



**Střední škola pedagogická, hotelnictví a služeb, Litoměřice, příspěvková organizace**

|                   |                                     |
|-------------------|-------------------------------------|
| <b>Předmět:</b>   | <b>Algoritmizace a programování</b> |
| <b>Téma:</b>      | <b>Algoritmy</b>                    |
| <b>Vyučující:</b> | <b>Ing. Milan Káža</b>              |
| <b>Třída:</b>     | <b>EK2</b>                          |
| <b>Hodina:</b>    | <b>3</b>                            |
| <b>Číslo:</b>     | <b>V/5</b>                          |

## Programování v jazyce C a C#

### 1. Obecné pojmy

#### 1.1. Algoritmus

Algoritmus je návrh, postupu řešení dané úlohy, který se nejčastěji používá ve výpočetní technice při programování. Algoritmus musí být univerzální řešení a není možné ho vázat na jeden programovací jazyk.

Úkolem algoritmu je rozložení problému na jeho jednoduché části a ty se pak postupně řeší. Každý algoritmus musí splňovat určitá pravidla:

Vstupní data (např. *Vyhledej jméno*)

Výstupní data (*Jméno bylo nalezeno*)

Výstupem na operaci může být i chybové hlášení. Algoritmus je možné ale i zobecnit a sestavit např. na denní režim.

Každý algoritmus musí splnit tyto podmínky, musí být:

- **Konečný** --Algoritmus musí skončit v konečném počtu kroků. Tento počet kroků může být libovolný a metody, které nejsou konečné jsou metody početní.
- **Deterministický** --Každý krok musí být přesně definován a vždy musí být jasné co má algoritmus vykonat.
- **Efektivní** --Algoritmus musí být efektivní, tak aby jeho použitelnost byla univerzální.
- **Obecný** --Algoritmus neřeší jeden konkrétní problém, ale řeší ho obecně.
- **Rezultivní** --Algoritmus při zadání vstupních dat vždy vrátí nějaký výsledek (může se jednat i jen o chybové hlášení)

## 1.2. Algoritmus- všeobecné využití

Využití algoritmu pro sestavení denního režimu sportovce, který se připravuje na Mistrovství republiky. Sportovec cvičí ve sportovním odvětví jako jednatel např. gymnasta. Z tohoto důvodu se sestavuje tento program jen pro daného sportovce.

07:00 - budíček  
07:15 - 08:00 - lehká rozvička- zaměření strečing a uvolnění svalů  
08:00 - 09:30 - snídaně a relaxace po snídani  
09:30 - 11:00 - kondiční a vytrvalostní trénink  
11:00 - 12:30 - procvičení vlastních sestav  
13:00 - 14:00 - oběd  
14:00 - 15:00 - relaxace, masáž, klid  
15:00 - 17:00 - silový trénink  
17:00 - 18:30 - samostatný trénink  
19:00 - 20:00 - večeře  
20:00 - 21:00 - samostatný trénink volný procvičení určitých částí, podle potřeby  
21:00 - 22:00 - odpočinek, sauna, uvolnění  
22:30 - spánek

Jak je vidět tak je zde jeden den, každý den v týdnu se pak přesně rozpracovává tedy do nejmenších detailů. Přesně tento princip využívají i algoritmy při návrhu řešení programů na PC, ale je nutné aby se mohla provádět volba, která je v některých případech nutná.

Podobně se může vypracovat plán letního tábora pro děti, ale tam musí být volba např. v úterý výlet na hrad, ale tento den je počasí na koupání tak se plán mění a děti se jdou koupat a ve čtvrtek kdy se měly děti jít koupat je ošklivě tak se jde na hrad.

Tedy některé přípravy jsou jednoznačné jiné je nutné volit a to se musí objevit i v algoritmech programů.

### 1.3. Editor

Prostředí pro psaní kódu programu, v něm se zadávají příkazy jazyka v kterém se tvoří program. Dnes jsou prostředí vytvářena pod Windows nebo Linuxem, kde je spousta „pomůcek“, které programátorovi usnadňují činnost a práci.

### 1.4. Preprocesor

Součást překladače, který předzpracovává zdrojový kód, pro snadnější činnost překladače. Vynechává komentáře, připravuje hlavičkové soubory, rozvíjí makra.

### 1.5. Compiler

Také zvaný překladač, ten provádí překlad ze zdrojového do relativního kódu tím vzniká soubor s příponou OBJ. Další produkt překladače je výpis o chybách.

### 1.6. Linker

Také sestavovací program, sestaví z relativního kódu, kód spustitelný ten má příponu EXE.

### 1.7. Debugger

To je ladící program, který slouží k ladění a vyhledávání chyb programu. Program se ve vývojovém prostředí i spustí, pokud v něm není chyba. Celý debugger běží tímto způsobem:

*editor* → *compiler* → *linker* → *debugger*

V případě nalezení chyby vývojové prostředí oznámí, kde je chyba, některá ji i označí, po jejím odstarnění se spustí celý proces znovu.

## 1.8. Bit a bajt

Bit je základní jednotka informace a nabývá dvou hodnot logické jedničky (true) nebo pravda a logické nuly (false) také nepravda. S bitem se počítá ve dvojkové soustavě.

Bajt se skládá z osmi bitů a s bajty se také pracuje jednodušeji i se lépe převádí. S bajty se počítá v šestnáctkové (hexadecimální) soustavě.

Kontrolní otázky:

1. Vysvětlete pojem algoritmus a na příkladu ukažte jeho využití.
2. Vysvětlete rozdíl mezi bitem a bajtem.

*Použitá literatura:*

*Miroslav Virius: C# pro zelenáče, 1.vydání, Neocortex, s.r.o Praha, 2002*

*Pavel Herout: Učebnice jazyka C, 3.vydání, KOOP České Budějovice, 1997*

*Miroslav Virius: od C++ k C#, 1.vydání, KOOP České Budějovice, 2002*