

Cvičení 0: Technické okénko

Fialové rámečky na začátku každého cvičení obsahují věci, které je bezpodmínečně nutné znát ještě před začátkem cvičení. Neslouží jako opakování přednášky, avšak jsou vám ku pomoci při přípravě na cvičení.

Před nultým, technickým cvičením je zapotřebí znát:

- ▶ svoje učo (univerzitní číslo osoby, správnost si ověříte přihlášením do ISu);
- ▶ svoje fakultní přihlašovací jméno (tzv. *xlogin*) a fakultní heslo (to je jiné než do ISu). Tyto přihlašovací údaje si můžete ověřit na <https://fadmin.fi.muni.cz/auth>. Svůj *xlogin* zjistíte a heslo můžete změnit na https://is.muni.cz/auth/system/heslo_fi.

Oranžové rámečky obsahují upozornění a varování.

Jindřiška varuje: Na cvičení je normální se připravovat. Nebudete-li s pojmy ve fialových rámečcích srozuměni, pravděpodobně si ze cvičení kromě pocitu neúspěchu příliš neodnesete a hrozí vám vyloučení ze cvičení.

Doufáme, že v semestru podzim 2021 bude probíhat výuka (alespoň cvičení) prezenčně. Bohužel to ale není jisté a je možné, že se v nějakém okamžiku bude třeba přesunout online na distanční výuku. Věnujte proto pozornost jak fungování fakultních systémů (počítačů v učebnách, studentského serveru Aisa), tak nastavení svých strojů. Mít vše rozjeté na vašich strojích se vám bude hodit nejen pokud bychom se museli uchýlit k distanční výuce, ale jistě to oceníte i při řešení domácích úkolů.

Nulté cvičení je velmi nestandardní. Probíhá pro všechny studenty v prvním týdnu (nejde tedy o 14denní blok) a slouží především k nastavení programů nutných pro fungování v tomto předmětu. Cvičení trvá necelou hodinu a nezabývá se náplní předmětu, ale má za úkol vás seznámit s počítačovou učebnou B130 a jejím programovým vybavením, aby se na prvním cvičení nemusely řešit problémy technického rázu. Čas na něm můžete také využít ke konzultaci problémů s instalací na vašich strojích – později na to již na cvičeních čas nebude, takže se budete muset spokojit s konzultacemi na diskusním fóru.

Vysvětlivky

I nultá kapitola sbírky je trochu zvláštní – slouží totiž zároveň jako seznámení se sbírkou samotnou. Jednou z věcí zasluhujících vysvětlení je význam piktogramů objevujících se u některých příkladů:

»= Symbolem »= jsou označeny ty příklady, které by se měly stihnout na cvičení. Příklady bez tohoto symbolu tak slouží spíše pro domácí studium a přípravu. Mimochodem, přesný význam tohoto symbolu se dozvíte případně na navazujícím předmětu IB016 Seminář z funkcionálního programování. Pokud mu chcete nějak říkat, můžete používat termín *bind*. Ještě více mimochodem, podobnost s logem Haskellu není náhodná.

✎ Tužkou (✎) jsou označeny příklady, které byste měli být schopni vyřešit správně s použitím pouze tužky a papíru, stejně jako se to po vás bude chtít u zkoušky.¹ Interpret doporučujeme použít až ke kontrole.

💎 Démantem (💎) označujeme příklady, které sice nejsou v plánu cvičení, ale přesto je velmi doporučujeme vyřešit, neboť je považujeme za zajímavé či zvláště přínosné.

¹Nepodlehnete však iluzi, že před zkouškou stačí projít tužkou označené příklady a zelenou známku máte v kapse. Piktogram totiž nijak nesouvisí s typem příkladů objevujících se u zkoušky.

Očekáváme, že tyto příklady budete řešit pravidelně v průběhu semestru a mohou vám pomoci při řešení odpovídajících domácích úkolů.



Mírně obtížnější, ale o to poutavější příklady pak nesou hvězdičku (☆) či několik hvězdiček podle obtížnosti či pracnosti. Trojhvězdičkové příklady mohou sahat i za rámec tohoto předmětu.



K vybraným příkladům existují **videa** demonstrující postup řešení příkladu. Tato videa v distanční výuce nahrazují „živé“ předvádění úlohy cvičícím, v prezenční jsou pak alternativou či slouží k připomenutí.

Etudy

Každá kapitola na začátku obsahuje několik základních příkladů, tzv. etud. Očekává se, že tyto příklady si vyřešíte ještě *před* samotným cvičením. Slouží především k ozkoušení syntaxe jazyka Haskell, případně připomenutí jeho standardních funkcí – tedy přibližně to, co po vás chce fialový rámeček na začátku.

Etudy doporučujeme vyřešit před otevřením příslušného *předcvikového* odpovědníku (který existuje ke všem kapitolám kromě této).

Zelené rámečky obsahují rady, doporučení a tipy na další zdroje ke studiu.

Pan Fešák doporučuje: K většině příkladů se na konci sbírky (případně kapitoly, čtete-li tyto samostatně) nachází řešení. Podporuje-li váš prohlížeč dokumentů odkazy, můžete klepnutím na číslo příkladu na jeho řešení skočit.

Etuda 0.η.0 Zkuste si to!

Etuda 0.η.1 Seznamte se s nultou kapitolou sbírky a s organizačními pokyny k předmětu v [interaktivní osnově](#).

Etuda 0.η.2 Stáhněte si záznam úvodního democvičení. Můžete k tomu použít odkaz pod číslem tohoto příkladu; vede do studijních materiálů předmětu v ISu.



*
**

Po etudách následují příklady určené k řešení na cvičení nebo samostatně po cvičení, například jako příprava na *pocvikový* odpovědník. Nic vám ale samozřejmě nebrání se s příklady seznámit ještě před cvičením.

0.1 Základní potřebné programy a nastavení

Je v podstatě jedno, zda budete na cvičeních používat svůj počítač nebo školní. Je však silně doporučeno abyste měli vše potřebné nainstalované u sebe (kvůli přípravám a domácím úkolům), ale zároveň je i vhodné, abyste uměli používat školní počítače.

Př. 0.1.1 **Základní využívání školních počítačů.** Přihlaste se na školní počítač (pomocí svého xloginu a fakulního hesla). Po přihlášení spusťte terminál (buď jej vyhledejte, nebo pomocí klávesové zkratky `Ctrl+Alt+T`). Po spuštění by měl terminál pracovat ve vašem domovském adresáři (`/home/xLOGIN`). Příkazem `pwd` si vypište aktuální adresář.



Př. 0.1.2 **Nastavení vlastního počítače.** Podle instrukcí v [interaktivní osnově](#) osnově předmětu si nainstalujte GHC na svůj počítač.

»=

Př. 0.1.3 Vytvořte si ve svém domovském adresáři adresář `ib015`. Můžete využít terminál nebo grafický správce souborů.

»=

Př. 0.1.4 Většinu úkolů budete vypracovávat v textovém editoru podle svého výběru. Proto si jej nastavte podle instrukcí v interaktivní osnově předmětu.

»=

Jindřiška varuje: Word ani LibreOffice Writer **nejsou** editory čistého textu.

Př. 0.1.5 V adresáři `ib015` vytvořte nový soubor `Sem0.hs` s následujícím obsahem:

»=

```
hello = "Hello, world"
```

0.2 Používání GHCi

Pokud si kapitolu procházíte samostatně, tak v této a následující sekci již přijde vhod democvičení ([intro](#)), je proto dobrý nápad jej nyní zhlédnout. Může být přínosné si nahrávku zastavovat a postupovat podle ní, případně se k nejasným částem vracet.

Př. 0.2.1 V terminálu v adresáři `ib015` si příkazem `ghci` spustíte interpret Haskellu.

»=

Komu zrovna zadáváte text, poznáte podle tzv. výzvy, neboli promptu. Prompt GHCi má obvykle tvar `Prelude>`, `Main*>` nebo `ghci>` – sem zadáváte výrazy jazyka Haskell a příkazy interpretu (ty, co začínají dvojtečkou). Prompt terminálu nabývá rozličných podob, ale obvykle končí znakem dolaru (např. `login@počítač:~/ib015 $`). Sem zadáváte např. příkazy pro práci se soubory a pohyb po adresářích.

Př. 0.2.2 V interpretu si zkuste vyhodnotit jednoduché aritmetické výrazy, tedy využít ho jako kalkulačku.

»=

Př. 0.2.3 Do GHCi načtete soubor `Sem0.hs` a nechte si vypsat konstantu `hello`. Následně si od interpretu vyžádejte její typ.

»=

Př. 0.2.4 Do souboru `Sem0.hs` přidejte konstantu `my_number` typu `Integer` a nastavte ji na svou oblíbenou hodnotu. Po uložení souboru jej znovu načtete do GHCi a konstantu vypište. Interpret následně ukončete.

»=

Pan Fešák doporučuje: Pokud se někdy program nechová tak, jak očekáváte, ačkoli jste si jisti, že tentokrát už jste ho určitě opravili, zkontrolujte, že jste soubor uložili a znovu načteli do interpretu. Je to jedna z nejčastějších chyb.

0.3 Vzdálené připojení na fakultní počítače

Používání vzdáleného připojení přes SSH není v tomto předmětu nezbytně nutné, ale může se vám hodit a určitě jej budete potřebovat jinde. Pan Fešák proto doporučuje se ho naučit již nyní. Vzdálené připojení vám umožní používat GHCi a další nástroje na počítačích na FI a dostat se k vašim kódům ze cvičení odkudkoli.

Na školní stroje se systémem Linux se lze připojit pomocí SSH. Na Linuxu či macOS se obvykle jako klient SSH používá příkaz `ssh`. Pokud klient nemáte, mělo by být možné jej nainstalovat z balíčků (bude se nejspíš jmenovat `openssh-client` nebo `openssh`). Na Windows 10 je příkaz `ssh` k dispozici v PowerShellu. Záložní možností je pak klient Putty.

K přihlašování z vnějšku fakulty slouží server aisa.fi.muni.cz (ze sítě FI dostupný i jako `aisa`). Přes tento počítač se pak případně dá dostat i k jiným, ale to nebývá potřeba, protože váš domovský adresář je sdílen přes všechny fakultní linuxové stroje.

Co se kde děje? Při práci se vzdáleným počítačem může být trochu nepořádek v tom, co se děje u vás a co na vzdáleném stroji. Měli byste to být schopni poznat podle promptu, tedy textu, který příkazová řádka vypisuje před místem, kam píšete příkaz. Na Aise byste měli (pokud jste si to nezměnili) vidět něco jako `aisa:/home/xLOGIN>$`. V promptu je tedy vidět i cesta k adresáři, v němž se aktuálně nacházíte.

Aby v tom nebyl nepořádek v našich příkladech, používáme konvenci, že příkazy spouštěné na Aise začínají `(aisa)$`, zatímco ty spouštěné u vás, začínají `(local)$` (pokud je to jedno, bude tam jen `$`). Ani jeden z těchto řetězců však nepatří na příkazovou řádku – slouží jen k označení příkazů, které můžete spustit.

Základní příkazy příkazové řádky

Tyto příkazy by měly fungovat v podstatě na všech platformách, včetně Windows (jejich přesný výstup v případě, že něco vypisují, se však může lišit). Na Windows však může být nutné použít PowerShell místo příkazové řádky (neboli `cmd`).

`ls` vypíše obsah aktuálního adresáře. Případně `ls -A` vypíše adresář včetně souborů a složek, jejichž jméno začíná tečkou.

`mkdir` slouží k vytvoření adresáře. Argumentem je cesta k adresáři, který chcete vytvořit: `mkdir ib015`. Pokud je potřeba vyrobit mezilehlé adresáře, lze použít `mkdir -p longer/path/to/ib015`.

`cd` přepíná adresáře. Jediný argument je cesta k adresáři, kam přepnout. Např. `cd ib015` pro přepnutí do adresáře `ib015`, `cd ..` pro přechod nahoru v adresářové struktuře nebo `cd ~` (vlnovka) pro skok do domovského adresáře (mimo Windows i jen `cd`).

`ssh` slouží pro vzdálený terminálový přístup na počítač (typicky se systémem Linux/Unix). Argumentem je adresa počítače a případně login: `ssh xLOGIN@aisa.fi.muni.cz` (s vaším loginem). Při prvním přihlášení je třeba potvrdit klíč stroje. Heslo se při zadávání nezobrazuje. Pokud je váš lokální login stejný, jako ten, do kterého se chcete přihlásit, nemusíte jej uvádět: `ssh aisa.fi.muni.cz` (také po vyřešení 0.3.3 níže).

`scp` slouží ke kopírování souborů mezi počítači. První argument je vždy zdroj, druhý cíl. Jedna z cest je lokální, jedna může být ve tvaru `login@stroj:cesta` a pak představuje cestu na daném stroji. Cílem může být i složka; cesta by pak měla končit lomítkem a složka musí na daném stroji existovat. Např. `scp Foo.hs xLOGIN@aisa.fi.muni.cz:` pro kopírování souboru `Foo.hs` do domovského adresáře na Aise (relativní cesty jsou vzhledem k domovskému adresáři). Aktuální adresář je označen „.“ (tečka), má-li být cílem ten.

`exit` ukončení příkazové řádky nebo sezení SSH. Můžete použít i `Ctrl` + `D`.

`cp` jako `scp`, ale jen lokálně.

`mv` přesun/přejmenování souboru, argumenty jako u `cp`.

`rm/rmdir` maže soubory, respektive prázdné složky zadané jako argumenty.

`nano` jednoduchý editor textu v terminálu (z Aisy nemůžete spustit grafický editor). Příkazy pro ovládání se zobrazují dole, `^` znamená `Ctrl`. Na vašem počítači možná nebude.

Nápovědu k příkazům a jejich volbám si vyžádejte příkazem `man` nebo volbou `--help`:

```
$ man rm
$ cd --help
```

Př. 0.3.1 Přihlaste se ze svého nebo školního počítače k serveru `aisa.fi.muni.cz`. Tam přidejte modul s novým GHC pomocí příkazu `module add ghc`². Ověřte, že máte dostatečně novou verzi GHCi (alespoň 9.0.1).

Př. 0.3.2 Pomocí příkazu `scp` si na svůj počítač zkopírujte z Aisy zdrojový kód ze cvičení. U sebe jej upravte a pošlete zpět na Aisu. Pokud jste pracovali na svém počítači, udělejte to obráceně – kód zkopírujte na Aisu, tam ho upravte (třeba pomocí `nano`) a zkopírujte zpět.

Alternativně můžete vyzkoušet některý z klikacích nástrojů: na Windows WinSCP, v Linuxových grafických správčích souborů hledejte volbu „Připojit k serveru“ nebo podobnou.

Př. 0.3.3 Nastavte si klient SSH tak, aby pro připojení k Aise stačilo zadat `ssh aisa`. Stejnou zkratku je pak možné využívat při kopírování souborů mezi počítači: `scp Sem0.hs aisa:ib015/`.



Pan Fešák doporučuje: Než navštívíte svůj oblíbený vyhledávač, vzpomeňte si na příkaz `man`. Zde se bude hodit `man ssh_config`.

Př. 0.3.4 Tento krok je určen pokročilejším uživatelům a sahá daleko nad rámec předmětu.



Pokud nedisponujete dvojicí klíčů SSH pro asymetrické šifrování, vytvořte si ji. Přidejte na Aise svůj veřejný klíč mezi autorizované, abyste při přihlašování ze svého počítače nemuseli psát heslo.

²Na Aise je nainstalované příliš staré GHC, které se od toho námi používaného liší typy některých seznamových funkcí.

0.4 Práce s dokumentací

Tak jako u ostatních programovacích jazyků je většina funkcí a typů zdokumentovaná. Primárním zdrojem dokumentace pro jazyk Haskell v rozsahu našeho kurzu je [webová dokumentace základního modulu Prelude](#). Dále můžete využít [lokální mirror vyhledávače Hoogle](#), kde můžete vyhledávat funkce podle názvu nebo typu.³

Jindřiška varuje: Naučit se pracovat s dokumentací je nevyhnutelné pro libovolný programovací jazyk, nejenom pro Haskell. Čím dříve se naučíte číst dokumentaci, tím budete mít lehčí život nejen tu, ale i v dalších předmětech.

Př. 0.4.1 Pomocí vyhledávače funkcí v jazyce Haskell najděte všechny funkce, které mají typ `Bool -> Bool -> Bool` a jsou v balíčku `base`.

»=

*
**

Jindřiška varuje: Ať už budete používat vlastní či školní počítač, je naprosto nezbytné, abyste před prvním cvičením zvládli jeho obsluhu alespoň v rozsahu fialového rámečku na začátku následující kapitoly.

Fialové rámečky na konci cvičení shrnují probrané koncepty. Můžete si s jejich využitím ověřit, jestli všemu ze cvičení rozumíte, nebo je potřeba se k některému tématu vrátit.

Na konci cvičení byste měli zvládnout:

- ▶ vytvořit textový soubor s Haskellovým kódem;
- ▶ spustit v terminálu GHCi a načíst do něj vytvořený soubor;
- ▶ pracovat v terminálu vzdáleně na školním počítači Aisa;
- ▶ pracovat s webovou dokumentací.

³Obě tyto služby pro vás hostujeme lokálně, protože ty oficiálně se ukázaly být pro naše účely neoptimální – originální dokumentaci najdete na [Hackage](#) a vyhledávání i v nestandardních balíčcích na [hoogle.haskell.org](#).

Řešení

Řeš. 0.η.0 Gratulki.

Nenechávejte si ale řešení prozradit příliš brzy. Má-li mít práce se sbírkou smysl, je potřeba se nad každým příkladem nejprve zamyslet a následně ho samostatně vypracovat. Dokonce ani když se vám implementace nedaří, nenakukujte hned do řešení – nejvíce se naučíte, pokud se nad příkladem sami trochu potrápíte. Samozřejmě klidně konzultujte přednášku nebo již hotové příklady.

Teprve až příklad zdárně vyřešíte, můžete nahlédnout do vzorového řešení. Občas se můžete dozvědět jiný (a v něčem třeba elegantnější) postup nebo nějaké tipy a triky.

Klepnutím na číslo řešení přeskočíte zpátky na příklad.

Řeš. 0.1.3 Vytvořit jej můžete buď v terminálu (příkazem `mkdir jméno` a následně do něj přepnout pomocí `cd jméno`), nebo v grafickém správci souborů, který by se měl otevřít v domovském adresáři a nový adresář by mělo být lze vytvořit pravým myšítkem nebo přes panel nabídek.

Řeš. 0.1.5 Spustíte textový editor (například Gedit nebo VS Code – code na fakultních PC) a napište či překopírujte do něj text. Nezapomeňte soubor uložit do správného adresáře.

Alternativně můžete soubor vytvořit i v terminálu, například editorem `nano`, který vám v dolní části zobrazuje klávesové zkratky, kterými se ovládá (znamená `Ctrl`).



Jiným oblíbeným terminálovým editorem je `vim`. Jeho obsluha vypadá zpočátku krkolomně, ale dá se rychle naučit: zkuste příkaz `vimtutor`, nebo `vimtutor cs`, případně `vimtutor sk`. (Ukončit je lze zadáním `:'q` a odetřováním.)

Řeš. 0.2.2 Zadejte výraz a stiskněte `Enter`. Např. `40 + 2` se vyhodnotí na `42`. Vyhodnocení výrazů zapisujeme ve studijních materiálech pomocí lomené šipky s hvězdičkou: `40 + 2 ~>* 42`. Další příklady:

- `4 * 10 + 2 ~>* 42`
- `2 ^ 8 ~>* 256`
- `(3 + 4) * 6 ~>* 42`

Řeš. 0.2.3 Načíst soubor lze buď pomocí `:load Sem0.hs` nebo jen `:l Sem0.hs` (pokud se nachází ve stejném adresáři, kde jsme spustili GHCi), nebo tak, že spustíme GHCi přímo s tímto názvem souboru (`ghci Sem0.hs`). Konstantu vypíšeme prostě tak, že ji napíšeme do GHCi a stiskneme enter: `hello ~>* "Hello, world"`. Její typ pak zjistíme příkazem interpretu `:t`, který jako parametr bere libovolný výraz: `:t hello ~>* hello :: [Char]`.

Řeš. 0.2.4

```
my_number :: Integer
my_number = 42
```

Znovunačtení lze provést příkazem `:r`, ukončení `:q`.

V případě, že vytvoříte proměnnou jenom v interpretu, pak ta přestane po ukončení interpretu existovat.

Řeš. 0.3.1 Na Linuxu, macOS a Windows s dostatečně novým PowerShellem by mělo stačit zadat do příkazové řádky (či PowerShellu) `ssh xLOGIN@aisa.fi.muni.cz` (samozřejmě s vaším loginem) a následně zadat své fakultní heslo. Nenechte se překvapit tím, že při zadávání hesla nevidíte žádné hvězdičky; v terminálu se hesla běžně zadávají „naslepo“. Ocitnete na Aise a můžete tam používat příkazovou řádku jako na cvičení. Na Windows s Putty musíte nastavit sezení (*session*), kde jako hostitele uvedete `xLOGIN@aisa.fi.muni.cz`.

Pro trvalé přidání modulu můžete příkaz vložit do svého souboru `.bashrc`:

```
(aisa)$ echo "module add ghc" >> ~/.bashrc # Nebo textovým
↪ editorem
```

Verze GHCi se vypíše po spuštění interpretu, nebo po zadání příkazu `ghci --version`.

Řeš. 0.3.2 Aby první následující příkaz fungoval, je potřeba být lokálně v adresáři, kde je soubor `Sem0.hs` a na Aise musí existovat adresář `ib015`.

```
(local)$ scp Sem0.hs xLOGIN@aisa.fi.muni.cz:ib015/
(aisa)$ nano Sem0.hs # Nebo jiný způsob editace
(local)$ scp xLOGIN@aisa.fi.muni.cz:ib015/Sem0.hs Sem0_edited.hs
```

Řeš. 0.3.3 Do souboru `~/.ssh/config` na svém počítači vložte (a nastavte svůj login):

```
Host aisa
  HostName %h.fi.muni.cz
  User xLOGIN
```

Řetězec `%h` v nastavení nechte, ten představuje speciální sekvenci, za kterou se doplní jméno napsané za `Host` – takových jmen může být více (oddělených mezerou). Pokud daný soubor neexistuje, vytvořte jej (i na Windows může být jednodušší vytvořit minimálně složku `~/.ssh` pomocí příkazové řádky, ta by však existovat měla). Soubor `config` nesmí mít žádnou příponu. Část `~` v cestě značí domovský adresář. Složka `.ssh` nebude ve výpisu přes `ls` vidět, pokud nepoužijete `ls -A`, což způsobí zobrazení skrytých souborů – těch co začínají `.` (tečkou).

Řeš. 0.4.1 Jděte na <https://hoogle.haskell.org/>. V rozbalovací nabídce u vyhledávacího políčka zvolte `package:base` a do vyhledávacího políčka zadejte hledaný typ. Po chvíli by vám vyhledávač měl najít funkce `(&&)`, `(||)`, `(==)` a `(/=)`. Poslední dvě mají obecnější typ, ale fungují i pro argumenty typu `Bool` – Hoogle vyhledává i obecnější funkce, do kterých lze za typové proměnné dosadit požadované typy.⁴ U každé funkce také vidíte, ve kterém balíčku (`base`) a modulu (`Prelude`⁵) se nachází, a také dokumentační text, pokud je uveden. Kliknutím na funkci se dostanete do její dokumentace v prvním modulu, kde byla nalezena.

⁴V tomto případě je navíc typ omezený typovým kontextem `Eq =>`, což znamená, že za `a` lze dosadit pouze porovnatelné typy, což `Bool` splňuje.

⁵`Prelude` je základní Haskellový modul, který je vždy k dispozici.