

Aritmetika + Proměnné

část 1.

A taky správa kódu

Git

- Systém pro správu verzí (verzování)
- Pracuje se složkou, zvanou repozitář
- Používá se pro sledování změn zdrojového kódu
- Změny se uloží vytvořením revize, které je pak přiřazeno číslo
- Výhody:
 - Lze uchovávat více verzí najednou (ve větvích) - například když programátor testuje několik způsobů řešení
 - Lze se vrátit na předešlé verze, když si programátor rozbije kód
 - Zjednodušuje kombinaci změn při spolupráci více programátorů
 - Lze nahrát na vzdálený repozitář (např Github nebo Gitlab) - jednoduchá záloha

Poznámka: pokud to není absolutně nutné, nepřidáváme do repozitáře binární soubory

Správa kódu

- Cargo vytvoří automaticky nový repozitář, není potřeba řešit
- Repozitář zaregistruje změny kódu příkazem 'git add'
 - pro jednotlivý soubor: 'git add jmeno.txt'
 - pro všechno v dané složce: 'git add .'
- Nová revize se vytvoří příkazem 'git commit'
 - zpráva revize == stručný popis změn
 - zvykem je první revizi nazývat 'initial commit'
- Zrcadlo na Githubu se přidá pomocí 'git remote add origin URL'
 - je potřeba udělat jen jednou
- Repozitář z Githubu si zklonujete (=stáhnete) pomocí 'git clone URL'
- Na Github se změny nahrají pomocí příkazu 'git push'
 - Pokud jste váš repozitář přidali jako zrcadlo a ne klonovali, příkaz je 'git push -u origin master'
 - zeptá se na username a heslo
 - repozitář musí existovat i na Githubu

Modus operandi

- Pro každou ‘vylepšenou’ verzi programů z prezentace se hodí udělat si revizi
- Změny v rámci mini-úkolů není potřeba zaznamenávat do repozitáře, ale volba je na vás
- Každý domácí úkol by měl mít repozitář v naší organizaci na Githubu. Gjkbót ho automaticky přesune do studentského týmu.
 - Jméno repozitáře si zvolte tak, aby bylo jedinečné, ale vystihovalo obsah repozitáře
 - Když je úkol hotový, přidejte k popisu repozitáře na Githubu tag ‘r’ (=ready), když gjkbót odevzdání zmerčí, pošle mi email a změní tag na ‘o’ (=ok)
- Doporučuji ‘pushnout’ všechny repozitáře, kde jste něco měnili než odejdete z hodiny, abyste předešli ztrátě dat

Příklad

```
$ # změna souborů
$ git add .
$ git commit
$ # napíšu zprávu, třeba 'přidal jsem králíčky'
$ git remote add origin
https://github.com/luciusmagn/helloworld
$ git push
$ # napíšu username a heslo pro Github
$ # změny kódu jsou nyní viditelné i na Githubu
```

Kalkulačka

- Vytvoříme nový projekt
 - `cargo new calc --bin`
- Vložíme kód do `main.rs` (další slide)
- Spustíme (`cargo run -- 5 + 2`)
- Jestli funguje, uděláme revizi
 - `git add . && git commit`
- hurá?

```
fn main() {  
    println!("mezi +nekonečno a -nekonečno");  
}
```

Kalkulačka v2.0

```
fn main() {  
    let a = 5;  
    let b = 2;  
    println!("vysledek: {}", a + b);  
}
```


Úkol

1. Součet 3 čísel
2. Jiné matematické operace
 - a. -
 - b. *
 - c. /
 - d. %
 - e. <<
 - f. >>
3. Závorky () ke změně pořadí operací

let

- Deklarace (resp. tvorba) proměnné (angl. variable binding)
- Váže hodnotu k bloku kódu, když blok skončí, hodnota se okamžitě smaže
- Části:
 - **let** - klíčové slovo
 - **a** - identifikátor/jméno
 - **=** - operátor přiřazení hodnoty
 - **5** - výraz
 - **;** - ukončuje příkaz