Algoritmizace- úvod

Ing. Tomáš Otáhal

Historie

- 9. století
- perský matematik a astronom
- Mohammed Al-Chorezím
- v latinském přepise příjmení= algoritmus

Nejstarší algoritmus

Euklides

- řecký matematik, 4. století před n. l. Euklidova geometrie (axiomy)

Euklidův algoritmus

(hledá největšího společného dělitele dvou daných čísel)

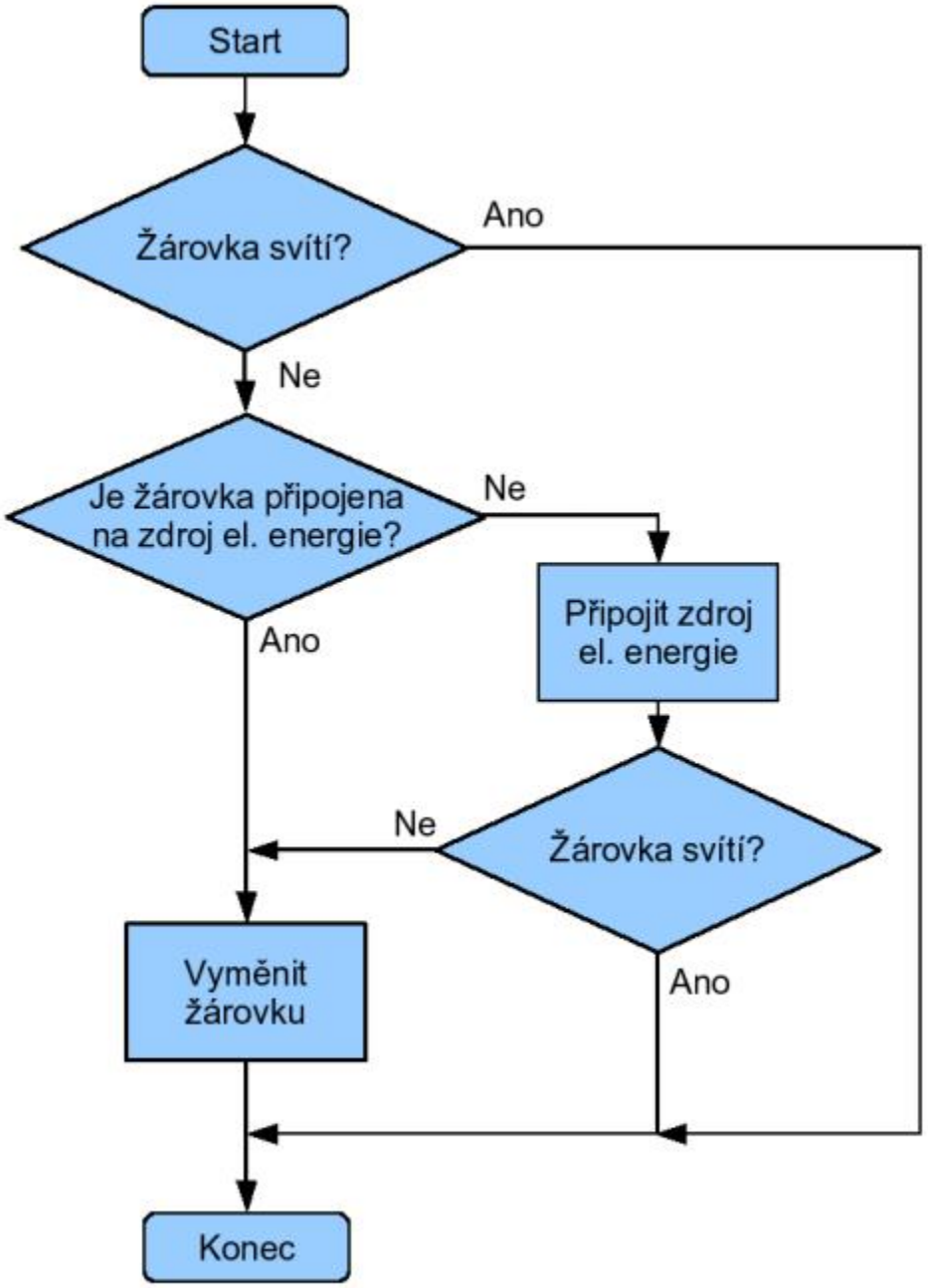
Algoritmus

Program x Programování

Postup, který je v počítači prováděn nějakým programem se nazývá **algoritmus (program)** a jeho tvorba **algoritmizace (programování).**

Algoritmus = popis procesu, který vede od měnitelných vstupních údajů k požadovaným výsledkům.

Algoritmus je jednoznačný a přesný popis řešení problému.


Start
Žárovka svítí?
Ano
Ne
Je žárovka připojena na zdroj el. energie?
Ne
Připojit zdroj el. energie
Ano
Ne
Žárovka svítí?
Ano
Vyměnit žárovku
Konec


Where should I eat?
Fast Food Edition
Are you in Canada?
No
Yes
No
Is it after 10:00 PM?
Yes
Is it before 10:30 AM?
Are you a police officer?
Yes
No
Yes
No
Yes
No
Do you live on the westcoast?
Is your name Jared?
No
Yes
No
No
Do you have more than $3?
Yes
Are you drunk?
Yes
Yes
No
Are you high?
On malt liquor?
No
Yes
No
Do you want dessert too?
Yes
Yes
No
Are you in a car?
Do you want fried food?
No
Will you eat fish?
Yes
No
Yes
No
Would you ever eat at Arby's?
No
Yes
Is it Sunday?
Yes
Are you older than 16?
Yes
No
No
Eating The Road

Algoritmus

Naší snahou je **vybrat pro řešení problému** co **nejefektivnější algoritmus, který řeší problém** v co nejkratším čase, je přehledný a srozumitelný.

Vstupní informace:

- vycházíme z řešení úlohy
- musí splňovat vstupní podmínky

Výstupní informace:

- získáme nové informace
- výsledek realizace algoritmu
- musí splňovat výstupní podmínky

Algoritmus lze vyjádřit:

1. **slovně:** jednotlivé kroky postupu jsou **vyjádřeny větami** v přirozeném jazyce
2. **graficky:** jednotlivé kroky jsou popsány grafickými značkami se slovním popisem, například pomocí tzv. **vývojových diagramů**
3. **matematicky:** soustavou **rovnic, vztahem** mezi veličinami
4. **programem:** jednotlivé kroky jsou **popsány instrukcemi** určitého procesoru

Efektivnost algoritmu

Danou úlohu řeší více algoritmů, vybíráme efektivnější podle určitých kritérií:

- **časové:** úloha vyřešena v kratším čase (strojový čas tj. počet instrukcí procesoru)
- **paměťové:** spotřeba paměti
- **přehlednost, srozumitelnost:** (důležité pro další vývoj a úpravy)

Každý algoritmus musí mít:

- **správnost:** **výsledek**, který vznikne použitím algoritmu, musí být **správný**
- **resultativnost:** po konečném počtu kroků **dospěje k řešení** (vrátí třeba jen chybové hlášení)
- **konečnost:** algoritmus se **nezacyklí**, po určitém počtu kroků skončí
- **determinovanost:** v každém kroku je **jednoznačně** určen způsob pokračování práce algoritmu
- **hromadnost:** znamená, že algoritmus lze použít pro **řešení obecné úlohy**, tj. že nepopisujeme postup jedné úlohy, ale poslouží k řešení libovolné úlohy, která patří do jisté třídy úloh
- **opakovatelnost:** algoritmus vede vždy ke **stejným výsledkům**, jsou-li **zadána stejná data**

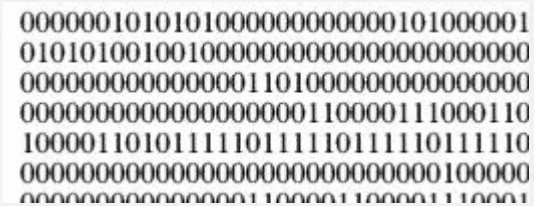
Programovací jazyk

Programovací jazyk

- Umělý jazyk jenž se používá pro definování sekvence programových příkazů, které lze zpracovat na počítači.
- Algoritmus má obecnou povahu, zatímco implementace algoritmu v určitém programovacím jazyku je ryze konkrétní.

Nižší programovací jazyky

- procesor bude vykonávat ty instrukce, které programátor napíše
- programátor musí vše vypisovat vše (složitý zdrojový kód)
- **strojový kód** (to, co uvidíte, když otevřete obsah „exe“ souboru v textovém editoru). Strojový kód = Soubor číslicových instrukcí, které je procesor počítače schopen rozpoznat a uskutečnit.



- Assembler – je velice blízký strojovému kódu

Vyšší programovací jazyky

- struktura je logická
- přenositelnost – může běžet na různých platformách OS, ale i HW pc
- Do strojového kódu se převádí kompilátorem
- Př. jazyk C++, JAVA, PHP, Delphi,..)

Programovací jazyky dále dělíme:

- **kompilované**
- **interpretované**
- **programování strukturované** (např. Pascal)
- **objektově orientované programování** (OOP)
  – např. Visual Basic, Delphi

Kompilované jazyky

- **nejdříve celé přeloženy** a až **potom mohou být spuštěny**
- Jsou rychlejší, vyšší nároky na formální správnost kódu
- překládají se **kompilátorem**
- výsledkem překladu je (většinou) .exe soubor
- patří sem většina klasických programovacích jazyků

Interpretované jazyky

- **překládány až za běhu programu**
- pomalejší, ale nemají tak velké požadavky
- překládají se **interpretem**,
- interpret **instrukce** při překladu zároveň **provádí**
- nevýhoda- musejí vždy spouštět v interpretu
- do této skupiny patří například Basic, skriptovací jazyky (PHP, Python, Perl ...).

Opakování- otázky

Otázky i s odpověďmi mi zašlete společně s vývojovým diagramem.

- Co je to algoritmus?
- Co jsou vstupní údaje?
- Co jsou výstupní údaje?
- Vyjmenuj a popiš vlastnosti algoritmu.
- Jak lze vyjádřit algoritmus?
- Co to je programování?
- Co to je algoritmizace?
- Vyjmenuj kritéria efektivnosti algoritmu.
- Co je programovací jazyk?
- Vysvětli rozdíl mezi vyššími a nižšími programovacími jazyky.
- Vysvětli rozdíl mezi interpretovanými a kompilovanými programovacími jazyky.

Algoritmizace- postup

Ing. Tomáš Otáhal

Algoritmizaci má tyto kroky:

1. Formulace problému
2. Analýza úlohy
3. Vytvoření algoritmu
4. Sestavení programu
5. Odladění programu

Algoritmizace- slovní úlohy na procvičení

Ing. Tomáš Otáhal

Algoritmus přípravy pomerančové bowle

Cvičení 1- napsat postup

Algoritmus přípravy pomerančové bowle

1) Formulace problému
   - Připrav pomerančovou bowli.
2) Analýza úlohy
   - Vstupní údaje: 1 kg pomerančů, 30 dkg práškového cukru, 5 dcl vína, 0,3 l sifonu, 3 lžíce rumu
   - Výstupní údaje: pomerančové bowle
   - Analýza: aplikovat správný postup
3) Sestavení algoritmu (slovní popis)
   - Oloupej pomeranče
   - Rozkrájej je na malé kousky
   - Dej kousky pomeranče do mísy a zasyp cukrem
   - Přidej víno a nechej zchladit
   - Před podáním přidej rum a sifon

Algoritmus zatloukání hřebíků

Cvičení 2- napsat postup

Algoritmus zatloukání hřebíků

1) Formulace problému
   - Zatluč hřebík do desky
2) Analýza úlohy
   - Vstupní údaje: kladivo, hřebík, deska
   - Výstupní údaje: hřebík zatlučen do desky
   - Analýza: tlouct tak dlouho, dokud není hřebík zatlučen až po hlavičku
3) Sestavení algoritmu (slovní popis)
   - A) Vezmi kladivo a hřebík
   - B) Přilož hřebík k desce
   - C) Uhoď kladivem na hlavičku
   - D) Je hřebík zatlučen?
     ANO – Pokračuj bodem E
     NE – Vrať se na bod C
   - E) Ukonči činnost a odlož kladiv

Algoritmus přechodu křižovatky, řízené semaforem

Cvičení 3- napsat postup

Algoritmus přechodu křižovatky, řízené semaforem

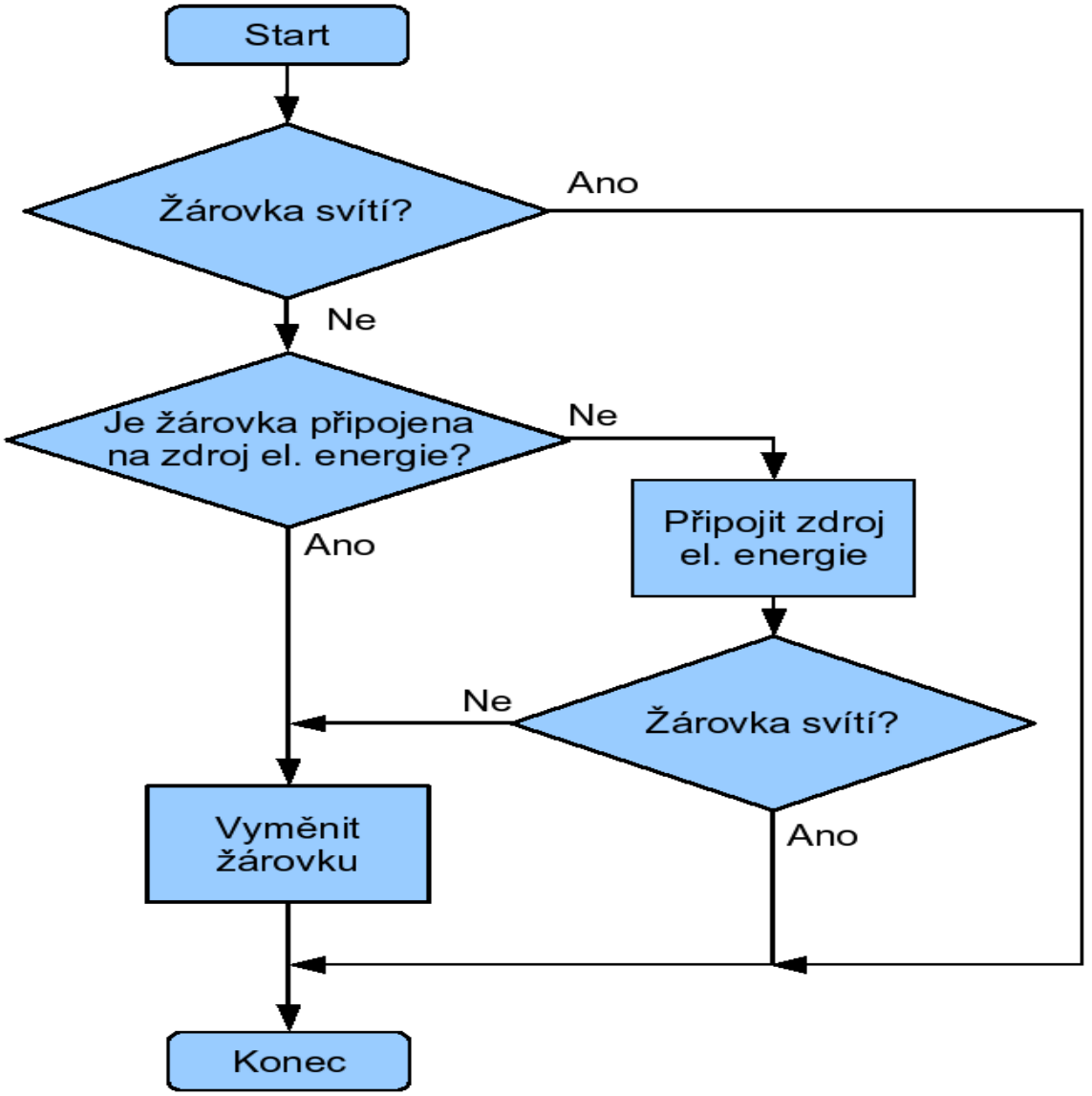
1) **Formulace problému**
   - Přejdi na druhou stranu ulice.
2) **Analýza úlohy**
   - Vstupní údaje: přechod se semaforem
   - Výstupní údaje: pozice na druhé straně ulice
   - Analýza: přes přechod se nechodí na červenou
3) **Sestavení algoritmu (slovní popis)**
   - A) Dojdi až k semaforu
   - B) Svítí na semaforu červená?
     ANO – Čekej, vrať se na bod B
     NE – Pokračuj bodem C
   - C) Přejdi přes přechod

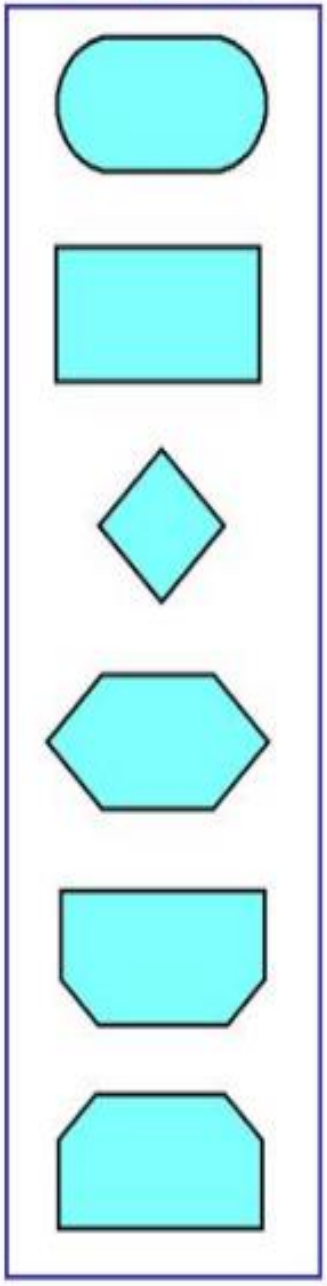
Vývojové diagramy

Ing. Tomáš Otáhal

Vývojový diagram

- VD= způsob znázornění algoritmů
- používá se několik typů značek
- každá značka má určitý význam
- do značek se vpisují operace nebo skupiny operací, které se mají provést
- pomocí značek se kreslí základní algoritmické konstrukce


Start
Žárovka svítí?
Ano
Ne
Je žárovka připojena na zdroj el. energie?
Ne
Připojit zdroj el. energie
Ano
Ne
Žárovka svítí?
Ano
Vyměnit žárovku
Konec



začátek/konec algoritmu

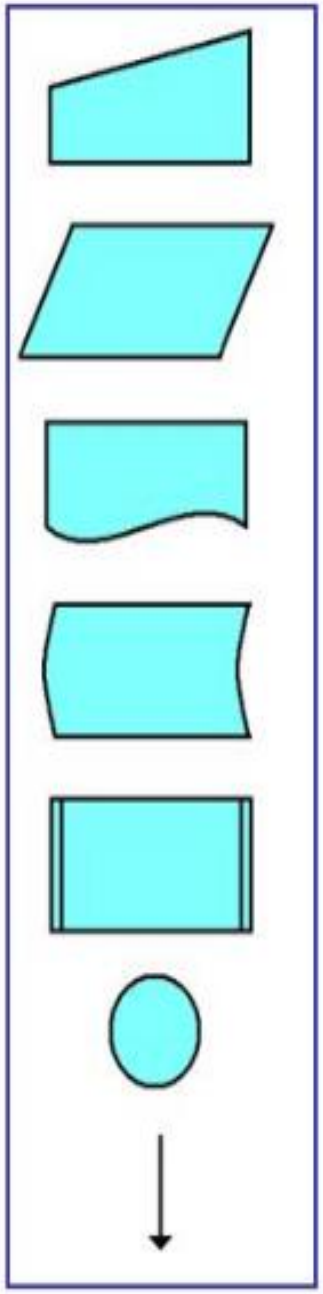
běžný příkaz

podmíněné větvení

cyklus s určeným počtem opakování

cyklus s podmínkou na konci

cyklus s podmínkou na začátku



ruční vstup

zobrazení výstupu

zpracování souboru

uložení dat do souboru

podprogram

spojovací značka

spojovací čára (tok algoritmu)

Cvičení- slovních algoritmů

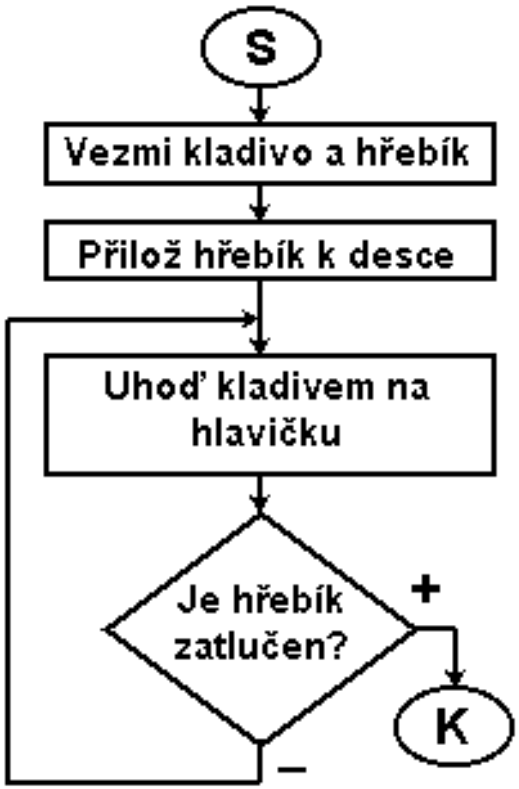
Zatloukání hřebíků

Slovní popis algoritmu:

1. Vezmi kladivo a hřebík
2. Přilož hřebík k desce
3. Uhoď kladivem na hlavičku
4. Je hřebík zatlučen?
   ANO – pokračuj bodem 5
   NE – vrať se na bod 3
5. Ukonči činnost a odlož kladivo

Řešení- zatloukání hřebíků

Vývojový diagram:

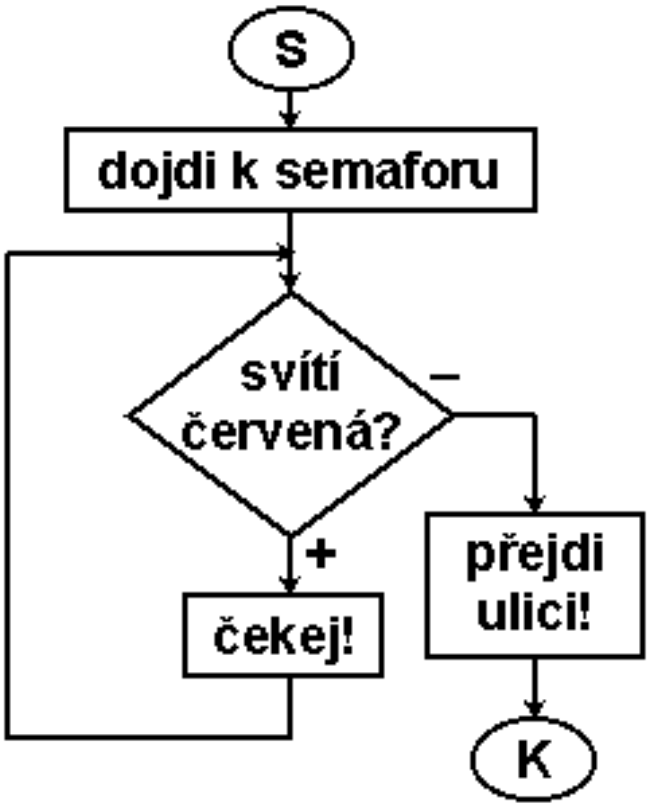


Algoritmus přechodu křižovatky, řízené semaforem

1) **Formulace problému**
   - Přejdi na druhou stranu ulice.
2) **Analýza úlohy**
   - Vstupní údaje: přechod se semaforem
   - Výstupní údaje: pozice na druhé straně ulice
   - Analýza: přes přechod se nechodí na červenou
3) **Sestavení algoritmu (slovní popis)**
   - A) Dojdi až k semaforu
   - B) Svítí na semaforu červená?
     ANO – Čekej, vrať se na bod B
     NE – Pokračuj bodem C
   - C) Přejdi přes přechod

Řešení- přechodu

Vývojový diagram:



Praktické cvičení

1. Napiš algoritmus na přechod z jedné místnosti do druhé (dveře jsou zavřeny, mají zámek a ty máš svazek klíčů).
2. Nakresli vývojový diagram na přechod z jedné místnosti do druhé (dveře jsou zavřeny, mají zámek a ty máš svazek klíčů).

Děkuji za pozornost.