podmínkami, proměnné, seznamy, podprogramy s parametry a návratovými hodnotami.
<b>Časová dotace</b>	30 minut
<b>Ročník</b>	2. ročník střední školy
<b>Pomůcky</b>	počítač s nainstalovaným Pythonem 3.x, vývojovým prostředím IDLE

### Cíl činnosti

Cílem aktivity je získání schopnosti pracovat se seznamy, konkrétně přidávat prvky do seznamu, odebrat je z něj a testovat přítomnost prvku v seznamu. Dalším cílem aktivity je upevnění dosavadních dovedností spojených s tvorbou a používáním podprogramů s parametry.

### Popis činnosti

1. Žák upraví podprogram tak, aby přijímal vstupní data pomocí parametru.
2. Žák zajistí opakované volání podprogramu pro každý prvek ze seznamu.
3. Žák vloží do seznamu nový prvek na základě dat získaných od uživatele.
4. Žák otestuje vytvořený program. Zjistí, co se stane po vložení duplicitního prvku do seznamu.
5. Žák zamezí vložení duplicitního prvku do seznamu.
6. Žák zajistí odebrání vybraného prvku ze seznamu.
7. Žák program důkladně otestuje a opraví případné chyby.

### Možnosti individualizace a diferenciacce

Od pokročilých žáků požadujeme, aby kruhy reprezentující jednotlivé vizitky měly náhodnou barvu z předem definované množiny barev. Od velmi pokročilých žáků vyžadujeme, aby kruhy měly zcela náhodnou barvu z barevného prostoru RGB (návod naleznou na internetu).



## Důkazy o učení

Důkazem učení je v obecné rovině vytvoření programového kódu, který je v souladu se zadáním.

Specifickým dokladem o pochopení předkládané látky, tj. správné manipulace se seznamem, je vykreslení správných vizitek studentů podle testovacího scénáře vytvořeného učitelem. Konkrétně:

- se vykreslí vizitky všech studentů, kteří byli do seznamu vloženi;
- vizitky duplicitně vkládaných studentů se vykreslí pouze jednou;
- vizitky odebraných studentů se nevykreslí.

## Metodická doporučení

V podprogramu `kresli_vizitky()` je poměrně neobvyklé předávat podprogramu `kresli_jednu()` plátno pomocí parametru a je možné, že někteří žáci nebudou tomuto přístupu rozumět. V tom případě je potřeba jim vysvětlit, že plátno je vytvářeno v podprogramu `kresli_vizitky()`. Aby bylo možné na plátno kreslit jednotlivé vizitky, je nutné použít právě předání pomocí parametru. Neočekáváme, že žáci si tento „trik“ zapamatují, a tak nedoporučujeme jej používat v písemkách ani od žáků vyžadovat, aby podobný koncept sami používali ve svých projektech. Na druhou stranu považujeme za vhodné žákům ukázat, že i plátno lze předat pomocí parametru jinému podprogramu.

Činnost je chápána jako komplexní úloha navazující na několik jednodušších úloh, na nichž žáci trénují dílčí dovednosti. Nepředpokládáme proto, že by žáci ihned po seznámení se seznamy byli schopni řešit úlohu takovéto obtížnosti. Uvedené problémy ze sekce *V čem žáci chybují* se tedy obvykle projevují již v těchto předchozích úlohách.

## Co se osvědčilo

Osvědčil se nám přístup, kdy žáci pracují zcela individuálně a v případě potřeby mohou ve dvojici či ad-hoc vzniklé trojici či čtveřici o řešení diskutovat. Učitel cyklicky prochází třídu, nahlíží do monitorů žákovských počítačů. Pokud se v řešení žáka objeví chyba a žák ještě pracuje na části úlohy, v níž má chybu, na chybu jej neupozorňujeme – zjistí ji, až program spustí a vloží do / odebere ze seznamu několik prvků a nechá si vykreslit vizitky studentů. Jestliže žák už evidentně pracuje na další části úlohy, doporučujeme jej na chybu upozornit, ale důvod chyby neprozrazovat. Místo toho je vhodnější žáka nechat program spustit, nadiktovat mu testovací scénář (tj. říci, jaké má do seznamu přidat prvky, popř. jaké z něj má odebrat) a upozornit jej, že vykreslený obrázek není v souladu s očekávaným stavem. V případě, že program není optimální, je potřeba žákovi na vhodném příkladu vysvětlit, proč by v jiné situaci tento program neobstál.



Dále se nám osvědčilo požadovat, aby nám žáci ukázali již hotovou úlohu. Při této závěrečné kontrole zkoumáme zdrojový kód, zda je v pořádku. Dále vyzveme žáka, aby program spustil, a nadiktujeme mu testovací scénář, který důkladně prověří vytvořený program. Při této kontrole snadno můžeme odhalit chyby, kterých jsme si při čtení zdrojového kódu nevšimli.

## Na co se zaměřit

Při průběžné kontrole žákovských řešení doporučujeme se zaměřit na problémy, které uvádíme v sekci *V čem studenti chybují*.

## V čem studenti chybují

- Žáci se domnívají, že se plátno s vizitkami studentů zobrazí ihned po spuštění programu. To není pravda, zobrazí se až po explicitním zavolání podprogramu `kresli_vizitky` z konzole (interaktivního okna).
- Žáci nerozumí pokynu v bodě 2, tj. příjmu jména vykreslovaného studenta pomocí parametru a úpravě volání daného podprogramu.
- Žáci ve snaze vykreslit pro každého studenta ze seznamu jednu vizitku nepoužijí cyklus, ale volají podprogram pro každé jméno ze seznamu „ručně“.
- Žáci ve snaze vykreslit pro každého studenta ze seznamu jednu vizitku upravují podprogram `kresli_jednu` namísto podprogramu `kresli_vizitky` (ač je postup v zadání explicitně uveden).
- Žáci jméno přidávaného studenta nezískávají od uživatele pomocí standardního vstupu, ale přidávají určité jedno jméno zadané „natvrdo“.
- Žáci neotestují reakci programu v případě vložení duplicitního prvku. Následkem toho ignorují bod 6 s tím, že v seznamu žádné duplicitní prvky nemají.
- Žáci sice při přidávání duplicitního studenta vypíší hlášení, že se tento student v seznamu nachází, ale stejně jej do seznamu přidají.
- Žáci při snaze odebrat studenta ze seznamu nekontrolují, zda se daný student v seznamu nachází. Kvůli tomu může program (v závislosti na zvolené metodě odebrání prvku) končit chybou `ValueError: 'Alenka' is not in list`.



## Zdroje a použité materiály

1] BLAHO, A., SALANCI, L. ŠIMANDL, V. *Programování v jazyce Python pro střední školy. Učebnice.* České Budějovice: Jihočeská univerzita v Českých Budějovicích, 2020. ISBN 978-80-7394-784-2. Dostupné z: <https://imysleni.cz/ucebnice/zaklady-programovani-v-jazyce-python-pro-stredni-skoly>

## Přílohy a pracovní listy

- Pracovní list: seznamy-vizitky-pl.pdf
- Zdrojový kód: seznamy-vizitky-zadani.py
- Vzorové řešení: seznamy-vizitky-reseni.py