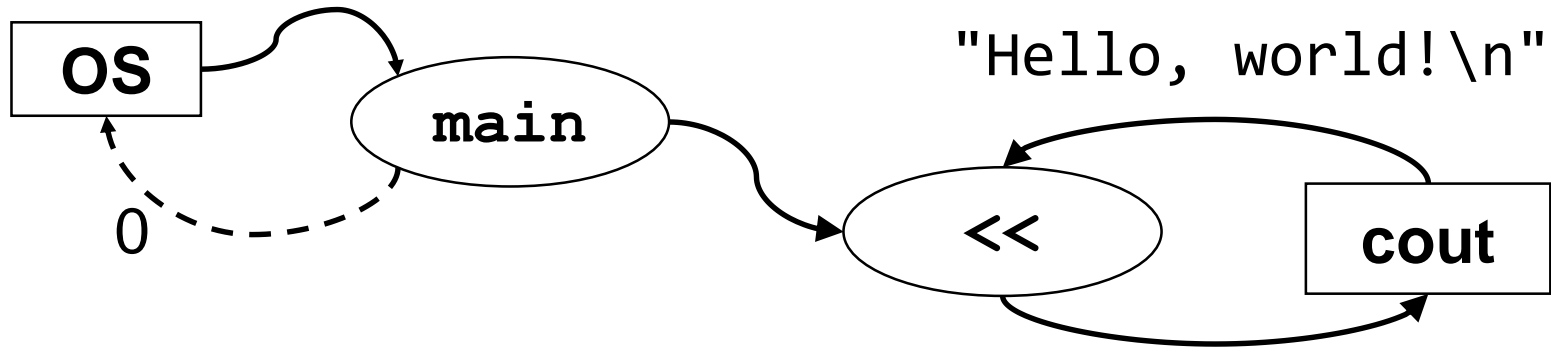


# Příklad “Hello, world!” v C++

<< znamená vypiš na výstup



```
#include <iostream>

int main() {
    std::cout << "Hello, world!\n";
}
```

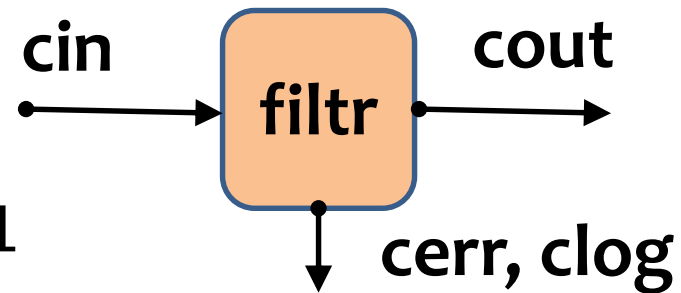
# Vstup a výstup v C++

- Každý program se chápe jako filtr – filtruje vstupní data na výstupní.
- Každý program proto má standardní vstup (`cin` – console input), standardní výstup (`cout` – console output), výstup pro hlášení chyb (`cerr` – console errors, `clog` – console log).
- Vstup pomocí `>>`  

```
std::cin >> x;
```
- Výstup pomocí `<<`  

```
std::cout << x;
```
- Konec řádku: `"\n"` nebo `endl`  

```
std::cout << "\n";  
std::cout << std::endl;
```



# Struktura programu v C++

- Program v C++ realizující řešení problému je sada definic tříd a funkcí, jedna z funkcí se jmenuje **main** a ta představuje hlavní program, který se spustí. Funkce **main** pak případně volá jiné funkce, či metody realizované ve třídách.
- Definice funkce **main** má např. tvar:

```
#include <iostream>
```

```
int main() {  
    std::cout << "Příklad - tisk vstupu do zadání nuly\n";  
    int hodn;  
    do {  
        std::cin >> hodn;  
        std::cout << "Hodnota = " << hodn << "\n";  
    } while (hodn != 0);  
    std::cout << "Konec\n";  
}
```

# Hlavičkové soubory

hello.h

```
#ifndef SAY_HELLO_H
#define SAY_HELLO_H
void sayHello();
#endif
```

hello.cpp

```
#include "hello.h"
#include <iostream>
void sayHello() {
    std::cout << "Hello\n";
}
```

world.h

```
#pragma once
void sayWorld();
```

world.cpp

```
#include "world.h"
#include <iostream>
void sayWorld() {
    std::cout << "World\n";
}
```

```
#include "hello.h"
#include "world.h"
int main() {
    sayHello();
    sayWorld();
}
```

# Preprocesor

- Povinná součást prostředí C a C++
- Provádí textové úpravy před vlastním překladem
- Řídí se příkazy pro preprocesor – **direktivami**
- Formát direktivy:

`#direktiva parametry`

- Příklady direktiv:

- Vložení textu

`#include <soubor>`

`#include "soubor"`

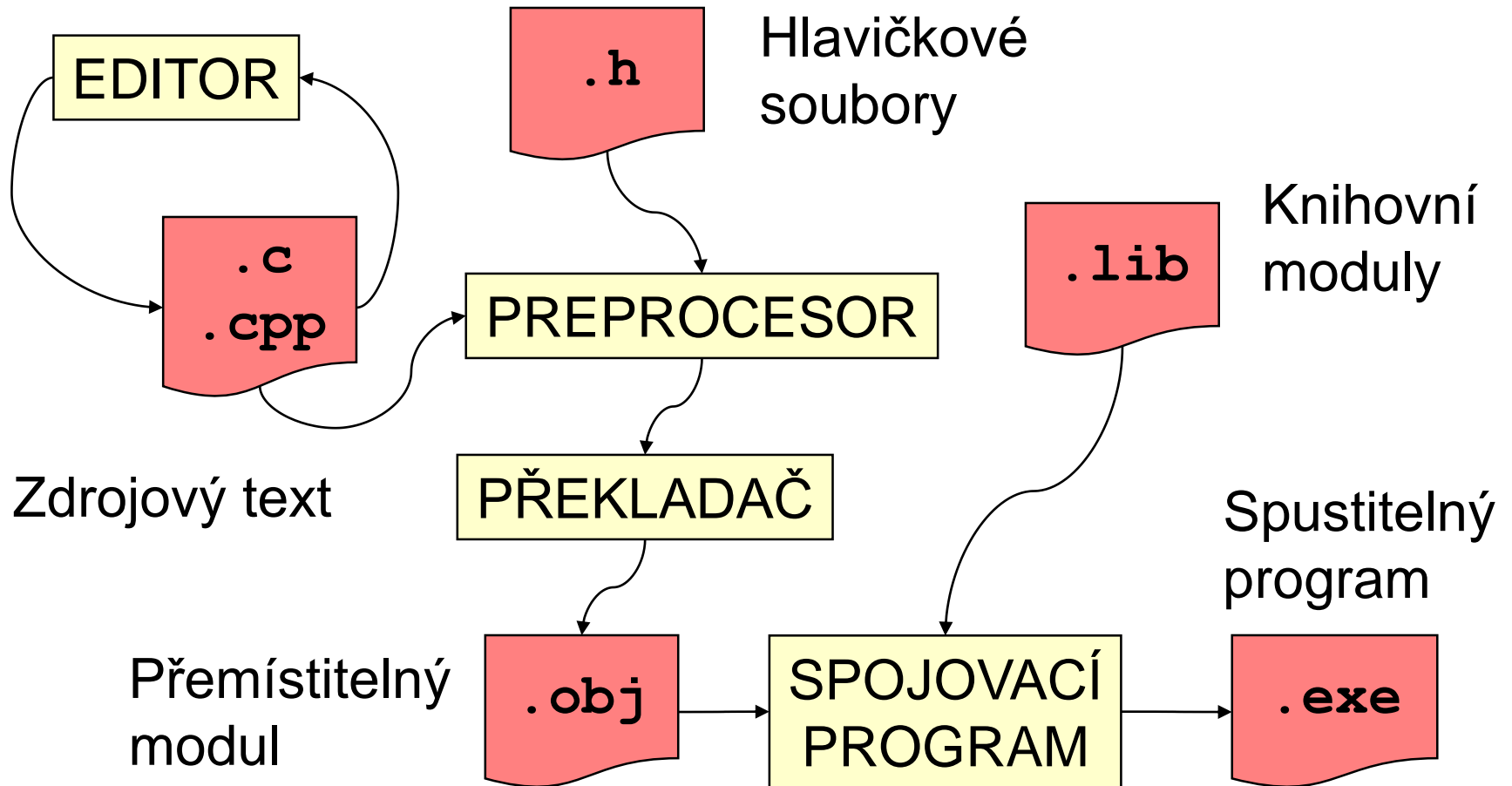
- Definice makra

`#define jméno text`

`#define jméno (parametry) text`

- Podmíněný překlad, parametry překladu (později)

# Fáze překladač



# Programy a programovací jazyky

**Program** je předpis pro provedení určitých akcí počítačem zapsaný v programovacím jazyku.

- Programovací jazyky:
  - strojově orientované (strojový jazyk, jazyk fyzického procesoru),
  - assembler (jazyk symbolických adres),
  - vyšší jazyky:
    - **imperativní** (příkazové, procedurální),
    - **neimperativní** (např. funkcionální, logické).
- Hlavní rysy imperativních jazyků (např. C, C++, Java, Pascal, Basic, atd.):
  - zpracovávané údaje mají formu **datových objektů** různých typů, které jsou v programu reprezentovány pomocí **proměnných**, resp. **konstant**, uložených v **paměti** počítače,
  - **program** obsahuje deklarace a příkazy,
  - **deklarace** definují význam jmen (identifikátorů),
  - **příkazy** předepisují akce s datovými objekty nebo způsob řízení výpočtu.

**Konec**



# Program v C++ je sada funkcí

`main`

`funkce1`

`funkceN`

`funkce2`

# Jedna z nich se jmenuje `main`

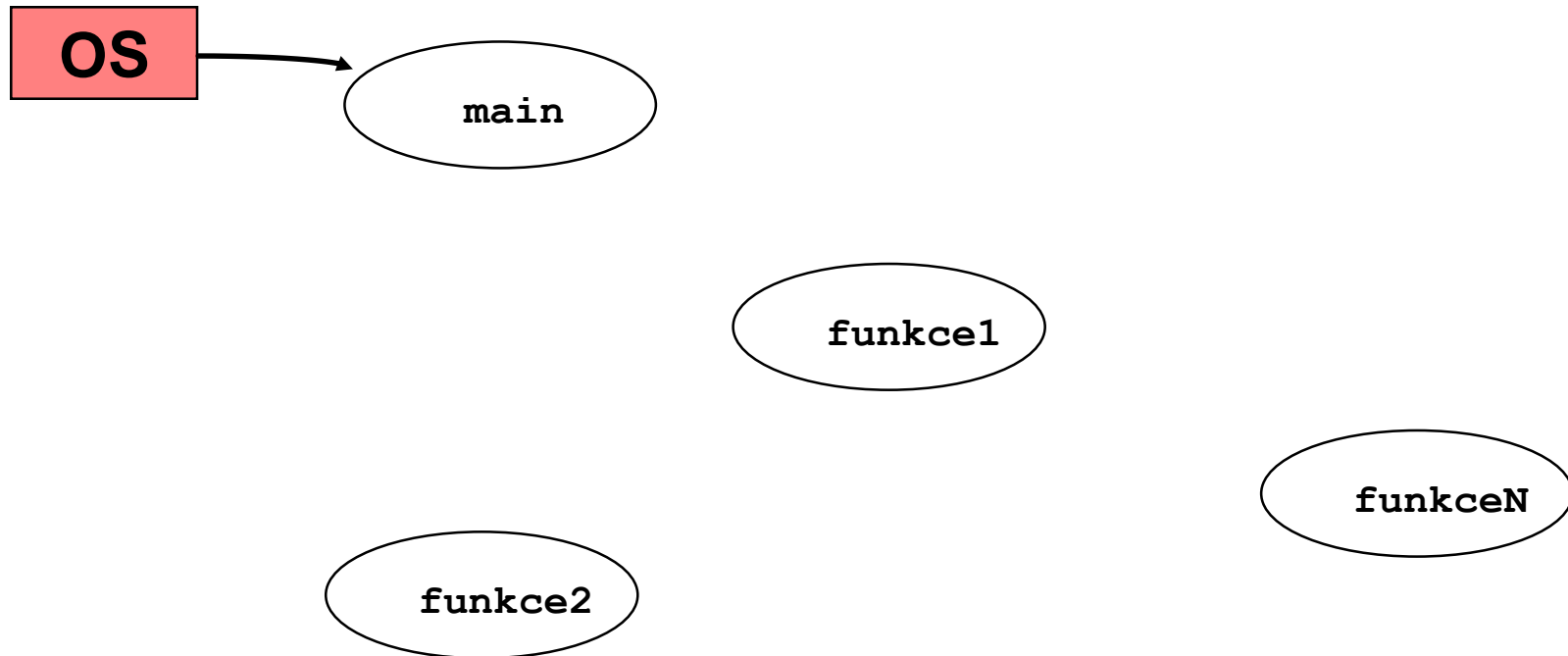
`main`

`funkce1`

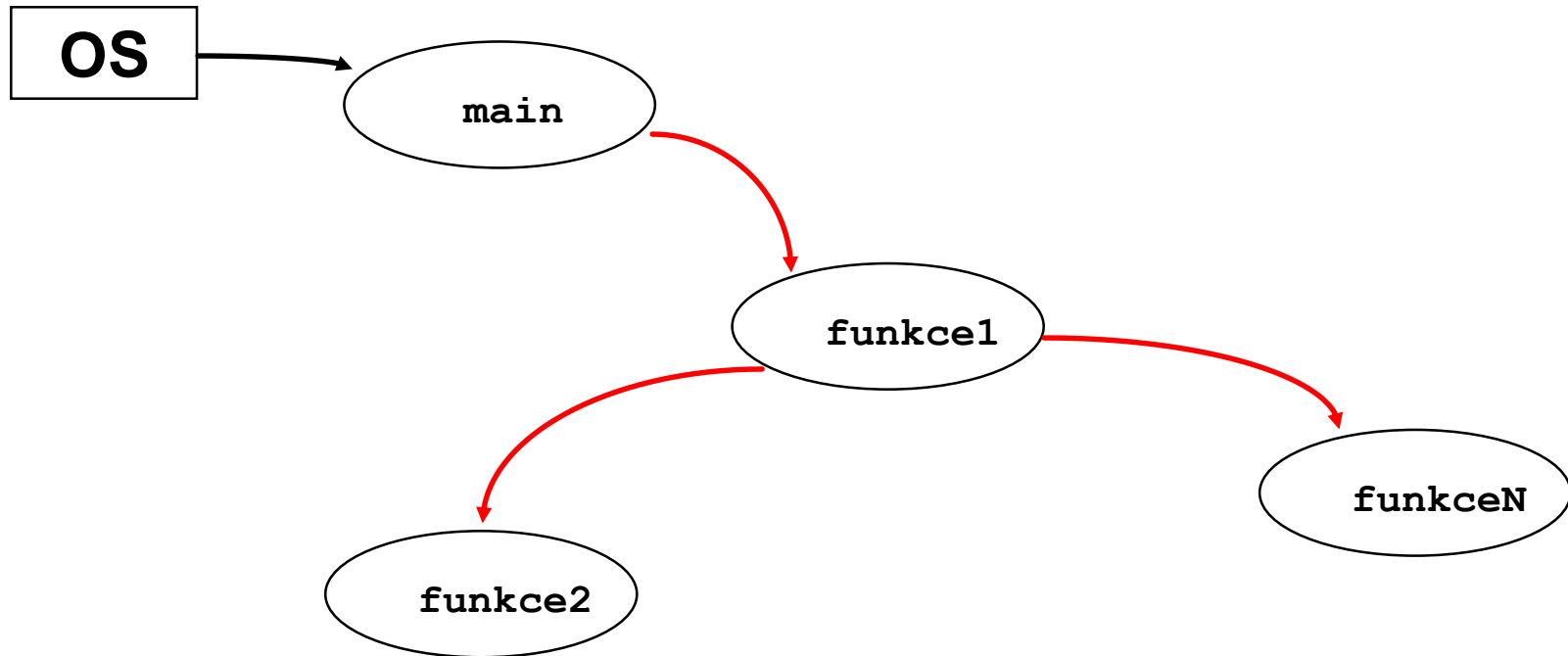
`funkceN`

`funkce2`

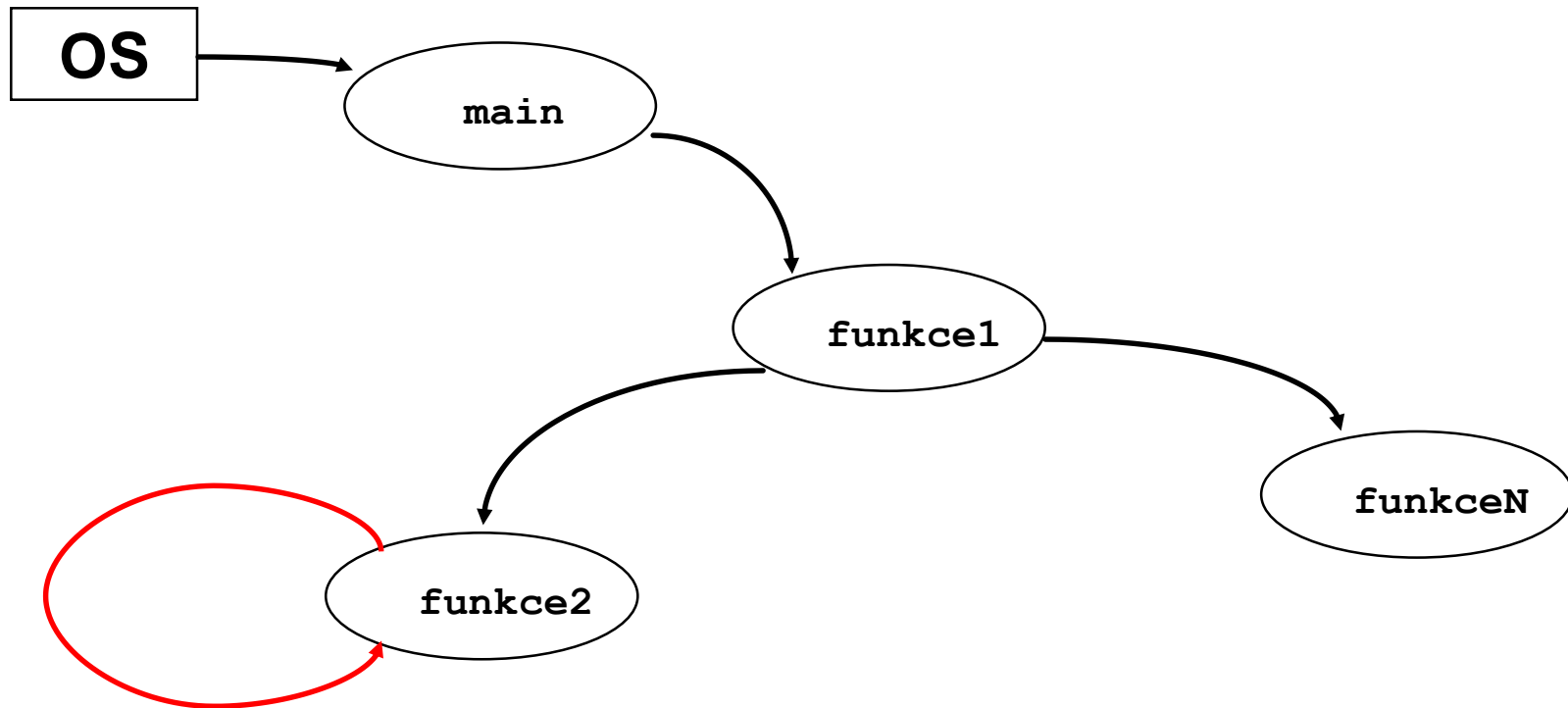
# Funkci `main` spustí OS



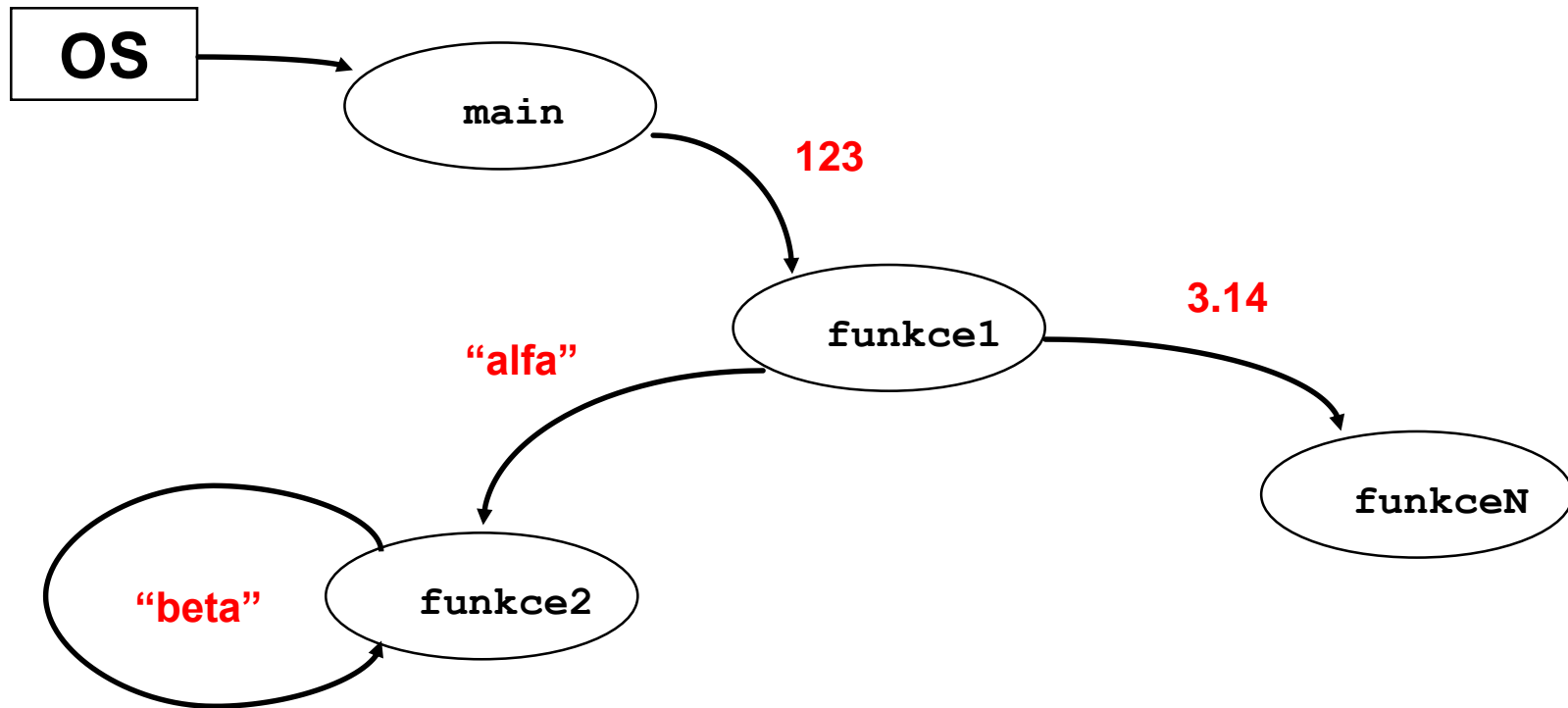
# Funkce spolu komunikují (volají se)



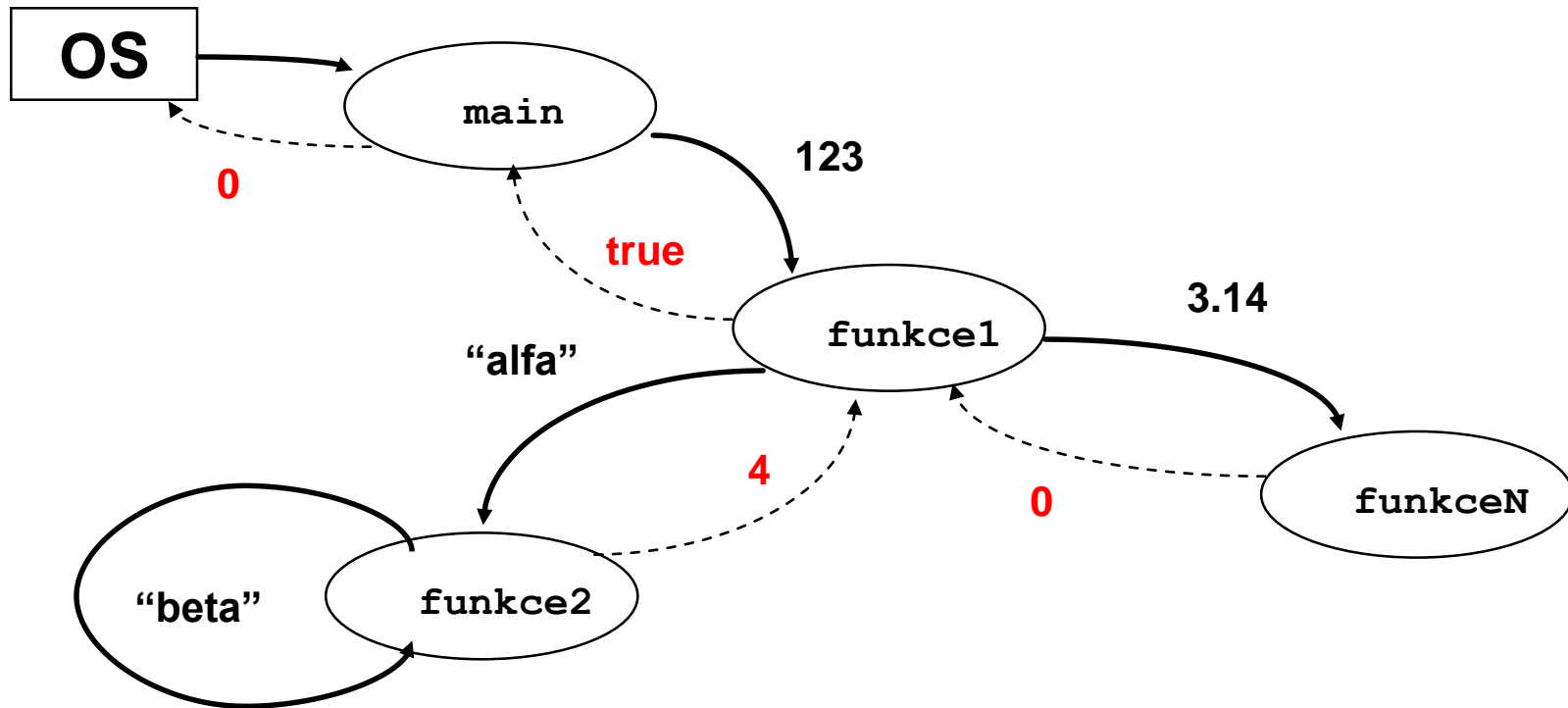
# Případně i rekurzivně



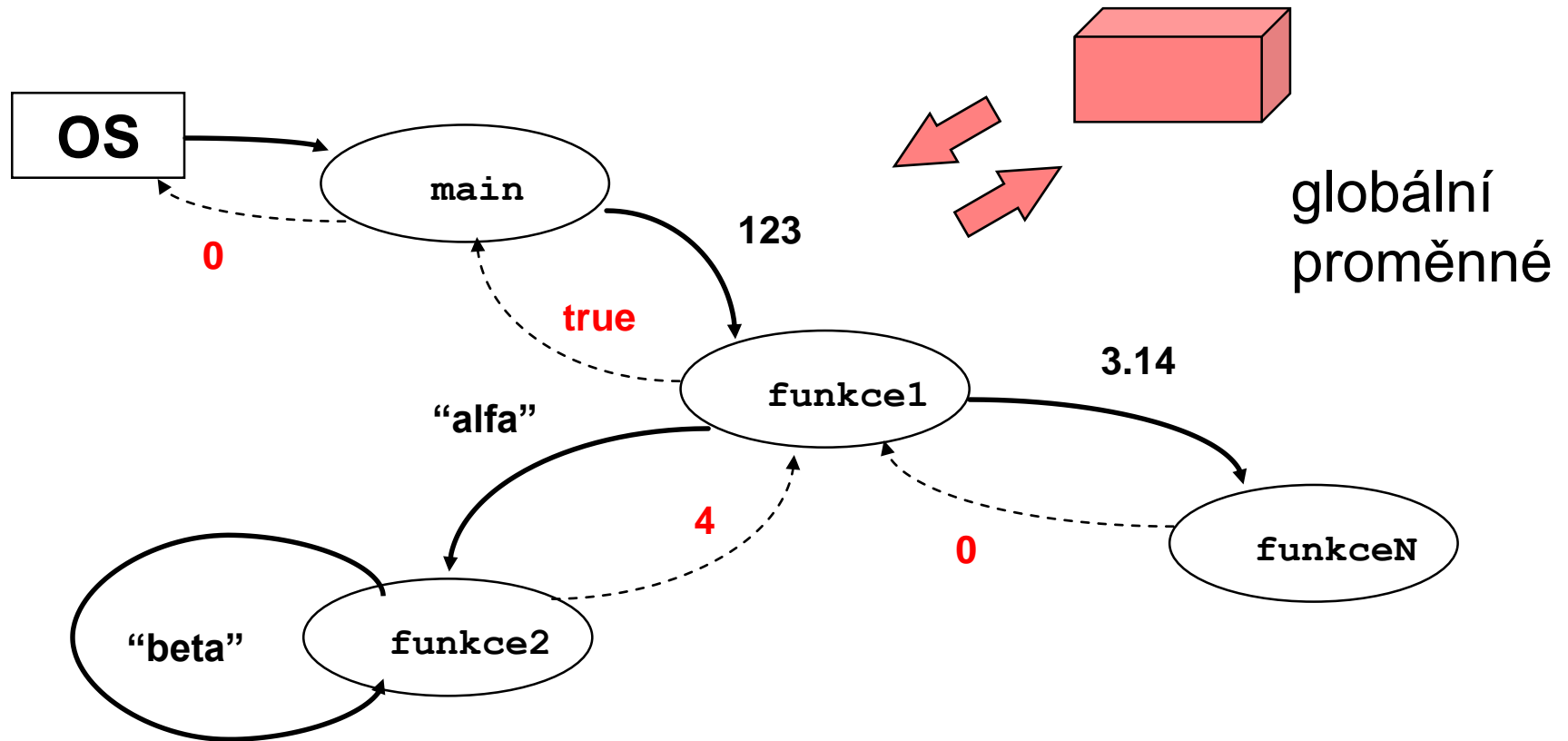
# Předávají si parametry



# A vracejí výsledky

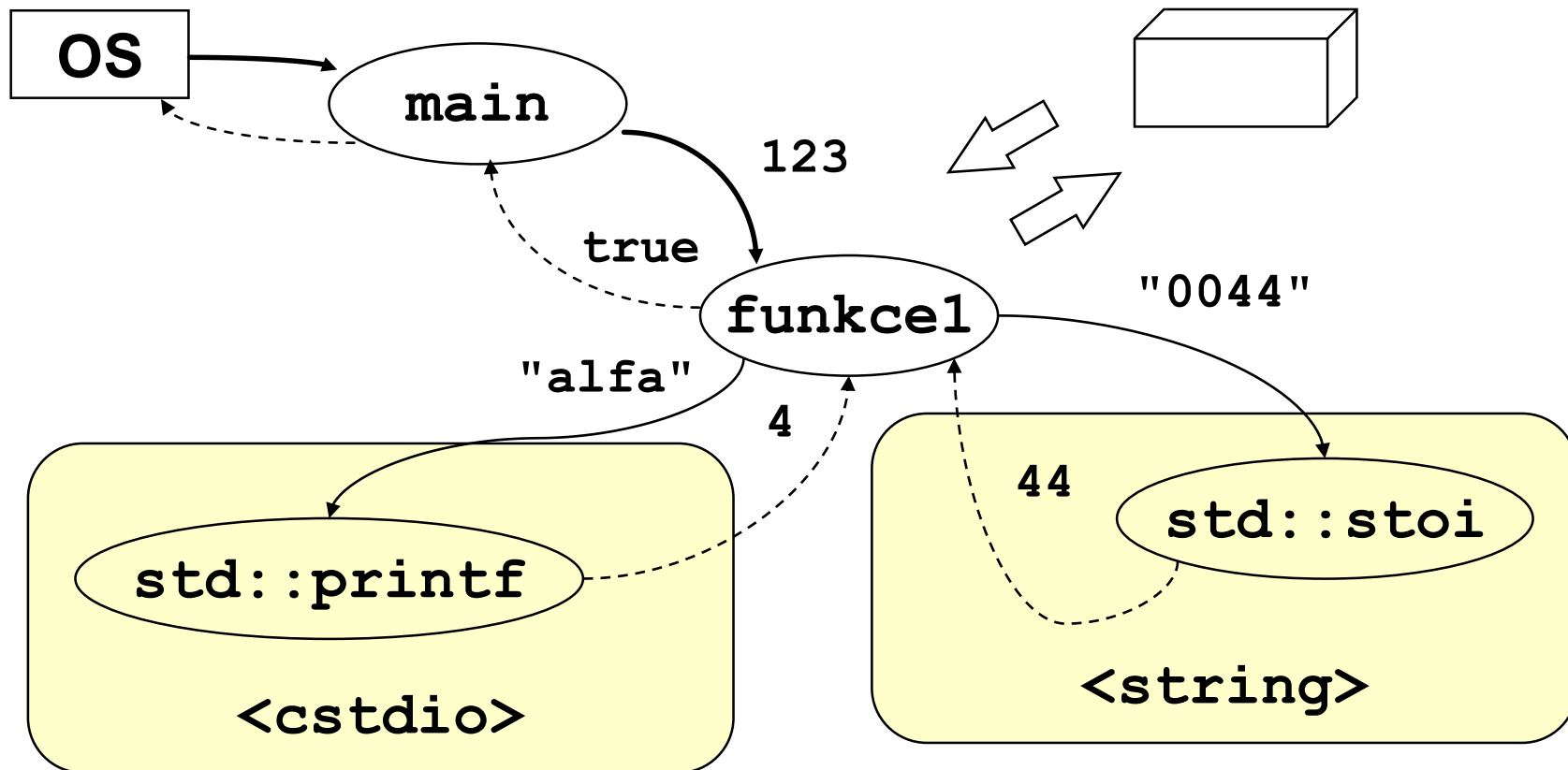


# Navíc existuje globální paměť'





# Některé funkce jsou připraveny v knihovných funkcích



# Formát zápisu programu v C není předepsán

```
#include <stdio.h>
int main(int t, int _, char *a){return!o<t?t<3?main(-79,-13,a+main(-87,1-_,
main(-86,o,a+1)+a)):1,t<_?main(t+1,_,a):3,main(-94,-27+t,a)&&t==2?_<13?
main(2, _+1,"%s %d %d\n"):9:16:t<o?t<-72?main(_,t,
"@n'+,#'/*}{w+/w#cdnr/+,}{r/*de}+,/*{*+,/w{%,/w#q#n+,/#{l,+,/n{n+,/+#n+,/#\
;q#n+,/+k#;*+,/r:'d*'3,}{w+K w'K:'+}e#';dq#l \
q#+d'K#!/+k#;q#'r}eKK#}w'r}eKK{nl]'/#;#q#n')}{#}w')}{nl]'/#n';d}rw' i;# \
){nl]'/#n{n#'; r{#w'r nc{nl]'/#{l,+K {rw' iK{;[{nl]'/#w#q#n'wk nw' \
iwk{KK{nl]'/#w{%'l##w# i; :{nl]'/*{q#l'd;r'}{nlwb!/*de}'c \
;;{nl'~}{rw]'/#,}##*}{#nc,',#nw]'/#kd'+e}+;#'rdq#w! nr'/ ') }+}{rl#'{n' ')# \
}'+'}##(!/!)"
:t<-50? _==*a?putchar(31[a]):main(-65,_,a+1):main((*a=='/')+t,_,a+1)
:o<t?main(2,2,"%s"): *a=='/||main(0,main(-61,*a,
"!ek;dc i@bK'(q)-[w]*%n+r3#l,}{:\nuwloca-O;m .vpbks,fxntdCeghiry"),a+1);}
```

Zdroj: Wikipedia, International Obfuscated C Code Contest

# Výstup tohoto programu

On the first day of Christmas my true love gave to me a partridge in a pear tree.

On the second day of Christmas my true love gave to me two turtle doves and a partridge in a pear tree.

On the third day of Christmas my true love gave to me three French hens, two turtle doves and a partridge in a pear tree.

On the fourth day of Christmas my true love gave to me four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the fifth day of Christmas my true love gave to me five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the sixth day of Christmas my true love gave to me six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the seventh day of Christmas my true love gave to me seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the eighth day of Christmas my true love gave to me eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the ninth day of Christmas my true love gave to me nine ladies dancing, eight maids a-milking, seven swans a -swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the tenth day of Christmas my true love gave to me ten lords a-lea ping, nine ladies dancing, eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the eleventh day of Christmas my true love gave to me eleven pipers piping, ten lords a-lea ping, nine ladies dancing, eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

On the twelfth day of Christmas my true love gave to me twelve drummers drumming, eleven pipers piping, ten lords a-lea ping, nine ladies dancing, eight maids a-milking, seven swans a-swimming, six geese a-laying, five gold rings; four calling birds, three French hens, two turtle doves and a partridge in a pear tree.

# Výstup tohoto programu (česky)

Na první den Vánoc mi moje pravá láska dala koroptev v hrušce (Ježíš Kristus).

Na druhý den Vánoc mi moje pravá láska dala dvě hrdličky (Starý a Nový zákon) a koroptev v hrušce.

Na třetí den Vánoc mi moje pravá láska dala tři francouzské slepice (víra, naděje, láska), dvě hrdličky a koroptev v hrušce.

Na čtvrtý den Vánoc mi moje pravá láska dala čtyři volající ptáky (evangelisté), tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Na pátý den Vánoc mi moje pravá láska dala pět zlatých prstenů (5 knih Mojžíšových), čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Šestý den Vánoc mi má pravá láska dala šest snášejších hus (šest dní stvoření světa), pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Sedmý den Vánoc mi moje pravá láska dala sedm plovoucích labutí (sedm darů Ducha svatého), šest snášejších hus, pět zlatých prstenů, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Osmého dne Vánoc mi moje pravá láska dala osm dojících služek (osm blahoslavenství), sedm plovoucích labutí, šest snášejších hus, pět zlatých prstenů, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

V devátý den Vánoc mi moje pravá láska dala devět tančících dam (ovoce Ducha svatého), osm dojících služek, sedm plovoucích labutí, šest snášejších hus, pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Desátý den Vánoc mi má pravá láska dala deset skákajících pánů (Desatero přikázání), devět tančících dam, osm dojících služek, sedm plovoucích labutí, šest snášejších hus, pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Jedenáctého dne Vánoc mi moje pravá láska dala jedenáct dudajících dudáků (jedenáct věrných apoštolů), deset skákajících pánů, devět tančících dam, osm dojících služek, sedm plovoucích labutí, šest snášejších hus, pět zlatých kroužků, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

Na dvanáctý den Vánoc mi moje pravá láska dala dvanáct bubnujících bubeníků (dvanáct bodů víry apoštolů), jedenáct dudajících dudáků, deset skákajících pánů, devět tančících dam, osm dojících služek, sedm plovoucích labutí, šest snášejších hus, pět zlaté prsteny, čtyři volající ptáky, tři francouzské slepice, dvě hrdličky a koroptev v hrušce.

# Výpočetní problém

- Specifikace výpočetního problému má dvě části:
  - definici vstupu (vstupních dat) a
  - definici výstupu (výstupních dat).
- Instance výpočetního problému → nějaká konkrétní přípustná vstupní data. K jedné instanci problému existovat několik správných výstupů (obecně se jedná o výpočet, představující relaci).
- Příklad: problém řazení čísel (Sorting Problem for Numbers).
  - Vstup: Posloupnost  $n$  čísel  $In = \langle a_1, \dots, a_n \rangle$ .
  - Výstup: Taková permutace  $Out = \langle a'_1, \dots, a'_n \rangle$  stejných čísel z  $In$ , pro kterou platí, že:  $a'_1 \leq \dots \leq a'_n$ .
- Instance problému řazení je např. zadání: seřad'te vzestupně posloupnost  $In = \langle 75, 11, 34, 176, 59, 6, 54 \rangle$ . Správným výstupem řazení je pak posloupnost  $Out = \langle 6, 11, 34, 54, 59, 75, 176 \rangle$ .
- Pozn.: Pro tento vstup je výstup určen jednoznačně, ale neplatí to pro každý

# Výpočetní problém – příklad

- **Úloha:** najděte největšího společného dělitele (NSD) dvou přirozených čísel (víme, co je největší společný dělitel dvou přirozených čísel?)
- **Řešení:**
  - Popišme postup tak, aby byl použitelný pro dvě libovolná přirozená čísla:
    - označme zadaná čísla  $x$  a  $y$  a menší z nich  $d$
    - není-li  $d$  společným dělitelem  $x$  a  $y$ , pak  $d$  zmenšíme o 1, test opakujeme a skončíme, až  $d$  bude společným dělitelem  $x$  a  $y$
- **Poznámka:**
  - Význam symbolů  $x$ ,  $y$  a  $d$  použitých v algoritmu:
    - jsou to proměnné (paměťová místa), ve kterých je uložena nějaká hodnota, která se může v průběhu výpočtu měnit

# Algoritmus pro největšího spol. dělitele

**Úloha:** najděte největšího společného dělitele dvou přirozených čísel

Přesnější popis:

- **Vstup:** přirozená čísla  $x$  a  $y$
- **Výstup:**  $\text{NSD}(x,y)$
- **Postup:**
  1. Je-li  $x < y$ , pak  $d$  má hodnotu  $x$ , jinak  $d$  má hodnotu  $y$
  2. Pokud je  $d$  dělitelem  $x$ , a zároveň je  $d$  dělitelem  $y$ , jdi na krok 5
  3. Zmenši  $d$  o 1
  4. Jdi na krok 2
  5. Výsledkem je hodnota  $d$
- Sestavili jsme algoritmus pro výpočet největšího společného dělitele dvou přirozených čísel.

# Zápis algoritmu NSD v C++

```
#include <iostream> // pro std::cin a std::cout
#include <algorithm> // pro std::min

int deli(int d, int x) {
    return x % d == 0;
}

int main() {
    int x, y;
    std::cout << "Zadej dve cisla: ";
    std::cin >> x >> y;
    int d = std::min(x, y);
    while (!(deli(d, x) && deli(d, y))) {
        d -= 1;
    };
    std::cout << "NSD(" << x << ", " << y << ") = " << d << "\n";
}
```



# Různé styly řešení problémů

- Zápis programu realizuje určitou funkci či algoritmus, které představují řešení určitého problému.
- K řešení problému můžeme dospět různými způsoby – styly.
- Dva základní styly, kterými se budeme zabývat, jsou strukturovaný styl a objektově-orientovaný styl.
- Každý z nich se hodí na řešení trochu jiných problémů.
- **Strukturovaný styl** se hodí pro řešení procesních problémů, kdy řešení skládáme z posloupností jednotlivých kroků.
- **Objektově-orientovaný styl** se hodí spíše pro řešení různých informačních, či návrhových systémů, které uvažujeme jako sít' komunikujících objektů.

# Strukturovaný styl: elementární algoritmické konstrukty

## **Posloupnost (sekvence):**

- kroky algoritmu, které jsou provedeny pouze jednou.

## **Větvení (selekce):**

- změna průběhu algoritmu na základě vyhodnocení podmínky.

## **Cyklus (iterace):**

- opakování části algoritmu (těla cyklu), dokud je splněna podmínka.

## **Dílčí algoritmus (podprogram, procedura, funkce, metoda):**

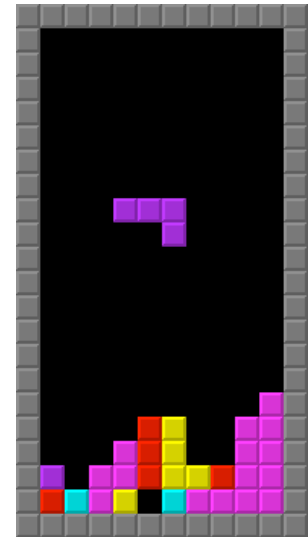
- část algoritmu, kterou lze použít opakovaně.

# Objektově-orientovaný styl

- Objektově Orientované Programování (OOP):
  - Vychází z intuitivního chápání světa.
  - Základním konceptem jsou objekty.
  - Objekty mají určité *vlastnosti (properties)* a *schopnosti (capabilities)*.
- Objekty jsou stavební bloky programu:
  - Program je tedy soubor interagujících objektů.
- Objekty mohou modelovat **reálné věci**:
  - auto, hrnek, židle, zeměkoule.
- Objekty mohou modelovat **koncepty**:
  - schůzka, shromáždění, telefonát.
- Objekty mohou modelovat **procesy**:
  - hledání cesty bludištěm, seřazení balíčku karet.

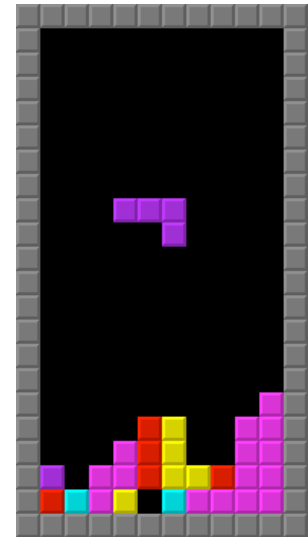
# Příklad – Tetris

- Jaké jsou herní objekty?
- Co musí tyto objekty umět vykonat?
- Jaké vlastnosti tyto objekty mají?



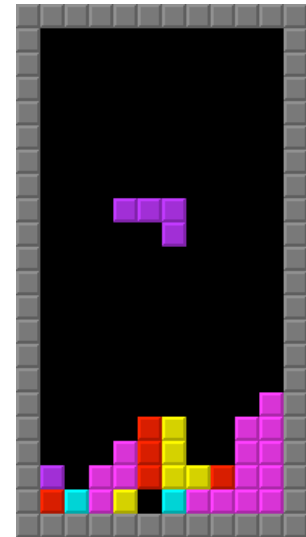
# Příklad – Tetris

- Jaké jsou herní objekty?
  - např. dílky a hrací deska.
- Co musí tyto objekty umět vykonat?
- Jaké vlastnosti tyto objekty mají?



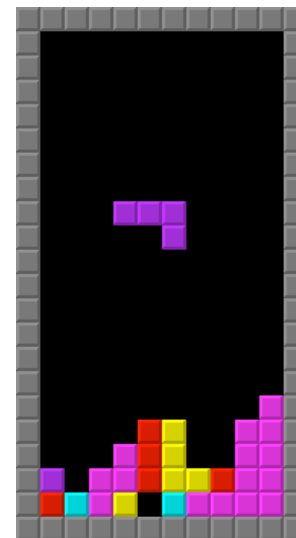
# Příklad – Tetris

- Jaké jsou herní objekty?
  - např. dílky a hrací deska.
- Co musí tyto objekty umět vykonat?
  - dílek:
    - být vytvořen, spadnout, zastavit se, otočit se,
  - hrací deska:
    - být vytvořena, odstranit řádek,
    - zkontrolovat konec hry.
- Jaké vlastnosti tyto objekty mají?



# Příklad – Tetris

- Jaké jsou herní objekty?
  - např. dílky a hrací deska.
- Co musí tyto objekty umět vykonat?
  - dílek:
    - být vytvořen, spadnout, zastavit se, otočit se,
  - hrací deska:
    - být vytvořena, odstranit řádek,
    - zkontrolovat konec hry.
- Jaké vlastnosti tyto objekty mají?
  - dílek
    - orientace, pozice, tvar, barva,
  - hrací deska
    - velikost, počet řádek.



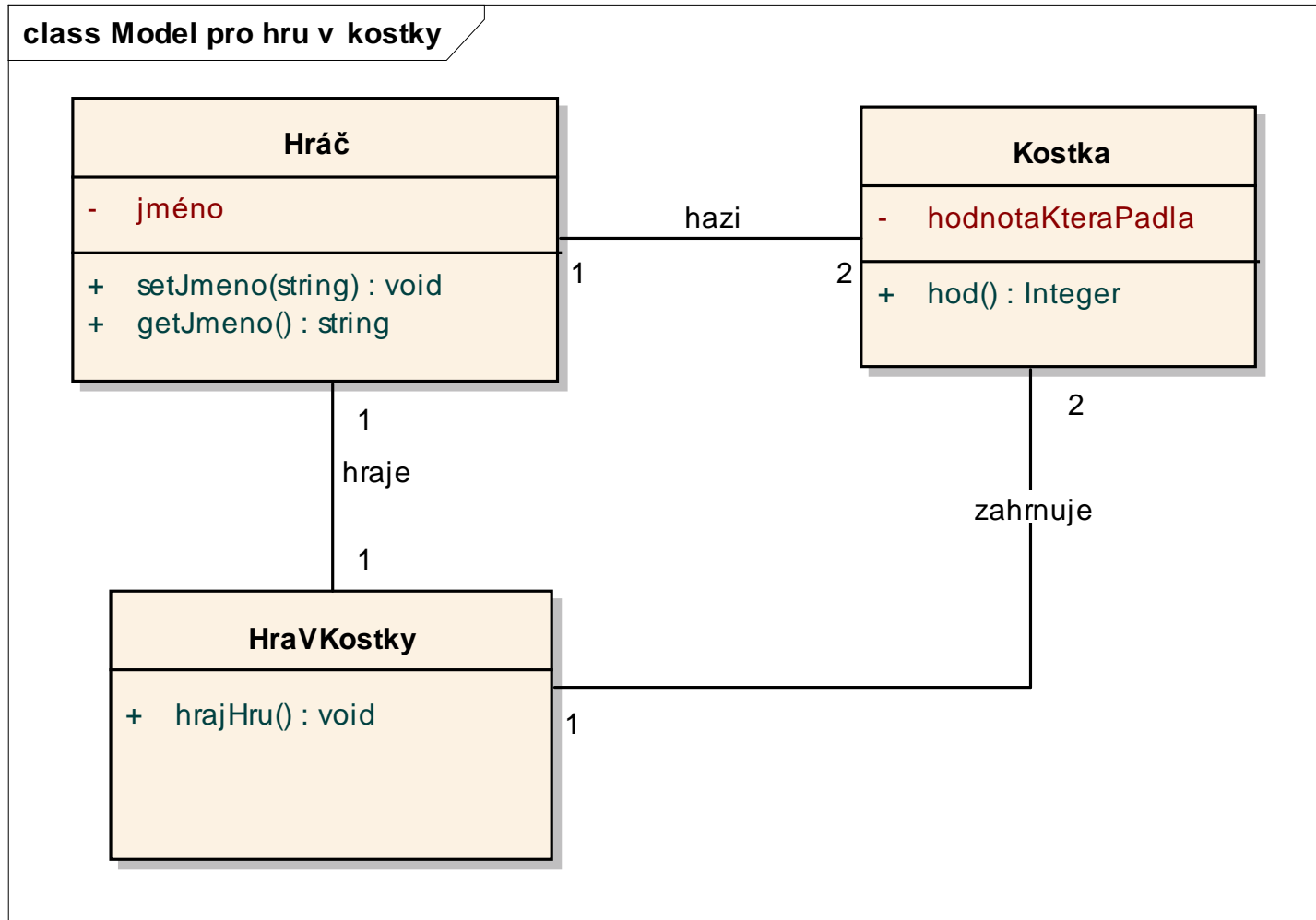
# Příklad: Hra v kostky

- Uvažme příklad velmi jednoduché hry s kostkami – máme dvě kostky a házíme. Pokud nám vyjde součet v daném hodů 7, vyhráli jsme, jinak jsme naopak prohráli.
- Objektové řešení vychází z podstaty úlohy – pracujeme s objekty typu „kostka“. To jsou instance obecné třídy `Kostka`.
- `Kostka` umí metodu `hod`, která nastaví interní hodnotu kostky hodnotou `random` na náhodné číslo a vrátí toto číslo jako výsledek.
- Hru hrají hráči, kteří mají jméno – to je možno jim pomocí metod `nastavit` a `číst` (`setJmeno/getJmeno`).
- Hra spočívá v tom, že hráč hodí kostkami – to nám může zajistit metoda `hrajHru`.





# Objektový model pro hru v kostky



# Třída Kostka

```
#include <cstdlib>

class Kostka {
private:
    int hodnotaKteraPadla;
public:
    int hod() { return hodnotaKteraPadla = (std::rand() % 6) + 1; };
};
```

# Třída Hrac

```
#include <string>

class Hrac {
private:
    std::string jmeno;
public:
    void setJmeno(std::string jm) { jmeno = jm; }
    std::string getJmeno() { return jmeno; }
};
```

# Třída HraVKostky

```
#include <cstdlib>
#include <ctime>

class HraVKostky {
private:
    Kostka kostka1;
    Kostka kostka2;
public:
    HraVKostky() {
        std::srand((unsigned)std::time(0));
    }
    int hrajHru() {
        return kostka1.hod() + kostka2.hod();
    }
};
```

# Hlavní program main pro kostky

```
#include <iostream>
```

```
int main() {  
    HraVKostky Hra;  
    Hrac hrac;  
    std::string jmeno;  
    int vysledek;  
    char volba;  
    std::cout << "Hra v kostky\n";  
    std::cout << "Vase jmeno: "; std::cin >> jmeno; hrac.setJmeno(jmeno);  
    std::cout << "Nazdar " << hrac.getJmeno() << "\n";  
    do {  
        vysledek = Hra.hrajHru();  
        if (vysledek == 7) std::cout << "Vyhral jsi!!!\n";  
        else std::cout << "Prohral jsi, soucet je: " << vysledek << "\n";  
        std::cout << "Pokracovat?: "; std::cin >> volba;  
    } while (volba != 'n');  
}
```

