

21. Procedúry bez parametrov.

- Vysvetlite použitie a štruktúru procedúry bez parametrov, s lokálnymi objektami a bez lokálnych objektov.
- **Úloha:** Zostavte program na sčítanie celých čísel pomocou pripočítavania jednotky, kde na pripočítavanie použijete procedúru Nasledovník, ktorej úlohou je dané číslo zväčšiť o 1.

Pri písaní programu sa často stáva, že niektoré príkazy, lepšie povedané sústava príkazov sa opakuje. Aby sme sa tomu vyhli, sprehľadnili program a hlavne uľahčili si prácu môžeme takéto príkazy zlúčiť do jedného a dať mu nové meno. Programátor tým vlastne vytvorí nový príkaz. Toto nám poskutujú procedúry (funkcie). Procedúra je vlastne taký malý program v programe. Takže až na malé zmeny má tú istú štruktúru.

Procedúry a funkcie predstavujú **podprogramy**. Podprogramy sú základným nástrojom hierarchického členenia programu. Tvoria do istej miery samostatné, na hlavnom programe nezávislé celky. Pri správnom použití sa zlepši **prehľadnosť kódu** a zmenší veľkosť programu.

Činnosť, ktorú má podprogram vykonávať, spôsob komunikácie a mechanizmus volania je predpísaný v jeho deklarácii.

TurboPascal používa dve triedy podprogramov:

procedúry - komunikujú výhradne pomocou svojich parametrov, príp. globálnych premenných

funkcie - priamo vracajú istú hodnotu

Podprogramy je výhodné použiť:

ak sa v programe vyskytuje rovnaká postupnosť príkazov viackrát

ak potrebujeme zvýšiť prehľadnosť programu

ak sa v programe vyskytuje podobná postupnosť príkazov odlišná len parametrami

riešenie je rekurzívne, t. j. používa "sám seba"

Procedúry a funkcie sa uvádzajú v deklaračnej a definičnej časti za deklaráciou premenných:

Procedúry bez parametrov

Procedure MenoProcedury;

var{deklarácia lokálnych premenných};

begin

....príkazy podprogramu;

end;

begin

....príkazy hlavného programu;

MenoProcedury; {volanie procedúry jej menom}

....príkazy hlavného programu;

end.

program scitanie;

uses crt;

var a,b,i:integer;

procedure nasledovnik;

begin

a:=a+1;

end;

begin

clrscr;

writeln('Zadaj cisla ktore sa maju scitat');

readln(a,b);

for i:=1 to b do nasledovnik;

writeln('vysledok je ',a);

readkey;

end.

Tabulkový kalkulátor II.

- Vysvetlite použitie vzorcov a odkazov v MS Excel.
- Diskutujte možnosti grafickej interpretácie hodnôt v tabuľke.
- **Úloha:** Pripravte tabuľku hodnôt kvadratickej funkcie $f: y = x^2 - 5x + 8$ pre $-4 \leq x \leq 6$ s krokom 0.5. Výsledok interpretujte graficky. Použite aspoň dva druhy grafov.

Vzorce tvoria neoddeliteľnú súčasť takmer každej tabuľky. Hodnota výsledku daného vzorca je aktualizovaná po každej zmene údajov, na ktoré sa vzorec odvoláva. Po vložení vzorca do bunky sa v bunke zobrazí výsledok daného vzorca, jeho tvar môžeme vidieť vo vstupnom riadku (tzv. vzorcový panel).

Vlastné vzorce

Začínajú sa znakom =, môžu obsahovať adresy buniek (do vzorca sa však dosadí hodnota odkazovanej bunky), znamienka matematických operácií: +, -, *, /, ^ (súčet, rozdiel, súčin, podiel, umocnenie). Príklad funkcie: =D5*E5 - vynásobí cenu jedného kusu výrobku počtom výrobkov

Štandardné vzorce

Zoznam vzorcov vyvoláme pomocou ikony v štandardnom paneli, alebo príkazom VLOŽIŤ/FUNKCIA... . Vzorce sú rozdelené do kategórií, výhodná je skupina najčastejšie používaných vzorcov - posledne použité, prípadne v kategórii všetky nájdeme abecedný zoznam všetkých vzorcov. (Tabuľka obsahuje aj návod o význame vzorcov, ako aj ich správny zápis) Najčastejšie používaná funkcia SUMA - súčet hodnôt rozsahu buniek - má ikonu v štand. paneli. Príklad funkcií: =SUM(F5:F12) - sčíta hodnoty v bunkách F5 až F12; =AVERAGE(E5:E12) - vypočíta priemernú hodnotu z hodnôt buniek E5 až E12

Adresovanie

Pri výpočte celkovej ceny v stĺpci F v našej úlohe stačí vzorec vytvoriť len v bunke F5. Do ostatných buniek v stĺpci tento vzorec nakopírujeme. Súradnice buniek, na ktoré sa vzorec odkazuje, sa prispôbia polohe vzorca v tej ktorej bunke. Takéto súradnice nazývame relatívne napr. F5, E14. Ak chceme, aby sa všetky vzorce odkazovali len na nejakú konkrétnu bunku, musíme zadať do vzorca jej tzv. absolútnu adresu napr. \$F\$5, \$E\$14. Existujú ešte tzv. zmiešané súradnice napr. F\$5 (riadok sa nemení), \$E14 (stĺpec sa nemení)

22. Procedúry s parametrami

- Vysvetlite: použitie a štruktúru procedúry s parametrami, deklarácia parametrickej procedúry, úloha parametrov.
- **Úloha:** Napíšte program, ktorý obsahuje procedúru s parametrom a ktorý vypíše všetky párne prirodzené čísla z intervalu, ktorého hranice budú zadané z klávesnice.

Vstup: hranice intervalu

Výstup: jednotlivé párne čísla z intervalu načítaného na vstupe

Štruktúra procedúry s parametrom:

```
procedure nazov_procedury(parameter: typ_prem.);
```

```
.....{deklaračná časť procedúry,}
```

```
.....{napr. deklarácia lokálnych premenných}
```

```
begin
```

```
prikaz 1..n;
```

```
end;
```

Parameter je v podstate akýsi vstup (prípadne i výstup - viď ďalej) do procedúry, ktorým odovzdávame procedúre hodnotu potrebnú k vykonaniu algoritmu v podprograme. Procedúry s parametrom môžu mať aj viac parametrov. **Parametre toho istého typu sa oddeľujú čiarkou (Príklad: procedure strana (a,b: integer);), parametre rôzneho typu sa oddeľujú bodkočiarkou. (Príklad: procedure strana (a,b: integer; c: real);).** Parametre v mieste deklarácie procedúry nazývame **formálne parametre**, parametrov mieste volania procedúry sú **skutočné parametre**, ktoré priradujú hodnoty formálnym parametrom. Programovací jazyk Pascal pozná dva druhy parametrov: Parameter nahradzovaný **hodnotou** (PNH) Parameter nahradzovaný **odkazom** (PNO) **Procedúra s parametrom nahradzovaným hodnotou** pracuje tak, že pri volaní procedúry z hlavného programu sa skutočný parameter (jeho hodnota) prekopíruje do formálneho parametra v tele procedúry a ich prepojenie je ukončené. To znamená, že ak sa v procedúre zmení hodnota formálneho parametra, **táto zmena sa neprejaví** na skutočnom parametri (premennej) v hlavnom programe. Spôsob deklarácie týchto parametrov je uvedený vyššie.

Procedúra s parametrom nahradzovaným odkazom pracuje tak, že pri volaní procedúry z hlavného programu sa skutočný parameter (jeho hodnota) prepojí s formálnym parametrom v tele procedúry a toto prepojenie trvá počas vykonávania procedúry. To znamená, že každá zmena hodnoty formálneho parametra sa prejavuje aj na skutočnom parametri (premennej) v hlavnom programe. Táto zmena zostáva zachovaná i po skončení procedúry. PNO sa využívajú ako výstupy z procedúry. Deklarácia procedúry obsahujúcej PNO: procedure delenie(a,b:integer; var c:real);(a,b -> PNH, c -> PNO)

```
program parne;
uses crt;
var a,b:integer;
procedure vypis(x,y:integer);
var i,d:integer;
begin
for i:=x to y do
begin
d:=i mod 2;
if d=0 then writeln(i);
```

```
end;
end;
begin
clrscr;
writeln('zadaj hranice intervalu');
readln(a,b);
writeln('tieto cisla su parne');
vypis(a,b);
readkey;
end.
```

Tabulkový kalkulátor III.

- Použitie štandardných funkcií. Funkcie štatistické, matematické a iné. Simulovanie náhodných výsledkov. Filtrovanie.
- **Úloha:** Vygenerujte do tabuľky 50 náhodných čísel 1-100. Utvorte z nich 4 tabuľky s hodnotami náhodných čísel 1-25, 26-50, 51-75, 76-100. Určte aritmetické priemery čísel v týchto štyroch tabuľkách.

Pre urýchlenie práce v tabuľkách sa využívajú automatické funkcie. Už názov **automatické** napovedá o tom, že ide o niečo, čo bude robiť Excel automaticky. Pri používaní automatických funkcií nemusíte zadávať do bunky znamienko rovná sa. Výhodou používania automatických funkcií je najmä ich rýchlosť. Medzi základné automatické funkcie patrí **súčet, priemer, minimum, maximum a počet**.

Najčastejšou automatickou funkciou v Exceli je **automatický súčet**. Postup pri používaní automatických funkcií je veľmi podobný tvorbe vlastných vzorcov. Nemusíte tu však vpisovať do bunky znamienko = (rovná sa), pretože automatická funkcia si ho zadá automaticky. Matematické a štatistické funkcie. Tieto dva druhy funkcií sa využívajú najviac. Ak sa pozriete na kategóriu **Matematických funkcií**, zbadáte, že pod túto kategóriu spadá 51 funkcií. Okrem využiteľných funkcií v bežnej práci, sú tu funkcie trigonometrické, teda určené na prácu s kosínusom, sínusom, atď. Pod kategóriu **Štatistické funkcie** spadá 80 funkcií využiteľných pri práci s údajmi. Opäť nie všetky budú využiteľné pri **bežnej** práci.

Matematické funkcie. V tejto časti sa oboznámite s matematickými funkciami, ktoré sú využiteľné pri bežnej práci s údajmi.

=ABS(číslo) Táto funkcia vráti absolútnu hodnotu čísla. Je to to isté číslo, ale bez znamienka. Argumentom tejto funkcie je jedno konkrétne číslo, alebo odkaz na bunku, v ktorej je hodnota. Takže do pola pre argument vpíšete buď priamo

=POWER(číslo; exponent) V tejto funkcii ide o umocňovanie čísel na n-tú. Prícom n je exponent v argumentoch funkcie. Použitie tejto funkcie je diskutabilné, pretože aj pomocou vlastného vzorca dokážete vypočítať mocninu čísla ($=12^3$).

=SQRT(číslo)

Táto funkcia vypočíta druhú odmocninu z čísla. Argumentom môže byť konštanta, alebo odkaz na bunku, ktorej obsahom je číslo.

=SUM(rozsah buniek A1:B5)

Táto funkcia sčítava čísla. Argumentom môže byť konštanta, odkaz na bunky, ktoré obsahujú čísla, alebo rozsah buniek. K funkcii **SUM** existuje ešte niekoľko variant.

=SUMIF(číslo; kritérium)

Štatistické funkcie.

V tejto časti sa oboznámite s použiteľnými štatistickými funkciami na spracovávanie údajov.

=AVERAGE(číslo; číslo; číslo;...)

Táto funkcia vypočíta priemernú hodnotu čísel. Argumentom môžu byť konštanty, odkazy na bunku, alebo rozsah buniek.

=COUNT(číslo; číslo; číslo;...)

Táto funkcia spočítava nie hodnoty, ale zisťuje počet vyplnených buniek. Argumentom môžu byť konštanty, odkazy na bunky, rozsah buniek.

=MAX(číslo; číslo; číslo;...)

Pomocou tejto funkcie zistíte najväčšie číslo (**maximum**) v rozsahu buniek, ktoré ste označili. Argumentom môžu byť konštanty, odkazy na bunky, rozsah buniek.

=MIN(číslo; číslo; číslo;...)

Pomocou tejto funkcie zistíte najmenšie číslo (**minimum**) v rozsahu buniek, ktoré ste označili. Argumentom môžu byť konštanty, odkazy na bunky, rozsah buniek.

Filtrovanie údajov v tabuľke.

Dalšou funkciou pri práci s tabuľkou je filtrovanie. Je to rýchly a jednoduchý spôsob, ako vyhľadať a spracovať podmnožinu údajov v tabuľke. Filtrovaný rozsah zobrazí iba riadky, ktoré spĺňajú určité **kritériá** pre stĺpec. Kritérium je podmienka, ktorú zobrazované údaje musia spĺňať. Tieto podmienky si zadávate vy. Program Microsoft Excel obsahuje dve príkazy na filtrovanie rozsahov. Je to **Automatický filter** pre jednoduché filtrovanie a **Rozšírený filter** pre zložitejšie filtrovanie. Počas filtrovania na rozdiel od zoradovania nedochádza k zmene usporiadania údajov. Filtrovanie dočasne skryje riadky, ktoré nechcete zobrazovať. Vyfiltrované údaje môžete upravovať, formátovať, vytvárať grafy a tlačit ich bez toho, aby ste ho premiestnili, alebo zmenili jeho usporiadanie. Skôr ako sa rozhodnete zapnúť automatický filter, musí byť aktívna jedna bunka v tabuľke. To zabezpečí, že sa filter použije pre všetky údaje. Ak by ste mali označené dve a viac buniek, filter sa zapne na tejto označenej oblasti buniek. Automatický filter zapnete tak, že využijete buď skupinu **Domov**, kde sa príkaz na zoradenie nachádza na konci celej skupiny, alebo na to využijeme samostatnú skupinu **Údaje**, kde sú ikonky filtrovania priamo k dispozícii. Toto spôsobí, že napravo od všetkých položiek záhlavia sa objavia šípky. Ak kliknete na niektorú šípku, rozvinie sa pole, v ktorom sa zobrazia všetky hodnoty vyskytujúce sa v príslušnom stĺpci. Okrem toho sa tam zobrazia aj ďalšie možnosti automatického filtra. Môžete údaje zoradiť vzostupne, alebo zostupne, môžete zobrazovať všetko, alebo iba prvých desať.

Simulovanie v exceli

Metódu počítačovej simulácie, ktorá je svojím charakterom blízka fyzikálnym experimentom a zároveň patrí medzi metódy numerickej matematiky, nazývame Monte Carlo. Metóda dostala názov podľa Monte Carla, známeho svojimi kasínami a najmä ruletou.

Všeobecný postup výpočtu metódou Monte Carlo je nasledujúci: 1. Vygenerujeme náhodné čísla y_i v intervale (0,1) s pravidelným rozdelením. 2. Náhodné čísla y_i pretransformujeme na náhodné čísla z_i s požadovaným rozdelením. 3. Prostredníctvom zís-

kaných náhodných čísel z_i buď priamo výpočtom odhadujeme charakteristiky náhodnej veličiny x , alebo odhady a hodnoty x_i vyrátame pomocou vhodného algoritmu. 4. Dosiahnuté výsledky sa následne štatisticky spracujú.

Na simuláciu modelov rôznych náhodných procesov je najlepšie použiť široko dostupný nástroj od Microsoftu - Excel. Na jednoduchých príkladoch, ktoré riešia finančné otázky, si ukážeme, ako si MS Excel dokáže poradiť s náhodnými číslami a pravdepodobnosťou. Keď do bunky vpišeme funkciu =RAND(), dostaneme náhodné číslo s rovnomernou distribúciou, ktoré môže predstavovať ľubovoľnú hodnotu z intervalu (0,1). Preto pri približne 25-percentnej pravdepodobnosti dostaneme číslo menšie, nanejvýš rovné 0,25; pri 10-percentnej pravdepodobnosti je to číslo, ktoré je väčšie ako 0,90, a podobne. Nasledujúce obrázky demonštrujú, ako funkcia Rand pracuje. Na obrázku č. 1 vidíme výsledky funkcií a obrázok č. 2 zobrazuje vzorce a funkcie používané pri metóde Monte Carlo.

23. Funkcie bez parametrov

- Definícia funkcie, štruktúra funkcie bez parametrov, s lokálnymi objektmi a bez lokálnych objektov.
- **Úloha:** Napíšte program, ktorý vypočíta obsah trojuholníka zo zadanej strany a výšky a zistí, či je z intervalu (d,h). Na výpočet obsahu trojuholníka použite funkciu.
Vstup: a, v, d, h. Výstup: s - obsah, oznam: „patrí“, „nepatrí“.

Funkcia je druh podprogramu a na rozdiel od procedúry nám dáva iba jednu výslednú hodnotu. Deklarácia funkcie pozostáva z hlavičky a tela funkcie. Telo by malo obsahovať aspoň jeden priradovací príkaz pre premennú, ktorá má rovnaký názov ako funkcia.

Funkcia sa používa vtedy, ak potrebujeme zostaviť čiastkový algoritmus, ktorého volaním sa získava funkčná hodnota (podobne ako pri funkciách na kalkulačke, v matematike,...). Získaná funkčná hodnota sa ukladá do identifikátora (mena) funkcie. V jej príkazovej časti musí byť teda najmenej jeden priradovací príkaz, na ľavej strane ktorého je identifikátor funkcie priradený.

```
function názov_funkcie;  
    deklarácia lokálnych premenných;  
begin  
    príkazy funkcie a medzi nimi aj nasledujúce priradenie;  
    názov_funkcie:=výraz;  
end;
```

```
program trojuholnik;  
uses crt;  
var a,v,d,h:real;  
function obsah:real;  
begin  
    obsah:=a*v/2;  
end;  
begin  
    clrscr;  
    writeln('zadaj stranu a vysku trojuholnika');  
    readln(a,v);  
    writeln('zadaj hranice intervalu');  
    readln(d,h);  
    writeln('obsah trojuholnika je ',obsah:2:3);  
    if (obsah>d) and (obsah<h) then writeln('patri do intervalu') else writeln('nepatri do intervalu');  
    readkey;  
end.
```

WWW stránky.

- WWW stránky, zdrojový text stránky. Vysvetli pojem tag - párové, nepárové, písanie nadpisov, pozadie, text, obrázky a odkazy v jazyku HTML. Webové editory. Umiestňovanie WWW stránok na Internet.
- **Úloha:** Vytvorte ukážku jednoduchej web stránky, nastav pozadie, farbu textu, veľkosť písam a utvorte linku na www.gymml.sk. Overte funkčnosť stránky.

World wide web - podstatou www sú hypertextové dokumenty HTML súbory (Hyper text markup language), ktoré umožňujú vzájomné previazanie spojenie link jednotlivých stránok dokumentov page pomocou odkazov. Najznámejšími prehliadačmi sú Microsoft Explorer, Google Chrome, Safari a Firefox Mozilla. WWW pracuje na princípe klient/server. Na nájdenie www stránky potrebuje prehliadač poznať jej adresu nazývanu URL. - uniform resource locator jednotné označenie zdroja. URL má tento všeobecný formát protokol://server-name(:port))/path

Môžu byť:

- **Statické** (vytvorené a uložené na ww serveroch, ktoré ich na základe požiadavky rozosiľajú).
- **Dynamické** (vytváranie až na základe komunikácie s užívateľom, ktorý môže konkretizovať svoje požiadavky na ich obsahovú štruktúru).

Na popis WWW stránok môžu mať vo všeobecnosti dva tvary:

Párové príkazy (tvorené otvárajúcim a uzatvárajúcim príkazom, ktoré sa líšia len tým, že uzatvárajúci príkaz obsahuje pre svojím menom znak / (lomítko)

Nepárové príkazy (tvorené jediným príkazom)

Zdrojový kód dokumentu HTML je jednoduchý text, ktorý sa dá prehliadať i upravovať v jednoduchom textovom editore.

Tagy = značky (sú uzavreté v <ostrých zátvorkách>

24. Funkcie s parametrami

- Definícia funkcie s parametrom, úloha parametrov vo funkcii.
- **Úloha:** Zostavte program na výpis všetkých prvočísel, ktoré sú menšie N. Na zistenie, či číslo je alebo nie je prvočíslo použijte funkciu s parametrom.

Vstup: N

Výstup: prvočísla menšie ako N

Napr. Vstup N=8 Výstup: 2, 3, 5, 7

Funkcia s parametrom: Funkcia dáva ako výsledok jednu hodnotu, táto hodnota musí byť priradená v tele funkcie do premennej rovnakého mena ako má funkcia. Využíva sa, keď niekedy potrebujeme funkciou poslať nejaké hodnoty, napríklad obsahy premenných.

program fprvocislo;

uses crt;

var n:integer;

function p(f:integer):integer;

var d,i,j,poc:integer;

begin

for i:=2 to f-1 do begin

poc:=0;

for j:=1 to i do begin

d:=i mod j;

if d=0 then poc:=poc+1;

end;

if poc=2 then writeln(i);

end;

end;

begin

clrscr;

writeln('zadaj cislo');

readln(n);

writeln('prvocisla mensie ako zadane cislo');

p(n);

readkey;

end.

Možnosti tvorby prezentácií.

- Vysvetlite: význam a použitie počítačovej prezentácie, pojem snímka, popíšte prostredie programu MS Power Point, diskutujte zásady zostavenia a použitia prezentácie
- Úloha: Spracujte krátku prezentáciu (3-5 snímok) na ľubovoľnú tému. Zabezpečte automatické prehrávanie prezentácie. Vhodný text a netextové objekty vyhľadajte na internete.

Vysvetlite: význam a použitie počítačovej prezentácie, pojem snímka, popíšte prostredie programu MS Power Point, diskutujte zásady zostavenia a použitia prezentácie

Počítačová prezentácia : je obsahovo a formálne ucelená séria snímok prezentujúca a vysvetľujúca určitý problém. Sú to snímky, ktoré sa postupne premietajú na stenu alebo plátno pomocou dataprojektora. Využíva sa pri konferenciách pri prednáškach ...

Snímka: je jedna obrazovka s ľubovoľnými vloženými objektami..

25. Číselné typy a operácie s nimi.

- Charakterizujte: číselné typy údajov v Turbo Pascale, rozsahy hodnôt pre číselné typy, funkcie pre číselné typy, výsledky operácií a číselné typy.
- Úloha:** Vytvorte program, ktorý načíta váš vek (počet rokov, mesiacov, dní) a vypíše koľko to je dní a koľko to je mesiacov a koľko to je rokov. (počítajte: 1 rok je 12 mesiacov alebo 365 dní, 1 mesiac je 30 dní)
Vstup: r, m, d (napr: 17, 4, 10)
Výstup: 17,36 rokov
208,33 mesiacov
6335 dní

Pre celé čísla používa Pascal tieto údajové typy:

- byte - rozsah: 0 .. 255, veľkosť 1B
- word - rozsah: 0 .. 65 535, veľkosť 2B
- shortint - rozsah: -128 .. 127, veľkosť 1B
- integer - rozsah: -32 768 .. 32 767, veľkosť 2B
- longint - rozsah: -2 147 483 647 .. 2 147 483 647, veľkosť 4B

Operácie nad celočíselnými údajovými typmi:

- sčítanie: +
- odčítanie: -
- násobenie: *
- celočíselné delenie: div
- zvyšok po delení: mod
- absolútna hodnota: abs(p)
- druhá mocnina: sqr(p)
- určenie nepárnosti: odd (p) - výsledok je true, ak je číslo nepárne, alebo false, ak je párne

Ak by ste na delenie použili operátor /, potom výsledok nemôžete uložiť do celočíselnej premennej, pretože výsledkom je racionálne číslo!

```
program vek;
uses crt;
var r,m,d:integer;
    vr,vm,vd:real;
begin
clrscr;
writeln('zadaj tvoj vek - pocet rokov, mesiacov, dni');
readln(r,m,d);
vr:=r+m/12+d/365;
vm:=m+r*12+d/30;
vd:=d+r*365+m*30;
writeln('Je to ',vr:1:2,' rokov, ',vm:1:2,' mesiacov, ',vd:1:0,' dni');
readkey;
end.
```

Informačná spoločnosť.

- Aké sú súčasné trendy informačno- komunikačných technológií, limity, riziká, kladné a záporné stránky informačnej spoločnosti.
- Vysvetlite úlohy zamestnancov softvérových firiem - funkcia analytik, programátor, dizajnér, animátor, testovač

Analytik- Analytik zabezpečí nasledovné činnosti na softvérovom projekte:

- návrh architektúry a dizajnu,
- analýzu riešenia,
- komunikáciu s klientom pri zisťovaní požiadaviek,
- špecifikáciu riešenia,
- návrh riešenia,
- návrh a definíciu testov a testovacích scenárov,
- asistenciu pri implementácii a integrácii SW.

Programátor - je ten, kto postupom zvaným programovanie implementuje výsledky návrhu projektu.

Dizajnér Animátor

Testovač je ten, kto vykonáva Testovanie softvéru - je súbor procesov slúžiacich na kontrolu kvality softvérového produktu, ktorých cieľom je dosiahnutie požadovanej kvality softvéru z hľadiska funkčnosti, spoľahlivosti, výkonnosti, použiteľnosti a podporovateľnosti. Kontrola kvality sa môže uskutočniť či už pre jednotlivé časti informačného systému alebo pre systém ako celok.

26. Znakový typ a operácie s ním.

Tabuľka kódov ASCII, deklarácia znakového údajového typu v Pascale, funkcie pre prácu so znakmi v Pascale.

Úloha: Zostavte program, ktorý bude pre zadaný znak z klávesnice udávať jemu zodpovedajúci kód. Program rozšírite o výpis všetkých znakov z určitého intervalu kódov.

Vstup: znak. Výstup: ASCII kód znaku zadaného na vstupe.

V druhej časti : Vstup : hranice intervalu, Napr. 63, 75 . Výstup: príslušné znaky

ASCII je anglická skratka pre American Standard Code for Information Interchange (americký štandardný kód pre výmenu informácií). Ide o kódovací systém znakov anglickej abecedy, číslíc, iných znakov a riadiacich kódov. Jednoznačne najpoužívanejším kódom v súčasnosti je ASCII kód. Ide o osembitový kód, ktorým je možné zobrazit' teda 256 rôznych znakov. Táto abeceda obsahuje čísllice 0-9, písmenka veľkej a malej abecedy, znamienka a rôzne znaky, riadiace znaky a semigrafické znaky. Identifikátor CHAR:

Char

rozsah #0...#255 (binárne kódy znakov).

hodnoty (znak pracovnej abecedy)

funkcie ORD (určí kód znaku ord(a)=65) alebo CHR (opačná funkcia - určí kód čísla chr(65)=a)

operácie: predchodca - pred (p)

následník - succ (p)

ordinálna hodnota - ord (p)

zvýšenie alebo zníženie ordinálnej hodnoty o 1 - inc (p) / dec (p)

program ascii;

uses crt;

var a,z:char;

b,c,d,i:integer;

begin

clrscr;

writeln('zadaj lubovolny znak');

readln(a);

b:=ord(a);

writeln('ascii kod zadaneho znaku je ',b);

writeln('zadaj interval ascii kodov');

readln(c,d);

for i:=c to d do begin

z:=chr(i);

writeln(z);

end;

readkey;

end.

Softvér a jeho triedenie.

- Triedenie softvéru, Autorské práva. Typy softvéru z hľadiska autorských práv. Netiketa.
- **Úloha:** Vyhľadajte na internete krátky freewerový softvér a inštalujte ho vo svojom PC. Demonštrujte jeho použitie.

Software je programové vybavenie počítača alebo súhrn všetkých programov, ktoré sa dajú použiť na výpočtovom zariadení.

Rozlišujeme ho na:

- **systémový software** - zahŕňa programy, ktoré sú potrebné pre korektný beh počítača, ako aj programy, ktoré podporujú tvorbu programov (prekladače a testovacie programy) a poskytujúce všeobecné služby (formát disku).
- **aplikačný software** - slúži na riešenie problémov používateľov, napr. pre účtovníctvo alebo riešenie integrálov. Ide o ľahko zmeniteľné časti počítača.

Typy softwaru z hľadiska autorských práv:

- o **komerčný software** - je každý softvér, ktorý jeho autor zaregistroval v súlade s uvedeným zákonom ako komerčný. Plne sa na neho vťahujú autorské práva.
- o **shareware** - je taký softvér, ktorý je buď v plnej verzii alebo v čiastočnej (viazané na určitú dobu).
- o **freeware** - je softvér, ktorý jeho autor určil na voľné komerčné aj nekomerčné použitie všetkým užívateľom pričom sa vzdali nároku na finančné vyrovnanie. Autorské práva freewaru sú však ponechané jeho autorovi, preto ho užívatelia nemôžu modifikovať ani vkladať do vlastných produktov.
- o **public domain** - je podobne ako freeware ponechaný verejnosti na voľné používanie, ale oproti freewaru s tým rozdielom, že sa jeho autori úplne vzdali autorských práv a dali každému užívateľovi právo tento softvér ľubovoľne modifikovať či vkladať ako súčasť do vlastných produktov.

Netiketa = pravidlá slušnosti na internete (odvodený z anglických slov net (sieť) a etiquette (etiketa)).

27. Logický typ a operácie s nimi

- o Deklarácia logického typu, množina hodnôt logického typu, priorita operátorov, operácie AND, NOT, a OR, tabuľky pravdivostných hodnôt.
- o **Úloha:** Zostavte program, ktorý zistí či náhodne vygenerované číslo z intervalu <1;100> je číslo párne. Ak áno vypíše true, ak nie - false. Hľadanie nech ukončí, ak našiel 5 párných čísel.
Vstup: číslo - vygenerované. Výstup: čísla a oznamy true alebo false

Deklarácia: **názov premennej:boolean;**

Logický výraz môže nadobúdať vždy len dve hodnoty - true (pravda) alebo false (nepravda). Súčasťou výrazov môžu byť operátory.

Priorita operátorov:

NOT - unárne operátory

*, /, div, mod, and - násobiace operátory

+, -, or, xor - sčítacie operátory

=, <, >, <=>, <=, >=, in - relačné operátory

Prioritu operácií môžeme upraviť pomocou guľatých zátvoriek (najvyššia priorita).

Logické typy:

- logická negácia: not - počet operandov: 1
- logický súčin: and - počet operandov: 2
- logický súčet: or - počet operandov: 2
- logický výhradný súčet (nonekvivalencia): xor - počet operandov: 2

Hodnoty operandov		Výsledky operácií			
P	q	p and q	p or q	p xor q	not p
false	False	False	false	false	true
false	True	False	true	true	true
True	False	False	true	true	false
True	True	True	true	false	false

```
program boolean_27;
uses crt;
var a:boolean;
    b,poc:integer;
begin
  clrscr;
  randomize;
  poc:=0;
  writeln('Je cislo parne ?');
```

```
repeat
  b:=random(100)+1;
  a:=b mod 2=0;
  writeln(b,' ',a);
  if a=true then poc:=poc+1;
until poc=5;
readkey;
end.
```

Informačná spoločnosť II.

- Vysvetlite: počítačový vírus, druhy počítačových vírusov, iný nechcený softvér (spyware, adware). Ochrana PC.
- **Úloha:** Vyhľadajte na internete krátky freewerový softvér, ktorý Vám zistí, či v PC máte nejaký nechcený vírus, inštalujte ho vo svojom PC a použite ho.

Počítačový vírus: je program, ktorý je schopný vytvárať svoje kopie. Najde iný program, skopíruje seba alebo svoju časť do jeho vnútra (infikuje ho) a potom sa tento program stáva sprostredkovateľom ďalšej aktivácie vírusu. - najčastejšie sa zapisujú na koniec hostiteľského programu a jeho začiatok prepisu skokom na telo vírusu. Šíri sa pomocou iných programov.

Počítačové virusy poznáme : vírusy , červy, trojske kone, n-arna infiltrácia, škriatkovia, bomby, špióni, kombinované prostriedky

Spyware: je počítačový program, ktorý sa bez vedomia užívateľa pokúša „vyšpehovať“ citlivé dáta z počítača (napr. heslá). Tieto dáta sa potom pokúša poslať tretej strane. Následné posielanie informácie tretej strane značne spomaľuje chod PC na internete a môže aj prepisovať URL zadané v internetovom prehliadači.

Adware: je akýkoľvek softvér, ktorý automaticky zobrazuje, prehráva alebo sťahuje reklamný materiál na počítači po svojej inštalácii alebo pri používaní tohto softvéru.

Ochrana pc: Aby sme zamedzili prístupu vírusov do počítača, musíme ho pred takouto nežiadanou návštevou chrániť. Vhodnou kombináciou dvoch aspektov (aktívna antivírusová obrana a prevencia) môžeme výrazne znížiť riziko vírusovej nákazy.

28. Jednorozmerné pole

- Štruktúrovaný typ údajov, deklarácia jednorozmerného poľa, typy zložiek, vysvetlite pojem index a jeho použitie.
- **Úloha:** Zostavte program na načítanie a vypísanie jednorozmerného celočíselného poľa A. Potom vytvorte pole B tak, že $B(i+1)=A(i)$ a $B(1)=A(n)$ a tiež ho vypíšte.

Jednorozmerné pole: indexy nadobúdajú hodnoty určitého typu, ktorý musí byť ordinálny. medzi zložkami, indexami existuje jednoznačné priradenie. Každý priraduje, označuje jednu zložku. • je štruktúrovaný údajový typ zložený z pevného počtu zložiek rovnakého typu, pričom každú zložku možno explicitne (presne) pomenovať pomocou indexu.

Deklarácia: (zoznam premenných, premenné sú oddelené čiarkou) : **array[z..k] of údajový_typ**; z je počiatočná hodnota indexu, k je posledná (koncová) hodnota indexu.

index: poradové číslo jednotlivých okienok v poli.

program pole1d;

uses crt;

var a,b:array of integer;

 n,i:integer;

begin

 clrscr;

 randomize;

 writeln('zadaj dĺžku pola');

 readln(n);

 setlength(a,n);

 setlength(b,n);

 writeln;

 for i:=0 to n-1 do begin

 a[i]:=random(100);

 writeln(a[i]);

 end;

 for i:=1 to n-1 do

 b[i]:=a[i-1];

 b[0]:=a[n-1];

 writeln;

 for i:=0 to n-1 do

 writeln(b[i]);

 readkey;

end.

Bezpečnosť na internete.

- Charakterizujte spôsoby ochrany počítača, operačného systému a ďalších SW aplikácií. Objasnite význam a spôsoby ochrany používateľa počítača a jeho osobných údajov. Diskutujte o pravidlách bezpečného správania v sieti Internet.
- Úloha: Otestujte nastavenie firewall-u na vašom PC. Použite niektorý z on-line nástrojov na internete.

Ochrana pc: používanie antivírusových programov.

- Vykonávajte pravidelný update svojho antivírusového programu!
- Nikdy neotvárajte e-mailovu prílohu, ktorú ste si nevyžiadali! (vírus IloveYou, ktorý v máji 2000 spôsobil nemalé ekonomické škody (odhadujú sa asi na 8.7 miliardy USD). Programovanie týchto vírusov je pomerne jednoduché a ich efekt je rozsiahly.)
- Majte kontrolu nad svojim počítačom a nad tým, kto ho používa
- Inštalujte včas všetky záplaty(patch) na používaný softvér!
- Vždy preverujte diskety a CD média predtým, ako ho použijete a s každým novým súborom nakladajte s najvyššou opatrnosťou!
- Nenechávajte diskety v mechanike!
- Využívajte viac ako len jeden spôsob antivírusovej ochrany!
- Vytvorte si zaručene čistú bootovaciu disketu a starostlivo ju uložte na bezpečné miesto!
- Pravidelne zálohujte!

29. Reťazce a práca s nimi II.

- Údajový typ reťazec, deklarácia reťazovej premennej.
- Vedieť používať funkcie a procedúry na prevody medzi reťazcami a číslami.
- Úloha: Zostavte program, ktorý načítá celé 10 miestne číslo a určí jeho ciferný súčet a najväčšiu použitú cifru.

Reťazec znakov je postupnosť znakov, ktorá tvorí jeden celok.

Deklarácia:

(zoznam premenných) : string;

napr.: reť, text1, názov : string;

Prevody medzi reťazcami a číslami:

- procedúra Str(číselný_výraz, reťazec) – prevedie výraz ľubovoľného číselného údajového typu na reťazec. Požadovanú dĺžku reťazca a počet desatinných miest určíme podobným spôsobom ako procedúry write a writeln.
- procedúra Val(reťazec, číslo, pozícia_chyby) – z reťazca vytvorí číselnú hodnotu. Rovnaký proces prevodu vykonávajú procedúry read a readln. Medzery pred začiatkom zápisu čísla nevadia, za zápisom čísla nesmú byť žiadne ďalšie znaky. Ak reťazec neodpovedá zápisu čísla, resp. prevedená číselná hodnota nie je kompatibilná s údajovým typom skutočnej premennej číslo, prevod sa neuskutoční a do parametra pozícia_chyby sa vloží číslo, ktoré odpovedá prvej pozícii reťazca, kdesa nachádza neprípustný znak v číselnom zápise. Ak je hodnota parametra pozícia_chyby 0, prevod prebehol bezchybne.

```
program cifernysucet1;
uses crt;
var a:string;
    b,i,max:integer;
begin
  clrscr;
  writeln('Zadajte 10-miestne číslo');
  readln(a);
  b:=0;
  max:=0;
  for i:=1 to 10 do begin
    b:=b+ord(a[i])-48;
    if ord(a[i])-48>max then max:=ord(a[i])-48;
  end;
  writeln('Ciferný súčet: ',b);
  writeln('Najväčšia použitá cifra: ',max);
  readkey;
end.
```

```
program cifernysucet2;
uses crt;
var a:string;
    b,c,i,max:integer;
begin
  clrscr;
  writeln('Zadajte 10-miestne číslo');
  readln(a);
  c:=0;
  max:=0;
  for i:=1 to 10 do begin
    val(a[i],b);
    c:=b+c;
    if b>max then max:=b;
  end;
  writeln('Ciferný súčet: ',c);
  writeln('Najväčšia použitá cifra: ',max);
  readkey;
end.
```

Grafický editor

- o Stručne charakterizujte 2 základné typy grafických editorov, grafické formáty. Vysvetlite kódovanie čiernobieleho a farebného obrazu.
- o **Úloha:** Fotograf nafotil na školskom plese 100 farebných fotografií s veľkosťou 3008 x 2000 pixelov s farebnou hĺbkou TrueColor (24 bit). Bude mať dost' miesta na CD médiu pre napálenie týchto fotografií? Vypočítajte veľkosť pamäte, ktorú dané fotografie zaberú. Charakterizujte nástroje editora Skicár.
- o **Úloha:** Vo vektorovom editore vytvorte jednoduchý obrázok. Ukážte možnosti jeho editovania.

RASTROVÁ GRAFIKA Rastrové obrázky sú rozdelené na sieť myslených štvorčekov - pixelov. Rozmer každého obrázka je pre počítač počet pixelov na šírku x počet pixelov na výšku. Pre každý pixel (štvorček) je nutné okrem polohy (riadok a stĺpec) zakódovať aj farbu, resp. ďalšie parametre. Ak je obrázok monochromatický (čierna a biela farba), kódovanie je jednoduché (1 - rozsvietený (biely) bod, 0 - nerozsvietený (čierny) bod). Pri niektorých obrázkoch je však potrebné zaznamenať i odtiene, preto dve farby - čierna a biela - nestačia. Výhodné je zakódovať odtiene sivej farby pomocou ôsmich bitov tak, aby informácia o každom bode zaberala 1 B, t.j. jedno pamäťové miesto počítača. Čím bude bod svetlejší, tým väčšia hodnota sa do pamäte uloží. Preto bod čiernej farby bude uložený ako 0 a bod bielej farby ako maximálna možná hodnota - 255. napr. Paint / Skicár, LogoMotion

VEKTOROVÁ GRAFIKA Vektorové obrázky sú tvorené nie skupinou štvorčekov, ale základnými objektmi, ktorých algoritmy zostrojenia sú vopred známe, len treba zaznamenať (zakódovať) ich základné charakteristiky (napr. pre kružnicu súradnice stredu, polomer, príp. farba čiary napr. : K40Ž200150 - kružnica s polomerom 40 žltej farby a so súradnicami stredu 200 150). napr. Zoner Callisto, Inkscape

Grafické formáty

- **BMP (Windows Bitmap)**
Uložený ako rastrový (bitmapový) obrázok. Nevýhodou je, že na disku zaberá príliš veľa miesta. Môže obsahovať až 16 miliónov farieb (24 bitová farebná hĺbka).
- **GIF (Graphics Interchange Format)**
Uložený ako rastrový obrázok skomprimovaný bezstratovou kompresiou. Skupiny rovnakých bodov sa uložia ako krátka informácia. Obrázky môžu obsahovať najviac 256 farieb (8 bitová farebná hĺbka), čím sa znižuje pôvodná kvalita obrázku. Je vhodný na prenášanie obrázkov.
- **JPG (JPEG = Joint Photographic Experts Group)**
Uložený ako rastrový obrázok. Môže obsahovať až 16 miliónov farieb (24 bitová farebná hĺbka). Do súboru sa uložia obrázky skomprimované stratovou kompresiou. Je vhodný na obrázky, ktoré obsahujú veľa jemných farebných prechodov (fotografie), ktoré sa pri komprimácii zjednodušujú. Pri veľmi ostrých hranách po skomprimovaní už dokážeme vnímať chyby. Na disku zaberá málo miesta, preto je vhodný na prenášanie obrázkov.

Kódovanie farieb

Aby bol obrázok farebný, musíme každému pixelu obrázka priradiť určitú farbu. Ľudské oko dokáže vnímať iba určitý rozsah vlnových dĺžok svetla. Svetlo s najväčšou vlnovou dĺžkou, ktoré ľudské oko vníma, je červenej farby (780 nm) a svetlo s najnižšou vlnovou dĺžkou je svetlo fialovej farby (380 nm). V tomto spektre farieb je ľudské oko schopné bez problémov rozlíšiť vyše 1,5 milióna rôznych farieb. Prvé riešenie problému kódovania farieb, ktoré sa ponúka, je zobrať celý rozsah vlnových dĺžok a rozdeliť ho na niekoľko častí a do pamäte počítača uložiť poradové číslo farby, ktorej prislúcha určitá vlnová dĺžka, ďalej jej sýtosť a jas. Kódovanie farieb takýmto spôsobom využívajú farebné modely HSB, HLS. Oba modely uchovávajú informáciu o odtieni (Hue) a sýtosti (Saturation). Tieto modely sa odlišujú iba v tom, že prvý model používa jas (Brightness) a druhý používa svetlosť (Lightness). Farebný model RGB je najčastejšie využívané kódovanie farby bodu obrázka. Empiricky sa zistilo, že takmer všetky farby sa dajú vytvoriť zmiešaním ľubovoľných troch nezávislých farieb (t.j. že pomocou zmiešania dvoch nedostaneme tretiu). Najvýhodnejšie pre výrobu svetelných lúčov bolo použitie farieb červená (Red), zelená (Green), modrá (Blue). Aby sme vedeli vytvoriť 1,5 milióna farieb stačí, ak každú z týchto farieb rozdelíme na 115 oddtiev, ktorých zmiešaním v rôznych pomeroch vzniknú všetky farby. Kvôli uchovaniu v pamäti počítača, je však výhodnejšie použiť až 256 odtiev každej farby (8 bitov), čo nám umožní vytvoriť až 16 777 216 rôznych farieb. Pri takomto kódovaní je každá farba zakódovaná 24 bitmi, čo sú tri pamäťové miesta počítača.

30. Textové súbory a práca s nimi.

- o Vytvorenie súboru na disku, zápis údajov do súboru, výpis údajov zo súboru, príkazy pre prácu so súbormi v Pascale.
- o **Úloha:** Zostavte program, ktorý:
 - a) uloží (zapíše) do súboru tuzapisuj.txt vaše krstné meno.
 - b) načíta z vopred vytvoreného súboru tonacitaj.txt vaše priezvisko

Pre prácu so súborom potrebujete definovať tzv. rukoväť súboru. Rukoväť je premená typu súbor a definujeme ju v časti var.
var f:text; { textový súbor }

Túto rukoväť najprv spojíme so súborom ktorého meno poznáme. To aby počítač vedel o aký súbor nám ide. Spojíme ho príkazom Assign. Ďalej už budeme pracovať so súborom len pomocou premennej f a názov súboru môžeme zabudnúť.

```
Assign(f, 'C:\subor.txt');
```

Pre ďalšiu prácu musíme určiť či budeme súbor čítať, alebo doň zapisovať. Na každú činnosť použijeme iný príkaz otvorenia súboru. Ak budeme čítať použijeme príkaz ReSet a v zátvorke uvedieme rukoväť súboru. Ak budeme zapisovať použijeme príkaz ReWrite a rukoväť.

```
ReSet(f); { budeme čítať subor }
```

```
ReWrite(f); { budeme zapisovať do súboru }
```

Ďalej zo súborom pracujeme pomocou príkazov Write alebo Read (prípadne WriteLn a ReadLn). Ak chceme zapisovať použijeme Write. Za normálnych okolností sa používa Write na výpis textu na obrazovku. Ak však ako prvý parameter uvedieme premenú rukoväť, tak sa všetko bude zapisovať do súboru. Obdobne funguje Read. Všetko čo by sme museli zadávať z klávesnice sa bude čítať zo súboru. Pri Read ako prvý parameter tiež uvedieme rukoväť.

Ukazuje rozdiel medzi zápisom na obrazovku a zápisom do súboru.

```
Write('Text na monitor');
```

```
Write(f, 'Text do suboru'); { f je rukoväť súboru }
```

```
Read(i); { prečítaj premenú i z klávesnice }
```

```
Read(f,i); { prečítaj zo súboru }
```

POZOR: Write použijeme len na súbor ktorý sme predtým otvorili v režime ReWrite. Read zase na súbor ktorý sme predtým otvorili v režime ReSet

V prípade že budeme súbor čítať, máme problém rozoznať kedy je koniec súboru, a teda kedy už nemá zmysel súbor ďalej čítať. K tomu slúži funkcia Eof (End-of-file), ktorú použijeme v podmienke neakého cyklu. Tým cyklom je najčastejšie while, ale môže to byť aj repeat. Funkcia Eof je True ak sme dosiahli koniec súboru, ináč je vždy False.

```
ReSet(f);
```

```
while ( not( eof(f) )) do begin
```

```
    ReadLn(f,s);
```

```
    WriteLn(s);
```

```
end;
```

Poslednou časťou je ukončiť prácu so súborom, zavrieť súbor. To vykonáme príkazom Close. Parametrom je rukoväť.

```
Close(f);
```

Práca so súborom sa delí na 4 časti.

ASSIGN - Spojenie názvu súboru s rukoväťou

RESET - Režim čítanie

REWRITE - Režim zápis

WRITE - Zápis do súboru

READ - Čítanie zo súboru

EOF - Zisťujeme koniec súboru pri Read

CLOSE - Uzavretie spojenia

```
program textsubor;
```

```
uses crt;
```

```
var s:text;
```

```
    a,b:string;
```

```
begin
```

```
clrscr;
```

```
writeln('Napiste vase krstne meno');
```

```
read(a);
```

```
assign(s, 'D:\tuzapisuj.txt');
```

```
rewrite(s);
```

```
write(s,a);
```

```
close(s);
```

```
assign(s, 'D:\tonacitaj.txt');
```

```
reset(s);
```

```
read(s,b);
```

```
writeln(b);
```

```
close(s);
```

```
readkey;
```

```
end.
```

Internet.

- Vyhľadávanie a získavanie informácií, webové prehliadače, webové katológy, vyhľadávacie stroje, rozšírené vyhľadávanie.
- **Úloha:** Vyhľadajte konkrétnu informáciu:
Vyhľadajte informácie o možnostiach štúdia odboru Informatika a kritéria na prijatie na FEI TU v Košiciach.

Vyhľadávanie a získavanie informácií-

prehliadače www stránok - Na prezeranie WWW stránok potrebujeme mať v našom počítači nainštalovaný prehliadač WWW stránok (WWW browser).

Prehliadače slúžia na:

1. komunikáciu so zdrojom WWW stránky - WWW serverom (starajú sa napríklad o presunutie WWW stránky z Internetu na používateľov počítač),
2. zobrazenie WWW stránky
3. prácu s WWW stránkou (uloženie, vytlačenie, založenie do záložkového zoznamu, vyhľadanie textu na nej). Najznámejšie prehliadače sú Netscape Navigator (NN) a Microsoft Internet Explorer (MSIE).

Internetové katalógy sú webové servery, ktoré zhromažďujú množstvo odkazov na veľké množstvo webových stránok a zároveň mnohé z nich slúžia ako vyhľadávače, čiže v pridaných odkazoch sa dá vyhľadávať a tak nájsť hľadanú informáciu. Medzi najznámejšie patria Atlas.sk, Zoznam.sk a Centrum.sk. Zaradením stránky do katalógu získate veľmi kvalitný odkaz na vašu stránku (z veľmi známej stránky, na ktorej vyhľadáva množstvo ľudí).

Vyhľadávacie stroje (Search Engines), sú softwarové nástroje, ktoré pracujú na inom princípe. Typickými predstaviteľmi vyhľadávacích strojov sú AltaVista, UltraInfoseek a Lycos. Vyhľadávacie stroje obsahujú software -- vyhľadávací robot -- ktorý aktívne prehľadáva webovské stránky na Internete a keď nájde novú alebo zmenenú webovskú stránku (za zmenenú sa považuje zmena v titulku webovej stránky), prečíta si ju, zaindexuje si ju a uloží na svojom serveri. Základný rozdiel medzi vyhľadávacím strojom a vyhľadávačom typu knižnica je, že vyhľadávací stroj sám aktívne vyhľadáva webovské stránky.