



## Práce s podprogramem s návratovou hodnotou

<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 větvení programu se složenými podmínkami, proměnné, 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 dovedností zaměřených na práci s podprogramy s parametry a návratovými hodnotami. Přínosem aktivity je porozumění rozdílu mezi výpisem výsledku v podprogramu a jeho vrácením pomocí návratové hodnoty.

### Popis činnosti

1. Žák nalezne na internetu aktuální výši DPH v základní i snížené sazbě (v případě dvou snížených sazeb si vybere jednu z nich).
2. Žák vytvoří podprogram se dvěma parametry, který vypočítá cenu zboží včetně DPH v základní sazbě. Výsledek výpočtu vypíše do konzole.
3. Žák zajistí volání podprogramu v programu. Přitom využije již připravené proměnné jako skutečné parametry volaného podprogramu.
4. Žák upraví podprogram, aby byl schopen vypočítat cenu zboží včetně DPH ve správné sazbě.
5. Žák program otestuje a opraví případné chyby.
6. Žák upraví podprogram, aby výsledek výpočtu nevypisoval do konzole, ale vrátil pomocí návratové hodnoty.
7. Žák upraví volání podprogramu, aby byla hodnota vrácená podprogramem uložena do proměnné a následně vypsána.
8. Žák program otestuje a opraví případné chyby.
9. Žák zavolá podprogram i pro druhý typ zboží. Kód pro získání vstupních dat o tomto zboží je připraven a stačí jej odkomentovat. Hodnotu vrácenou podprogramem uloží do proměnné.
10. Žák sečte hodnoty proměnných vyjadřující cenu prvního typu zboží s DPH a druhého typu zboží s DPH. Tuto celkovou hodnotu vypíše do konzole.
11. Žák program důkladně otestuje a opraví případné chyby.



## Možnosti individualizace a diferenciacce

Od pokročilých žáků požadujeme, aby bylo možno určit celkovou cenu za libovolný předem daný počet typů zboží. Očekáváme od nich, že využijí `for` cyklus. V rámci jedné jeho iterace si vyžádají vstupní data o ceně zboží bez DPH a sazbě DPH daného zboží, zavolají podprogram a přičtou jím vrácenou hodnotu do proměnné `celkem`. Na závěr vypíší celkovou cenu do konzole.

## 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. práce s parametry a návratovými hodnotami podprogramů, je vypsání správné celkové ceny zboží podle testovacího scénáře vytvořeného učitelem.

## Metodická doporučení

Činnost je chápána jako komplexní úloha navazující na sadu úloh, na nichž žáci trénují dílčí dovednosti. Nepředpokládáme proto, že by žáci ihned po seznámení s podprogramy s parametry a návratovými hodnotami byli schopni řešit úlohu takovéto obtížnosti. Každý z bodů 2 až 6 uvedených v sekci Popis činnosti by měl být trénován na jedné či několika předchozích úlohách. Některé 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. 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 a upozornit jej, že vypsáný výsledek neodpovídá očekávané hodnotě. V případě, že program není optimální, je potřeba žákovi na vhodném příkladu vysvětlit, proč by v jiné situaci tento program neobstál. Toto **neplatí** pro chyby uvedené ve 4. a 7. odrážce v sekci V čem žáci chybují – jde o chyby v návrhu řešení úlohy. Tyto chyby se projeví při řešení dalších částí úlohy. Tehdy by si měl žák sám uvědomit, že jím zvolený způsob řešení není dostatečně robustní a je potřeba řešení přepracovat.

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

- Žák nesprávně vytváří podprogram až na konci programového kódu. Kvůli tomu při snaze podprogram zavolat program končí chybou, že podprogram ještě není definován.
- Žák není schopen sestavit vzorec pro výpočet ceny včetně DPH v základní sazbě.
- Žák si pro uchování výše DPH v základní, resp. snížené sazbě nevytvoří vhodné proměnné, ale píše hodnoty přímo do vzorce pro výpočet. Toto není chyba, nicméně je vhodné vést žáky z důvodu přehlednosti k tomu, aby podobné údaje ukládali do proměnných, kde je v případě potřeby bude snadné nalézt a upravit.
- Žák v podprogramu nevyužije hodnotu prvního parametru obsahující cenu zboží, ale použije (globální) proměnnou `hodnotaZbozi1`. Obdobně později nevyužije hodnotu druhého parametru, ale užije (globální) proměnnou `snizenaSazba1`. Ačkoliv bude v tuto chvíli program fungovat, je to **závažná chyba (miskoncept)**.
- Žák není schopen vytvořit volání vytvořeného podprogramu na základě již existujícího kódu.
- Žák při snaze rozlišit výpočet pro základní, resp. sníženou sazbu DPH neumí vytvořit podmínku, ve které by vyhodnocoval logickou hodnotu (`True / False`) druhého parametru. Místo toho předpokládá, že parametr obsahuje textový řetězec `"ano" / "ne"` (resp. se snaží program upravit, aby tomu tak bylo).
- Žák ignoruje požadavek, aby podprogram vypočítanou cenu včetně DPH vrátil, a nadále ji uvnitř podprogramu vypisuje.
- Žák nerozumí požadavku, že má pro výpočet ceny druhého typu zboží použít tentýž podprogram jako pro výpočet ceny prvního typu zboží. Místo toho vytváří druhý podprogram.
- Žák není schopen provázat (sečíst) hodnoty vrácené podprogramem při výpočtu ceny prvního a druhého typu zboží.
- Žák není schopen program řádně otestovat. Typicky jako cenu zboží volí 1 Kč, resp. 100 Kč, což nevede k odhalení některých chyb.



## 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: podprogramy-dph-pl.pdf
- Zdrojový kód: podprogramy-dph-zadani.py
- Vzorové řešení: podprogramy-dph-reseni.py