

Číslo a název šablony	III/2 Inovace a zkvalitnění výuky prostřednictvím ICT
Číslo didaktického materiálu	EU-OPVK-VT-III/2-ŠR-101
Druh didaktického materiálu	DUM
Autor	RNDr. Václava Šrůtková
Jazyk	čeština
Téma sady didaktických materiálů	Programování v C# v příkladech I
Téma didaktického materiálu	Algoritmus, instalace C#
Vyučovací předmět	Seminář z informatiky
Cílová skupina (ročník)	Žáci ve věku 16–17 let
Úroveň žáků	začátečníci
Časový rozsah	1–2 vyučovací hodiny
Klíčová slova	Algoritmus, program, zápis algoritmu, C#
Anotace	Studenti se seznamují s pojmem algoritmu a jeho vlastnostmi, řeší úlohy, které prohlubují algoritmické myšlení
Použité zdroje	<p>DRÓZD, Januš a Rudolf KRYL. <i>Začínáme s programováním</i>. 1.vyd. Praha: Grada, 1992, 306 s. ISBN 80-854-2441-X.</p> <p>LIBICHER, Ivan a Pavel TÖPFER. <i>Od problému k algoritmu a programu: sbírka řešených úloh z programování</i>. 1. vyd. Praha: Grada, 1992, 119 s. Educa '99. ISBN 80-854-2482-7.</p> <p>TÖPFER, Pavel. <i>Algoritmy a programovací techniky</i>. 1. vyd. Praha: Prometheus, 1995, 299 s. ISBN 80-858-4983-6.</p> <p>TÖPFEROVÁ, Dana a Pavel TÖPFER. <i>Sbírka úloh z programování</i>. Vyd. 1. Praha: Grada, 1992, 98 s.</p> <p>VYSTAVĚL, Radek. <i>Moderní programování: učebnice pro začátečníky</i>. Ondřejov: moderníProgramování s.r.o, 2007, 2 sv. ISBN 978-80-903951-0-7.</p> <p>RENC, Zdeněk. <i>Sbírka řešených úloh z matematiky, fyziky a informatiky: přijímací řízení na MFF UK v letech 1992-96</i>. Vyd. 1. Editor Zdeněk Renc. Praha: Matfyzpress, 1997, 208 s. ISBN 80-858-6322-7.</p>
Typy k metodickému postupu učitele, doporučené výukové metody, způsob hodnocení, typy k individualizované výuce apod.	<p>Text je možno využít ke společné práci, samostatné přípravě studentů, domácímu studiu apod.</p> <p>Prezentace obsahuje stručné shrnutí poznatků potřebných pro řešení příkladů. V pracovním listu je zadání cvičení – většinou se jedná o úlohy, které by měli studenti vyřešit samostatně. Není nutné, aby všichni zpracovali všechno,</p>

	<p>vhodné je diferencovat podle jejich zájmu a schopností. Obtížnější úlohy jsou označeny hvězdičkou. Součástí materiálu je řešení těchto příkladů.</p> <p>Návrh způsobu hodnocení: ohodnocení samostatné práce během hodiny např. podle volby a počtu úloh a elaborace řešení (efektivnost, komentáře...).</p>
--	---

## Metodický list k didaktickému materiálu

### Prohlášení autora

Tento materiál je originálním autorským dílem. K vytvoření tohoto didaktického materiálu nebyly použity žádné externí zdroje s výjimkou zdrojů citovaných v metodickém listu.

Obrázky (schémata a snímky obrazovek) pocházejí od autora.

## 101. Úvod, algoritmus a jeho charakteristiky

### Cíl

- Naučit se řešit problémy s pomocí prostředků výpočetní techniky – jako výpočetní technika
- Pochopit do hloubky co se dá od těchto prostředků očekávat
- Naučit se myslet určitým způsobem – jiným než jsou třeba postupy v matematice – zpětné hlubší pochopení určitých problémů
- Naučit se základům řemesla

Naučit se dá všechno, ale chce to systematickou práci a trpělivost

### Jazyk a prostředí

Pro psaní programů používáme programovací jazyk.

Na jazyce tak moc nezáleží (Je to podobné jako s řízením auta. Naučíte-li se formulovat algoritmy v jednom jazyce stejného typu, není tak těžké přejít na jiný)

Jazyky jsou:

- strukturované, které zahrnují práci s objekty – C++, C#, Object Pascal, VBA, PHP
- funkcionální jako Prolog nebo Lisp – jiný způsob myšlení

Abychom mohli program napsat, potřebujeme také textový editor, nějaký ladící prostředek, v současnosti se používá vizuální návrh programů, kde standardní komponenty (ovládací prvky) Windows, jako jsou tlačítka, nástrojové lišty apod. nemusíme pracně graficky vytvářet, ale jsou k dispozici – to všechno tvoří integrované prostředí.

(Borland C++, Borland Delphi, Visual Basic,...)

## Proč právě C#

Visual C# 2010 Express Edition je jednodušší variantou Microsoft Visual Studio 2010. Nabízí všechny výhody moderního programování a je k dispozici volně. Poskytuje:

- Grafické uživatelské rozhraní
- Použití moderního, plně objektového programovacího jazyka

## Instalace a registrace vývojového prostředí

1. Spustit program vcs\_web.exe a postupovat dle návodu.  
(nebo použít odkaz Microsoft Visual Studio Express - Build cutting edge Windows applications a dále stejně)
2. Po nainstalování je nezbytná registrace: (bezplatná, ale poněkud pracná)  
Online nebo Offline
3. Online
  - a. Spustit C#, pak Help/register Product
  - b. Tlačítko Obtain Registration key online
  - c. Je třeba zadat identifikační údaje Windows Live ID – buď přímo nebo přes účet od [www.hotmail.com](http://www.hotmail.com) nebo [www.passport.net](http://www.passport.net)
4. Offline – Přímo zadat Registration key zasláný např. mailem

## Algoritmus a jeho charakteristiky

### Algoritmická úloha a její specifikace:

Jsou dány vstupní údaje vyhovující vstupním podmínkám a výstupní údaje splňující výstupní podmínky.

Vytvoř algoritmus – tj. napiš postup, který pro všechny údaje vyhovující vstupním podmínkám získá výstupní údaje vyhovující výstupním podmínkám.

Příklad 1: Vejce natvrdo. Předpokládáme, že existují tři druhy vajec: malá, střední a velká

- Vstup: Vejce syrové, čerstvé

- Výstup: Vejce natvrdo uvařené, jedlé

Postup:

- Dáme vařit vodu.
- Když se voda vaří (bublá), vložíme vejce.
- Nastavíme kuchyňský budík
  - V případě, že je vejce malé, na 1 minutu.
  - V případě, že je vejce střední, na 5 minut.
  - V případě, že je vejce velké, na 8 minut.
- Dokud budík nezazvoní, vejce vaříme.
- Vejce vyjmeme a opláchneme studenou vodou.

Příklad 2: řešení lineární rovnice v oboru reálných čísel ( $a \cdot x = b$ )

- Vstup  $a, b$  reálná čísla
- Výstup podle  $a, b$  buď kořen rovnice nebo oznámení, že daná rovnice má nekonečně mnoho řešení nebo nemá řešení.

Postup:

- Zjistíme  $a, b$
- Když  $a$  není nula, potom můžeme vypočítat kořen  $x = b / a$ , řešení je  $x$
- Jinak když  $b = 0$  potom je řešení libovolné reálné číslo
  - Jinak řešení neexistuje
- Vypíšeme zprávu o výsledcích

### Vlastnosti algoritmu:

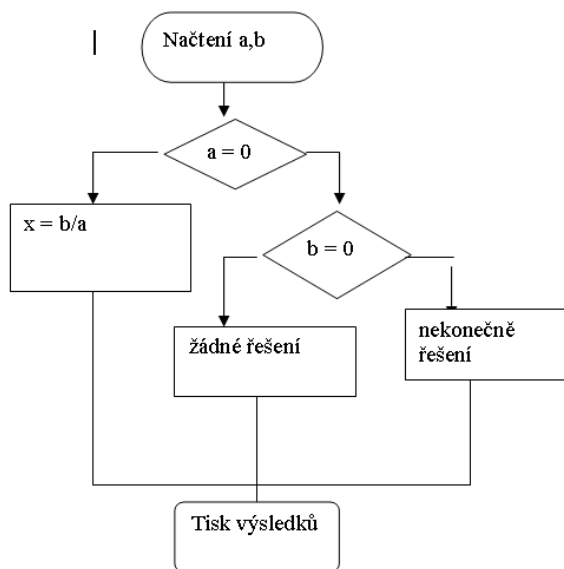
- **Přesnost a elementárnost:** algoritmus musí být formulován vzhledem k procesoru, který podle něj bude pracovat, tj. jazykem, kterému procesor rozumí a pomocí akcí, které umí vykonávat.
- **Hromadnost** – řeší dost velkou třídu úloh (ne jednu rovnici s konkrétními čísly, ale celou skupinu rovnic – vzpomínáte si na rovnice s parametry?)
- **Rezultativnost a konečnost** – po určitém počtu kroků musí skončit

- **Determinovanost** – pro tentýž vstup musí algoritmus poskytovat týž výstup. (V každém kroku je zřejmé, co bude následovat.)

## Zápisy algoritmů

- V přirozeném jazyce – nešikovné, nepřehledné
- Graficky – vývojové diagramy, struktogramy – přehledné, ale někdy zbytečně pracné
- V programovacím jazyce – pokud si zvykneme na přehledný zápis, nejobvyklejší

### Vývojový diagram řešení lineární rovnice $a x = b$ :



### Program je zápis algoritmu v programovacím jazyce.

Tyto jazyky jsou odvozeny z jazyka přirozeného (většinou angličtiny), ale mají jednoznačnou a jednoduchou gramatiku.

Aby procesor podle programu mohl pracovat, potřebuje překladač, který překládá z programovacího jazyka do instrukcí strojového kódu. (Ve skutečnosti složitější)

**Kompilátory** – přeloží celý program najednou (rychlejší)

**Interprety** – překládají po příkazech (Interpretovány bývají příkazy OS)

### Zásady tvorby algoritmů:

- Důkladná analýza problému
- Algoritmický problém nemusí mít vždy řešení (A je to dobře, protože jinak by nás počítače brzy k ničemu nepotřebovaly)

- Studium literatury
- Metoda shora dolů
  - Rozklad problému na podproblémy
  - Zpracování částí
  - Jejich složení do původního problému
- Zápis v programovacím jazyce
- Ladění
- Dokumentace

## Pracovní list

### Cvičení

Vyzkoušejte si vyřešit několik příkladů z přijímacích zkoušek na Informatiku na Matematicko-fyzikální fakultě v Praze.

1. V algebrogramu nahradte písmena číslicemi, aby platila rovnost v řádcích i sloupcích. Každé písmeno nahradte jednou číslicí (cifrou), různým písmenům odpovídají různé číslice. Uveďte postup úvah, který vede k řešení. (Najděte všechna řešení)

$$A + B = BC$$

— — —

$$D + B = E$$


---

$$E + C = E$$

2. Obdélníková tabulka čokolády má velikost  $m \cdot n$  dílků. Tabulku chceme rozlámat na jednotlivé dílky. Každý zlom musí být rovný a rozlomíme jí jen jeden kus čokolády. Jaký nejmenší počet zlomů potřebujeme? Zdůvodněte.

3. Máme devět na pohled stejných kuliček a víme, že osm z nich je stejně těžkých, zatímco jedna je těžší než ostatní. Navrhněte postup, jak na rovnoramenných vahách naleznete tuto kuličku s nejmenším počtem vážení. Zdůvodněte.

4.\* Je dána posloupnost cifer 1 2 3 4 5 6 7 8 9 .

Před každou cifrou vložte znaménko plus nebo mínus, aby výsledný aritmetický výraz měl hodnotu 33. Nalezněte všechna řešení a vysvětlete, proč další řešení neexistuje.

5.(\*) Herní plán hry je tvořen osmi poli v řadě za sebou. Na začátku jsou první čtyři pole černá a druhá čtyři bílá. Jedním tahem se současně změní barvy jedné dvojice sousedních políček. Navrhněte posloupnost tahů, po níž budou:

- a. Všechna políčka bílá
- b. První černé a ostatní bílá
- c. Střídavě černé a bílé
- d. Pokud něco není možné, zdůvodněte

6. Zapište algoritmus nějaké činnosti. Soustředte se na vstup, výstup a přesnou definici popisovaných akcí.

### Řešení

1. C je zřejmě 0, aby fungoval součet jednociferných čísel, musí být B 1. Pak A je 9,

$10 = 2E$ , tedy E je 5 a D je 4.

$$9 + 1 = 10$$

- - -

$$4 + 1 = 5$$

---

$$5 + 0 = 5$$

2. Každým zlomem se počet kusů zvýší o 1. Na začátku je jeden kus, po kompletním rozlámání  $n \cdot m$  kusů. Proto potřebujeme  $n \cdot m - 1$  zlomů, na jejich pořadí nezáleží.

3. Kuličky rozdělíme na tři trojice, dvě trojice porovnáme. Buď je trojice obsahující nejtěžší kuličku jedna z nich, nebo ta třetí. (První vážení)

Z vítězné trojice vezmeme dvě kuličky a porovnáme stejně. (Druhé vážení)

4. Součet čísel je 45. Má-li být výsledek o 12 menší, musí součet čísel, před kterým je mínus, dát 6..

Možnosti: 1, 2, 3, 2, 4, 6, 1, 5

$$(x + y = 45, x - y = 33, 2x = 12)$$

5. Očíslujeme políčka zleva doprava čísly 1 až 8, tah popisujeme číslem levého políčka:

- a. Dva tahy 1, 3
- b. Nelze, protože libovolný tah zachovává sudost (případně lichost) bílých (i černých) polí.
- c. Čtyři tahy 2, 3, 5, 6

## 6. Individuální řešení