



**mpc**  
METODICKO-PEDAGOGICKÉ CENTRUM



**Moderné vzdelávanie pre vedomostnú spoločnosť / Projekt je spolufinancovaný  
zo zdrojov EÚ**

PaedDr. Erik Guman

## **Informatika vo vyučovacom procese školy**

Osvedčená pedagogická skúsenosť edukačnej praxe

Osvedčená skúsenosť odbornej praxe

Prešov, 2013

**Vydavateľ:** Metodicko-pedagogické centrum, Ševčenkova 11,  
850 01 Bratislava

**Autor OPS/OSO:** PaedDr. Erik Guman

**Kontakt na autora:** Gymnázium, Konštantínova 2, Prešov,  
guman@gk2-po.sk.

**Názov OPS/OSO:** Informatika vo vyučovacom procese školy

**Rok vytvorenia OPS/OSO:** 2013

**Odborné stanovisko vypracoval:** Mgr. Lukáš Gulden

Za obsah a pôvodnosť rukopisu zodpovedá autor. Text neprešiel jazykovou úpravou.

Táto osvedčená pedagogická skúsenosť edukačnej praxe/osvedčená skúsenosť odbornej praxe bola vytvorená z prostriedkov projektu Profesionálny a kariérový rast pedagogických zamestnancov. Projekt je financovaný zo zdrojov Európskej únie.

## **Kľúčové slová**

Delphi, Pascal, programovanie, štýly učenia, motivácia, osobnosť žiaka, informatika, počítač

## **Anotácia**

Veľa učiteľov a žiakov sa v čase mohutného rozvoja informatiky a IKT zaujíma napr. o tvorbu webových stránok, programovanie a iné s informatikou súvisiace činnosti. Táto práca má slúžiť ako návod, či motivácia pre učiteľov, ktorí vyučujú programovanie, prípadne začínajú programovať v prostredí Delphi, alebo pre žiakov, ako pomôcka pri ich samoštúdiu. Človek, ktorý vie programovať, ovláda univerzálny jazyk, ktorý môže využiť nielen doma, v zahraničí, na vysokej škole alebo pri realizácii vlastných projektov.

# OBSAH

Úvod

1 POČÍTAČ V EDUKAČNOM PROCESE .....	6
1.1 Zefektívňovanie vzdelávacieho procesu novými metódami .....	6
1.2 Sústava metód .....	7
1.3 Motivácia vo vyučovacom procese a jej vplyv na osobnosť žiaka .....	8
2 ŠTÝL UČENIA.....	12
2.1 Ako ovplyvňovať štýl učenia .....	13
2.2 Osobnosti ovplyvňujúce štýl učenia .....	13
3 ZÁKLADY PROGRAMOVANIA V DELPHI .....	15
3.1 Prostredie Delphi .....	15
3.2 Grafická plocha – Image .....	16
3.3 Grafické príkazy .....	17
3.4 Náš prvý program .....	18
3.5 Kreslenie geometrických útvarov .....	20
3.6 Deklarácia celočíselných premenných a náhodnosť .....	23
3.7 Cyklus For .....	27
3.8 Podmienený príkaz .....	31
3.9 Cyklus s podmienkou .....	34
Záver .....	37

## ÚVOD

Programátorské prostredie Delphi používa jeden z prvých programovacích jazykov Objektový Pascal. Jazyk Pascal je na celom svete považovaný za jazyk, ktorý učí programovať a zároveň rozvíja a podporuje algoritmické myslenie. Pascal pre svoju jednoduchosť a jednoznačnosť je odporúčaný na ZŠ a SŠ. V súčasnosti vzniká množstvo nových programovacích jazykov, ktoré sa inšpirujú Pascalom. Preto programátorské štýly, ktoré študenti získajú v tomto jazyku, využijú na mnohých vysokých školách. Kto ovláda pascal, umožni mu to veľmi prirodzený prechod na iný programovací jazyk.

Rôzne programátorské prostredia umožňujú vyvíjať okrem aplikácii pre grafické rozhranie (napr. Windows), ale aj aplikácie pre web, ktoré sú založené na vizuálnom, kinestetickom princípe. Vizuálnosť, auditívnosť, čítanie - písanie, kinestetika patria medzi teórie učebných štýlov. Študenti, ktorí preferujú logicko - matematický štýl učenia, sú výnimoční pri zostavovaní a riešení rôznych programátorských úloh, ktoré vyžaduje abstraktné myslenie, logiku, práce s číslami, s logickými hádankami a matematickými výrazmi, experimentovanie a pod.

Jednotlivé učebné štýly je možné podporiť cieľavedome. Podpora štýlov učenia sa pomáha zvýšiť efektívnosť výučby informatiky (programovania).

Štýly predstavujú stabilné spôsoby pristupovania k riešeniu úlohy a sú individuálnymi črtami jednotlivca, kým stratégie poukazujú na spôsob narábania s konkrétnymi úlohami. Štýly sú teda zamerané na osobnosť a stratégie na úlohu.

Existuje vzťah medzi preferovaným štýlom učenia sa a jeho vplyv na náročnosť učenia sa programovať. Preto by sme mali prispôbiť formy, metódy výučby tak, aby z toho profitovali žiaci s rôznym štýlom učenia sa.

Prácu sme rozdelili do troch kapitol. Prvá kapitola sa zameriava na zefektívňovanie vzdelávacieho procesu novými metódami, osobnosti žiaka a vplyvom motivácie na jeho rozvoj. Druhá kapitola popisuje štýl učenia a možnosti ich ovplyvňovania. Tretia kapitola sa venuje programátorskému prostrediu Delphi, ktoré používa jeden z prvých programovacích jazykov Objektový Pascal. Témy z Delphi sú teoreticky spracované a ku každej téme budú uvedené vyriešené vzorové príklady.

# 1 POČÍTAČE V EDUKAČNOM PROCESE

## 1.1 Zefektívňovanie vzdelávacieho procesu novými metódami

Súčasná didaktika vyvinula a neustále vyvíja nové metódy, ktoré sa snažia žiaka aktivizovať a rozvíjať jeho tvorivosť. Podľa J. Bajtoša a J. Pavelku (1999) *aktivovať*, znamená zvyšovať určitými prostriedkami pracovnú pohotovosť energetických zdrojov a udržiavať túto pohotovosť i samotnú výkonnosť žiaka na zvýšenej aktivizačnej úrovni. S pojmom aktivizácia súvisí aj pojem tvorivosť (vynaliezavosť), ktorá spočíva v tom, že žiak nevykonáva danú činnosť mechanicky, stereotypne, ale snaží sa riešiť problémy novým, netradičným spôsobom. Učiteľ môže žiakom svojimi metódami vyučovania pomôcť pri rozvoji motivácie. Uvedieme najčastejšie používané aktivizujúce metódy a metódy rozvíjajúce tvorivosť žiakov:

### *Brainstormingové metódy*

Brainstorming môžeme charakterizovať ako metódu riešenia problémov skupinou žiakov prostredníctvom nových myšlienok a najoriginálnejších nápadov vedených formou voľnej diskusie na určitú tému. Základnou úlohou je vytvorenie tvorivej klímy. Táto metóda je založená na týchto princípoch:

- *princíp úplnej voľnosti nápadov* - žiak môže uviesť čo najviac nápadov, pričom sa všetky nápady akceptujú,
- *princíp produkcie kvantity a nie kvality* - najlepšie je vyprodukovať čo najviac nápadov, ktoré môžu byť neobvyklé, bláznivé, fantastické a pod,
- *princíp straty autorského práva na nápad* - nikto si nemôže robiť nárok na autorstvo svojho nápadu, pretože jeden nápad je podmienený druhým,
- *princíp zákazu kritiky* - aby sa nenarušila tvorivá atmosféra, je zakázané analyzovať návrhy a akákoľvek kritika.

Táto metóda sa využíva najčastejšie na začiatku výkladu učiva alebo pri začiatku nového tematického celku. Učiteľ môže prostredníctvom tejto metódy zistiť, čo už žiaci o danej téme vedia.

### *Systém dokonalého osvojenia učiva (Mastery learning)*

Podstatou tejto metódy je, že každý žiak si má osvojiť učivo na takzvanej dokonalej úrovni, ktorá predstavuje 80 - 90% učiva. Realizácia prebieha tak, že učivo sa rozdelí na menšie moduly (celky). Jeden modul obsahuje učivo, ktoré sa vyučuje asi dva týždne. Po naplnení požiadavky žiaci píšú didaktické testy, v ktorých ak neuspeli, nepokračujú k ďalšiemu modulu. Výhodou je, že žiaci nie sú hodnotení vzhľadom k iným žiakom, ale vzhľadom na zvládnuté učivo. Ak by takáto forma vyučovania splnila určité podmienky, mohla by sa použiť vo výučbe predmetu Práca s počítačom, kde je dôležité, aby žiak dokonale zvládol základy učiva.

### *Situačná metóda*

Jej podstatou je motivácia žiakov opisom určitej situácie, ktorá môže byť skutočná alebo neskutočná. Úloha žiakov spočíva v tom, že majú hľadať správne východisko z danej situácie alebo riešiť modelový prípad. Opis musí byť veľmi priehľadný a stručný, aby žiaci sami navrhovali riešenie. Táto metóda má využitie pri tvorbe počítačových programov.

### *Inscenačná metóda*

Podstata metódy spočíva v tom, že žiaci budú hrať svoju úlohu zo života, to znamená, že zinscenujú určitú situáciu. Po skončení a následnej diskusii sa pokúsia nájsť isté

východisko zo vzniknutej situácie. Väčšinou sa jedná o simuláciu takých javov, ktoré sa môžu prihodiť v skutočnosti.

#### *Kooperatívne vyučovanie*

Podstatou tejto metódy je práca žiakov v malých a rovnoprávných skupinách, kde sa riešia, vykonávajú rôzne činnosti. Žiaci sú hodnotení podľa výkonu skupiny, pričom si môžu osvojiť napríklad deľbu práce, spoluprácu, pomoc a pod. Cieľ takéhoto vyučovania smeruje k spolupráci medzi spolužiakmi a nevedie k súťaživosti, či individualizmu.

#### *Projektové metódy*

Takýto druh vyučovania sa vyznačuje najvyšším stupňom samostatnosti a poznávacej činnosti žiakov. Vychádza zo skúseností bežného života a snaží sa prebúdzat' u žiakov prirodzený záujem o poznávanie, pretože je späté s aktuálnymi požiadavkami spoločnosti. Jeho realizácia spočíva v tom, že žiakom sa zadá určitý problém, ktorý majú samostatne riešiť, pričom sa im zdôrazňuje to, aby sami získali čo najväčší počet informácií o zadanej úlohe. Napríklad použitie Internetu, ako zdroja rozsiahleho množstva informácií.

Projektová výučba je jednou z najdokonalejších podôb konštruktívneho spôsobu výučby. Dovoľuje nielen aktívne zapojiť žiaka do práce, ale umožňuje aj nacvičiť skupinovú spoluprácu a zaradiť medzipredmetové vzťahy. Projekty, ktoré žiaci riešia, môžu byť dlhodobé, individuálne alebo skupinové. Do projektov sa môžu zapojiť nielen žiaci, ale aj odborníci z príslušných odborov. Tému, ktorú ma spracovať každý žiak, by sa mal dozvedieť celý kolektív triedy, škola, ale aj verejnosť. Preto dôležitú úlohu tu má opäť Internet, ktorý umožňuje prostredníctvom služby www prezentovať zadanie, ale aj výsledky projektu.

## **1.2 Sústava metód**

Úlohou učiteľa vo vyučovaní je sprostredkovať učivo žiakom tak, aby si ho mohli osvojiť, pričom sa plnia úlohy a ciele výchovno-vzdelávacieho procesu pri použití rôznych metód a didaktických prostriedkov. Metódou vyučovania rozumieme cieľavedomý, koordinovaný a zámerný postup, ktorým sa realizuje výchovno-vzdelávací proces, zameraný na dosiahnutie vytýčených cieľov. Súčasná didaktika popisuje množstvo vyučovacích metód, ale my sme sa pozastavili predovšetkým pri tých najklasickejších.

Pedagogická literatúra uvádza niekoľko základných druhov klasifikácie vyučovacích metód podľa rôznych hľadísk:

#### *a) podľa zdroja informácií:*

- slovné metódy,
- názorné metódy,
- praktické metódy.

Počítač môžeme využiť pri sprostredkovaní všetkých týchto troch metód. Pri slovných metódach môže byť zdrojom informácií písaná alebo tlačená reč, ktorá sa dá sprostredkovať cez Internet. Čo sa týka názorných metód, zdrojom poznania je živé nazeranie napríklad prostredníctvom multimédií žiaci pozorujú nejaký predmet alebo jav. Pri praktických metódach je podstatou aktívna činnosť žiakov realizovaná prostredníctvom počítača. Patria sem rôzne projektové, grafické, konštrukčné, technologické a písomné práce.

#### *b) podľa funkcie vo vyučovacom procese poznáme:*

- motivačné metódy,
- expozičné metódy,
- fixačné metódy,
- diagnostické metódy.

Aj tieto metódy môže učiteľ realizovať pomocou počítača. Počítač je výborný motivačný prvok, môže byť aj médiom pri sprostredkovaní nových poznatkov a ich prehĺbovaní, upevňovaní. Pomocou počítača je možné diagnostikovať vedomosti žiakov.

c) z hľadiska logického postupu poznáme:

- induktívnu metódu,
- deduktívnu metódu,
- analyticko-syntetickú metódu,
- porovnávaciu metódu.

d) z hľadiska aktivity a samostatnosti žiakov:

- metódy odovzdávania poznatkov,
- metódy pozorovania žiakov,
- metódy samostatnej práce žiakov,
- metódy bádateľské a výskumné.

Dá sa konštatovať, že pomocou počítača môže učiteľ uplatniť takmer všetky základné druhy vyučovacích metód. Dôležitá úloha učiteľa spočíva v jeho správnom rozhodovaní pri výbere vyučovacej metódy, so zreteľom na skúsenosti pri používaní metód v istom ročníku a na obsah učiva.

### **1.3 Motivácia vo vyučovacom procese a jej vplyv na osobnosť žiaka**

Rozsiahly pojem, ako je pojem osobnosť, nemožno plne definovať stručným výrokom. Názory na definíciu osobnosti, možnosti a spôsoby jej hodnotenia sa doteraz líšia. Väčšinou sa pritom berie do úvahy povaha, inteligencia, emocionálna a motivačná stránka človeka.

Pojem osobnosť sa používa v bežnej reči, vo vede, i filozofii a označuje:

- a) jednotlivého človeka ako osobu, ako subjekt vzťahov k svetu a vedomej činnosti,
- b) relatívne stabilnú sústavu spoločenských vlastností, ktoré charakterizujú jedinca ako člena spoločnosti a skupiny.

Najčastejšie sa osobnosť chápe v druhom význame, v užšom zmysle. V pedagogike hovoríme o osobnosti žiaka, v sociológii o osobnosti umelca, v kriminológii o osobnosti páchatel'a. Osobnosť tvorí to, čo robí z človeka žiaka, umelca, páchatel'a.

M. Homola (1971) definuje osobnosť ako „celok psychickej organizácie, súbor a usporiadanosť psychických vlastností prejavujúcich sa ako celok“ (Homola, 1971, str. 10).

Pardel a Boroš (1975) uvádzajú definíciu osobnosti podľa Tardyho (1964), ktorý definuje osobnosť ako „individuálnu jednotu človeka; je to jednota jeho duševných vlastností a dejov založená na jednote tela a utváraná a prejavujúca sa v jeho spoločenských vzťahoch“ (Pardel, 1975, str. 50). Mašek (1969) osobnosť chápe ako integrovaný súbor vlastností biologických, psychických a spoločenských. Osobnosť je súbor duševných vlastností jednotlivca, trvalý celok jeho duševného života.



Preto v tomto zmysle osobnosť vyjadruje:

- a) jednotu všetkých čiastkových psychických funkcií,
- b) jednotu duševného diania a činností,
- c) jednotu duševného života a podmienok existencie.

(Nákonečný, M., 1972)

A. N. Leontiev hovorí, že reálny základ osobnosti človeka nie je v jeho genetických programoch, nie je v hĺbke jeho prirodzených dispozícií, nie je v získaných návykoch, vedomostiach ani zručnostiach, teda ani v profesionálnych, ale v systéme činností, ktoré sa týmito vedomosťami a zručnosťami realizujú.

Osobnosť je otvorený súbor sociálnych kvalít ľudského individua čiže osoby, jeho názorov, schopností, potrieb, záujmov, morálneho presvedčenia a pod.

„Osobnosť je chápaná ako integrálna psychická individualita, ako neopakovateľný, dynamický, svojrázny a pritom zákonite sa štrukturalizujúci komplex. Táto entita je utváraná súčinnosťou vrodenej bázy a individuálnou životnou praxou v konkrétnych spoločenských, kultúrnych a situačných podmienkach“ (Fandelová, 1999, str. 30). Existuje v reálnom sociálnom prostredí a životných situačných podmienkach. Aktualizuje sa do systému vnútornej a vonkajšej činnosti toho ktorého subjektu v dôsledku samotnej aktivity individua.

Pod osobnosťou sa rozumie konkrétny človek danej spoločnosti. Aby sa človek stal osobnosťou počas svojho vývinu, musí dosiahnuť určitú psychickú úroveň. Osobnosť je charakterizovaná určitými názormi, postojmi a vzťahmi, vlastnými morálnymi potrebami, úrovňou aktivity a inými dôležitými vlastnosťami, ktoré nevznikajú naraz. Vznikanie a formovanie, ale aj zdokonaľovanie (utváranie) osobnosti prebieha počas života človeka, ale tiež ako dôsledok výchovy.

Základnou požiadavkou súčasnej školy je riadiť výchovno - vzdelávací proces tak, aby žiak nebol len jeho objektom, ale aj jeho subjektom. Rovnako je potrebné, aby tento proces mal na osobnosť žiaka formatívny účinok, aby ho vychovával k samostatnosti, tvorivosti a pestoval uňho úsilie vzdelávať a pretvárať sa.

Preto súčasná psychologická teória a pedagogická prax sú postavené pred závažnú úlohu – ako riešiť otázku psychického rozvoja osobnosti žiakov. Práca školy sa presúva do novej polohy, ktorej stredobodom pozornosti už nemôžu byť len kognitívne procesy, vedomosti, osvojovanie poznatkov, ale rozvoj celej osobnosti žiaka. Tak sa do popredia dostávajú záujmy, otázky dynamickej sféry žiakovej osobnosti, otázky regulácie a motivácie.

Na pojem motivácia nie sú v psychológii jednotné názory. Napríklad M. Homola (1971) chápe motív ako „označenie vnútorných podmienok, ktoré vzbudzujú a udržujú aktivitu človeka a ktoré určujú, že sa správa určitým spôsobom“ (Homola, 1971, s. 5).

Podľa M. Zelinu (1996) *motiváciu* chápeme v najširšom slova zmysle ako súhrn činiteľov, ktoré vyvolávajú, usmerňujú a zacielujú ľudskú aktivitu. Mnohé výskumy

ukázali, že motivácia je základnou otázkou pre rozvoj tvorivosti, nadania a schopnosti každého človeka.

*Motivácia môže byť:*

a) **vnútorná** – chápe sa ako stav, ktorý núti jednotlivca niečo robiť alebo niečomu sa učiť pre vlastné uspokojenie, pre vlastný zážitok,

b) **vonkajšia** – chápe sa ako stav, keď sa jednotlivec neučí z vlastného záujmu, ale vplyvom vonkajších motivačných činiteľov. (Ďurič L. a kol., 1991)

„Motiváciu nemožno chápať ako izolovaný vonkajší prvok, ktorým sa oživuje vyučovanie. Často práve ona určuje štýl vyučovania a celkovú atmosféru, v ktorej výchovno-vzdelávací proces prebieha. Motivácia činnosti žiakov vo vyučovaní môže byť rôzna“ (Ondrušová, 1995, s. 77). Motivačne pôsobí napr. sám obsah učiva, aktuálnosť látky, problém, ktorý má žiak vyriešiť, osobnosť učiteľa, úspech, uznanie, počítač a pod.

*Motivácia v škole pramení z troch vonkajších zdrojov:*

a) poznávacích potrieb (poznávanie a získavanie nových poznatkov),

b) sociálnych potrieb (sociálnych vzťahov),

c) výkonových potrieb (obťažnosť úloh, ktoré sa zadávajú žiakom). (Hrabal V., 1989)

Pri klasickom spôsobe vyučovania učitelia zdôrazňujú a preferujú primárnu motiváciu t.j. stav, v ktorom je študent aktívny pre vlastnú aktivitu. Pri používaní výučbových prostriedkov akými sú počítače sa využíva taktiež sekundárna motivácia, ktorá je s učebnou aktivitou spojená – napr. žiak sa učí pre radosť z práce s počítačom. Je pedagogicky efektívnejšie využívať motiváciu *kladnú*, t.j. radosť zo zvládnutia znalostí, založenou na poznávaní (Fialová, I., 1999).

Vo vyučovacom procese motivačné činitele práce s počítačom musia pôsobiť v troch základných smeroch:

1. Aktivizovať žiaka a tým utvárať lepšie, priaznivejšie vnútorné podmienky pre výchovu a učenie.
2. Ovplyvňovať študijné zameranie žiaka a tak prispieť k osvojeniu takých foriem správania, ktoré od žiaka požaduje škola a spoločnosť.
3. Regulovať a usmerňovať výber podnetov a činností, podieľajúcich sa na rozvoji a kvalite žiakových výsledkov práce.

Počítač ako učebná pomôcka vybavený programovým softvérom, môže pôsobiť na žiaka motivujúco respektíve ako pomocník v učení, pretože:

a) akceptuje individuálne tempo žiaka,

b) reaguje na žiakové riešenia,

c) umožňuje vyberať vhodné príklady, úlohy a problémy,

d) výber úloh možno prispôbiť schopnostiam žiaka,

e) môže vyhodnotiť prácu každého žiaka podľa daných kritérií,

f) je schopný opakovať predchádzajúce učivo,

g) umožňuje spätnú väzbu,

h) umožňuje optimálny postup riešenia úlohy,

i) ak žiak potrebuje pomoc, počítač odporúča podobné schémy a vzory postupov a ukazuje, ako ich aplikovať.

V závislosti na jednotlivých stupňoch sa mení aj vzťah žiakov k počítačom. Na začiatku je hlavným motivačným prvkom túžba po praktickej práci na počítači a radosť z jeho ovládania. Ako náhle žiaci zvládnu základy práce na počítači, motivuje ich na hodine „hra na počítačoch“. Vtedy dochádza k veľkej zmene vo využívaní počítača na hodine a k budovaniu si u žiaka vzťah k počítaču. Preto pri správne zvolenej motivácii by mali spoznávať počítač ako nástroj, pomocou ktorého sa mnohé javy dajú naučiť ľahšie, príjemnejšie a jednoduchšie ako pomocou klasickej učebnice.

Učitelia by mali byť vedení k novému ponímaniu vyučovania, ktoré namiesto prioritného sledovania pamäťového učenia žiakov a s tým spojeného pretvárania metód a foriem vyučovania sa bude orientovať na koncepciu rozvoja osobnosti žiaka a k tomu sa prispôsobí prístup k vyučovaniu.

## 2 ŠTÝL UČENIA

V šesťdesiatych a sedemdesiatych rokoch došlo k výrazným zmenám v názoroch na ľudské učenie. Teória programového učenia presadzovala úplné riadenie žiakovo učenia. Daný výraz môže byť chápaný ako rozdielna činnosť rodičov, učiteľov, spolužiakov zúčastňujúcich sa na vyučovacom procese, ktorý umožňuje popísať a analyzovať, čo sa vlastne odohráva, keď niekto kontroluje alebo usmerňuje žiakovo učenie. Týmto problémom sa dôkladne zaoberali D. Tollingerová a V. Kulič. Venovali sa vonkajšiemu riadeniu, teda situácii, keď niekto okrem žiaka samotného organizuje a riadi jeho učenie, napríklad počítač, špeciálne učebnice, učiteľ a pod.

V posledných rokoch stúpa záujem o individualitu človeka, výskumní pracovníci a učitelia majú záujem o *individuálne postupy*, ktoré žiak používa pri svojom učení. Preto môžeme vecne aj terminologicky rozlíšiť rôzne pohľady na individuálne postupy pri učení, kde je dôležité charakterizovať niektoré pojmy, ktoré uvádza J. Mareš (1998):

- *stratégia učenia (learning strategies)* sú postupy väčšieho rozsahu, ktorými žiak realizuje určitý plán pri riešení úlohy, chce niečo dosiahnuť, niečo sa vyvarovať,
- *kognitívny (poznávací) štýl (cognitive style)* je charakteristický spôsob, ktorými človek vníma, myslí, zapamätáva, rieši problémy, rozhoduje sa,
- *štýly učenia (learning style)* sú postupy pri učení, ktoré majú charakter metastratégie učenia sa. Jedinec ich používa väčšinou v určitej pedagogickej situácii, v určitom období života.

Podľa toho, ktorý druh inteligencie človeka prevažuje pri učení, môžeme štýl učenia rozdeliť:

- *ingvistická (jazyková)*: žiak veľmi rád pracuje so slovom., rád číta, píše, vedie diskusiu, rád sa rozpráva,
- *logicko-matematická*: preferuje prácu s číslami, s logickými a matematickými výrazmi, sú výnimoční pri riešení logický programátorských úloh,
- *priestorová*: žiaci majú vyvinutú predstavivosť a fantáziu, radi kreslia, snívajú a radi sledujú televíziu,
- *telesne-kinestetická (pohybová)*: študent je pohybovo nadaný a rýchlo si osvojuje psychomotorické zručnosti. Prejavuje pri športe, tanci a pod.,
- *muzikálna (hudobná)*: *dokážu si veľmi dobre zapamätať rytmus, tempo melódie, tóny. Rád si spieva, počúva hudbu a veľmi dobre rozoznáva hudobné tóny,*
- *interpersonálna*: pracuje v kolektíve, ktorý rád riadi a organizuje,
- *intrapersonálna*: je presným opakom študenta s interpersonálnym učebným štýlom, rád pracuje sám a učí sa osamote. Má vlastné učebné tempo.

(Gardner,1983)

Štýl učenia vzniká na vrodennom základe a rozvíja sa pôsobením vnútorných a vonkajších vplyvov. Vedie k výsledkom určitého typu a zabraňuje dosiahnutiu výsledkov iných. Každý človek má vrodené postupy, ktorými poznáva okolitý svet. Tieto postupy sa označujú ako kognitívne štýly napríklad reflexibilita - impulzivita, riskovanie - opatrnosť, závislosť - nezávislosť a pod.

Psychológia osobnosti, ktorá sa zaoberá zvláštnosťami ľudskej osobnosti, poukazuje na súvislosť medzi osobnosťou človeka a jeho učením. Osobnosť človeka a jeho učenie

spolu úzko súvisia. Niektoré zložky osobnosti majú skúsenostný základ, vznikajú a dotvárajú sa v procese socializácie, výrazne ich ovplyvňuje práve učenie. Iné zložky majú naopak vrodený, biologický základ a učenie ich nemôže zmeniť.

## **2.1 Ako ovplyvňovať štýl učenia**

V súčasnej škole sa venuje veľa pozornosti obsahovej stránke vyučovania a učenia. Spätnou väzbou sa zisťuje, čo sa žiaci z daného učiva majú naučiť a či sa to správne naučili. V poslednej dobe sa tento pohľad začína meniť. Žiaci sa snažia okolitý svet, ktorý ich obklopuje, znovu utvárať respektíve konštruovať. To predpokladá vlastný spôsob učenia, učiť sa a jednať z vnútorných motívov. Žiak v takomto chápaní vystupuje ako samostatne jednajúci a konštruujúci jedinec, preto škola musí rozvíjať a trénovať rôzne stratégie učenia, rozumovo a citovo ovplyvňovať štýly učenia.

Štýly učenia sa dajú diagnostikovať a meniť, ale nie je to jednoduché. Dá sa to realizovať po celý ľudský život. Špecifickým prípadom učenia je učenie, ako sa učiť tzv. metaučenie. Učenie podľa Kuliča (1992) je proces, počas ktorého človek mení svoj súbor poznatkov, mení svoje formy chovania a spôsob činnosti, vlastnosti svojej osobnosti a obraz o sebe samom. Mení svoj vzťah k ľuďom, k spoločnosti s cieľom vyššej účinnosti a dosiahnutie rozvoja. Učenie, ako sa učiť je dynamický proces, v ktorom si jedinec začína uvedomovať učenie, ktoré sa stáva dostupnejšie a postupne sa vyvíja. Prehľbuje sa porozumenie, jedinec tak jedná efektívnejšie, flexibilnejšie organizuje svoje učenie podľa meniacich sa podmienok. Učenie, ako sa učiť môže byť zlepšované školou a malo by sa stať súčasťou všetkých vyučovacích predmetov, súčasťou vyučovacích cieľov na všetkých úrovniach školského systému.

Štýl učenia závisí od pôsobenia okolitého prostredia (rodičov, učiteľov, spolužiakov, kamarátov), alebo môže byť riadený samostatne žiakom. Ukazuje sa, že žiak nedokáže vlastnými silami meniť štýl učenia, nie je schopný systematickej autoregulácie, nedokáže sa kvalitne rozvinúť. Najvýhodnejšie je tieto dve možnosti vzájomne doplňovať a snažiť sa o ich zlučiteľnosť.

## **2.2 Osobnosti ovplyvňujúce štýl učenia**

Veľké nároky a zodpovednosť za zmenu žiakovho štýlu učenia sa očakáva od učiteľov. Ovplyvňovať štýl učenia je veľmi delikátna a citlivá záležitosť zo strany učiteľa, ktorá zasahuje do jeho vnútorného sveta. Sám učiteľ nemôže plne ovplyvniť zmeny na strane žiaka, lebo nedokáže zabrániť pôsobeniu ďalších faktorov, ako sú jeho zvláštnosti, rodinné zázemie, podmienky vyučovania, učenia sa žiaka a pod.

Niekedy nastáva situácia, že žiaci rozumovo vedia, čo by mali robiť inak, ale návyky a emócie im to nedovolia. Práve v týchto prípadoch potrebuje žiak pomoc učiteľov. Pomoc môže byť poskytnutá v troch úrovniach:

- a) na úrovni jednotlivca - diagnostikovať žiakov štýl učenia, vysvetliť mu prednosti a nedostatky štýlu učenia, ktorý žiak preferuje. Cieľom je, aby pochopil príčiny nesúlady medzi svojim učením a požiadavkami učiteľov na učenie.
- b) na úrovni triedy - diagnostikovať štýl učenia všetkých žiakov v triede, diagnostikovať štýl učenia a vyučovací štýl učiteľa s cieľom zmeniť vyučovacie metódy, ktoré zodpovedajú profilu triedy.

c) na úrovni školy - škola sa môže rozhodnúť pre výučbu založenú na štýloch učenia. Napríklad zabezpečiť pre učiteľov študijné materiály pre zmenu štýlu učenia, školenia, overovať rôzne metódy diagnostikujúce štýly učenia.

R. Parwatt (1991) navrhuje, aby počas vyučovacieho procesu učiteľ využíval skryté prístupy. Žiaci pracujúci s učivom si ani neuvedomujú, že sa jedná o ďalšiu požiadavku učiteľa. Až keď sa žiaci naučia a pochopia užitočnosť zvoleného postupu, učiteľ začne upozorňovať na prvky autoregulácie a prenáša ich do uvedomelého postupu žiakov.

Smith (1996) identifikoval niekoľko znakov, ktoré zlepšujú učebnú stratégiu žiakov:

1. prostredníctvom skupinovej práce podporovať osobný rozvoj žiaka,
2. stimulovať samostatné učenie prostredníctvom kontaktu medzi učiteľom a žiakom,
3. predkladať žiakom problémy z reálneho života,
4. viesť žiakov k tomu, aby sa zamýšľali nad svojim postupom pri učení,
5. organizovať samostatné skupinové práce bez pomoci učiteľa,
6. vo vyučovacom procese využívať prvky simulácie, hier a pod.,
7. pomáhať žiakom, aby si uvedomili požiadavky zadanej úlohy a účel učenia sa,
8. účasť žiakov na projektovej výučbe,
9. zmena tradičných metód, aby vyhoveli zvláštnostiam žiakom.

Ochota žiaka zmeniť svoj štýl učenia závisí aj na požiadavkách učiteľa. Žiak uvažuje, na čo učiteľ pri skúšaní kladie dôraz, čo od neho očakáva a podľa toho sa snaží zariadiť.

Na zmene stratégie učenia a štýlov učenia sa môžu podieľať psychológovia. Žiaci ich chápu ako nestranných partnerov, ktorí bránia ich záujmy. Poradenská činnosť môže mať rôzne podoby, napríklad konzultácie so žiakmi, skupinová práca žiakov, ďalšie vzdelávanie učiteľov v problematike štýlov učenia, konzultácie s rodičmi a pod. Nevýhodou psychológov oproti učiteľom je, že nemajú možnosť spájať nové stratégie učenia s učivom.

Ukazuje sa, že pre deti sú veľkou inšpiráciou a pomocou v učení súrodenci, spolužiaci a kamaráti, ktorí vystupujú ako "komerčný vzor". Dôvodom je tá skutočnosť, že rovesníci sú vhodným vzorom, ľahko dosiahnuteľní a dokážu rozprávať v rovnakých pojmoch. Rada súčasných výskumov, uprednostňujúca sociálne učenie, realizuje pedagogický proces v dvojiciach, alebo v malých skupinách tzv. kooperatívne učenie. Skupina je zámerne zostavená tak, aby každý jedinec bol expertom v niečom inom a tomu naučil svojich spolužiakov. Ďalším variantom je partnerské učenie, kde výučba prebieha v dvojiciach. Jeden žiak je pokročilejší, skúsenejší, zdatnejší a vyučuje svojho spolužiaka. Obidvaja sa učia ako zvládnuť nové učivo, môžu sa radiť, vzájomne pomáhať, učenie potom prebieha efektívnejšie a jednoduchšie.

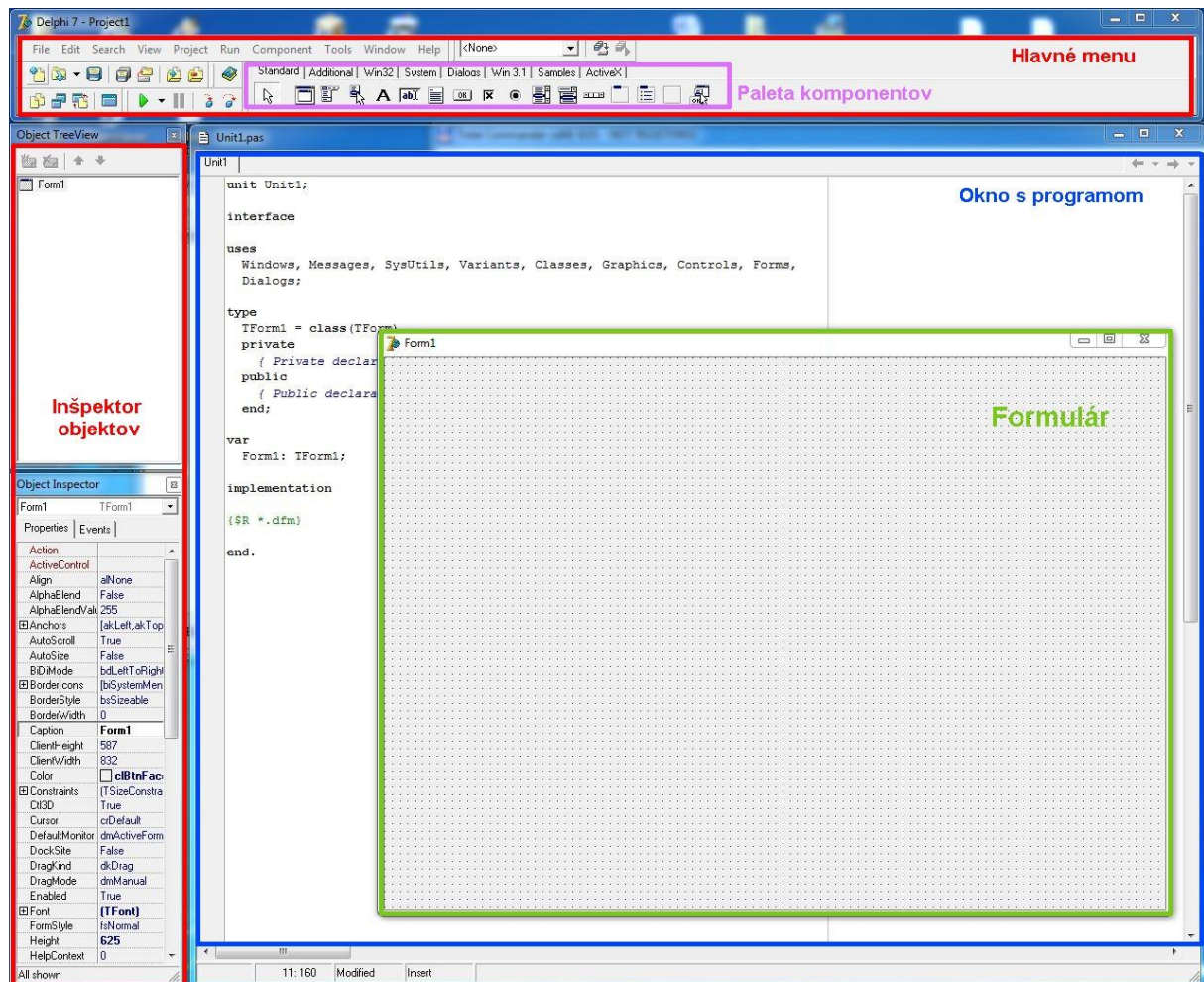
Základy štýlov učenia sa kladú v rannom veku a práve vtedy rodičia môžu vykonať pre svoje dieťa mnoho pozitívneho, ale aj negatívneho. Zatiaľ v tomto smere neexistuje ucelený záver.

## 3 ZÁKLADY PROGRAMOVANIA V DELPHI

Naším cieľom je naučiť základy programovania v objektovo orientovanom prostredí Delphi založený na jazyku Pascal. Pri zápise algoritmov chceme naučiť používať grafické príkazy, premenné, programové konštrukcie, cykly a podmienky.

### 3.1 Prostredie Delphi

Pri spustení Delphi sa na vašom monitore zobrazí jeho prostredie, ktoré môžeme rozdeliť na tieto časti:




1. *Hlavné menu* – je podobné ako menu v iných programoch napr. Word, Excel. Pre nás sú na začiatok dôležité tieto tlačidlá v ponuke **File**:
  - a. *New* → *Application* : začať nový projekt
  - b. *Open Projects* – otvor existujúci projekt
  - c. *Save* – uložiť projekta ponuka *Run* → *Run*, kde spustíme modul.
2. *Paleta komponentov* – obsahuje komponenty, z ktorých budeme vytvárať programy,
3. *Inšpektor objektov* – umožňuje meniť (nastavovať) parametre pre komponenty umiestnené vo formulári,

4. *Okno s programom* – slúži na zápis algoritmov (správanie programu v rôznych situáciách),
5. *Formulár* – objekt, ktorý dáva vzhľad celému nášmu projektu a slúži na vkladanie komponentov.

Novú aplikáciu (program) vytvoríme z ponuky **File** → *New* → *Application*. Hneď na začiatku tvorby programy je potrebné program uložiť, aby sme sa neskôr vyhli niektorým chybám. Takže z hlavnej ponuky **File** vyberieme *Save Project As ...*

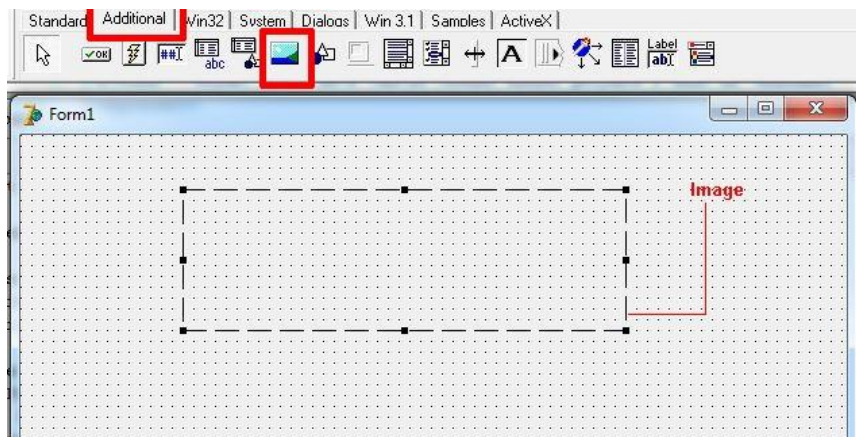
Spustenie programu realizujeme v ponuke **Run** → *Run* alebo tlačidlom 

Na obrazovke sa nám objaví prázdne šedé okno s nápisom *Form 1* a s malými systémovými tlačidlami na minimalizovanie, maximalizáciu a ukončenie aplikácie.

Bežiaci program ukončíme podobne ako iné programy, kliknutím na tlačidlo 

### 3.2 Grafická plocha - Image

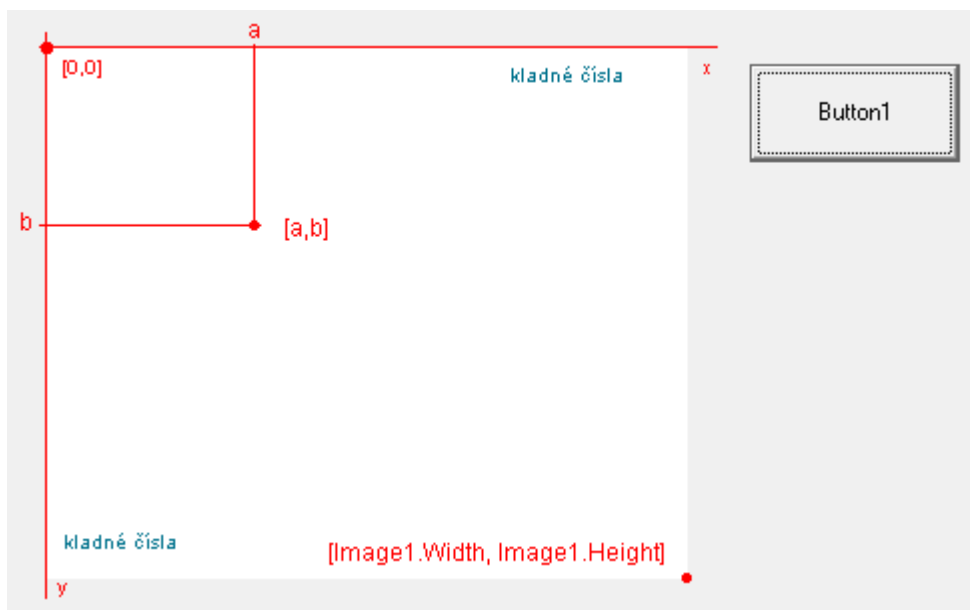
Grafická plocha - je to obdĺžnik vo formulári, do ktorého môžeme nástrojmi kresliť, písať, obsah ukladať na disk alebo zo súboru do nej čítať nejaký obrázok. Komponent *Image* nájdeme v paleta komponentov *Additional*, ktorý vložíme do plochy dvojkliknutím. Jeho veľkosť môžeme ťahaním meniť.



Kresliť do grafickej plochy budeme ako akciu pri zatlačení nejakého tlačidla. Napr. do procedúry *Button1Click* zapíšeme grafické príkazy medzi **begin** a **end**. Grafická plocha má meno **Image1**. Má svoje plátno pod názvom **Canvas**, do ktorého práve kreslíme. Kresliť môžeme pomocou neviditeľného pera (**Pen**), vyfarbovať nejaké plochy štetcom (**Brush**). **Pen** a **brush** majú svoje farby (**Color**) a hrúbku (**Width**).

Šírka grafickej plochy je daná vlastnosťou *Image1.Width* a výška je daná vlastnosťou *Image1.Height*. X-ové súradnice ľubovoľného bodu sú celočíselné a sú z intervalu  $\langle 0, \text{Image1.Width} \rangle$ . Y-ové súradnice ľubovoľného bodu sú tiež z intervalu  $\langle 0, \text{Image1.Height} \rangle$ . Počiatok súradnej sústavy (bod  $[0,0]$ ) je v ľavom hornom rohu grafickej plochy. **X**-sova os je orientovaná rovnako ako v matematike zľava doprava. **Y**-ová os je orientovaná opačne ako v matematike, zhora nadol ako je to vyznačené na obrázku.





Farby v prostredí Windows sú zložené z troch základných farieb: červenej, zelenej a modrej (**RGB: Red, Green, Blue**). Požadovanú farbu z týchto zložiek zadávame číslom od 0 do 255.

Preto farbu môžeme definovať podľa funkcie **RGB**, ktorá dostane tri čísla v desiatkovej sústave od 0 po 255.

*Niektoré preddefinované farby:*

clBlack	RGB(0,0,0)
clBlue	RGB(0,0,255)
clYellow	RGB(255,255,0)
clMaroon clGreen	RGB(128,0,0)
clOlive	RGB(0,128,0)
clNavy	RGB(128,128,0)
clPurple	RGB(0,0,128)
clTeal	RGB(128,0,128)
clGray	RGB(0,128,128)
clSilver	RGB(128,128,128)
clRed	RGB(192,192,192)
clLime	RGB(255,0,0)
	RGB(0,255,0)

Prameň: Machová, 2004.

### 3.3 Grafické príkazy

Grafická plocha ponúka na kreslenie rôznych objektov väčšie množstvo nástrojov.

**Medzi základné príkazy (procedúry) na prácu s perom:**

- **MoveTo(x,y)** - presunie pero bez kreslenia čiary
- **LineTo(x,y)** - kreslí úsečku od momentálnej pozície do bodu (x,y)
- **Rectangle(x1,y1,x2,y2)** - nakreslí obdĺžnik (štvorec), štyri čísla v zátvorkách vyjadrujú súradnice v dvoch protíľahlých vrcholov. Prvé dve sú x-ová a y-ová súradnica ľavého horného rohu a druhé dve pravého dolného rohu

- **Polygon**([Point(x1,y1),Point(x2,y2),Point(x3,y3),...]) - uzavretý N-uholník, napr. napr. Polygon([Point(10, 10), Point(30, 10), Point(130, 30), Point(240, 120)]);
- **Ellipse(x1,y1,x2,y2)** - zadávame súradnice opísaného obdĺžnika, ak je obdĺžnikom štvorec, nakreslí sa kruh

Pri kreslení je pero na začiatku nastavené na kreslenie čiernou farbou. **Vlastnosti pera môžeme meniť týmito príkazmi:**

- **Pen.Color := clRed** - nastaví farbu pera na červenú (cl = skratka zo slova color, Red = červená). Ďalšie mená farieb, napr.: clBlack, clWhite, clRed, clBlue, clYellow, clGreen, ...
- **Pen.Width := 3** - nastaví hrúbku pera 3
- **Pen.Style := psDash** - nastaví štýl kreslenia čiar na čiarkovaný, iné štýly ako psSolid (normálne čiary) fungujú len pre hrúbku 1 (psSolid – plná čiara, psDash - čiarkovaná, psDot - bodkovaná, psDashDot - bodkočiarkovaná, psDashDotDot - bodkobodkočiarkovaná, ...)

Pri kreslení objektov (napr. obdĺžnik, štvorec) sú tieto štandardne vyplnené bielou farbou. **Farbu výplne môžeme meniť nasledovne:**

- **Brush.Style := bsVertical** - nastaví šrafovanie, **bsSolid** je normálne vyplňanie, **bsClear** znamená obdĺžnik bez vyplneného vnútra, ďalšie štýly: bsSolid, bsClear, bsHorizontal, bsVertical, ...
- **Brush.Color:=clYellow** -nastaví farbu výplne alebo **Brush.Color:=RGB(255,255,0)**

**Príkazy pre výpis textového reťazca:**

**TextOut(x,y,textový-reťazec)** - textový reťazec 'nejaký text' napr. *TextOut(100,200,'Programovanie')*. Text píšeme medzi apostrofy pre písmo.

**Font.Color := farba;** - nastaví farbu písma

**Font.Height := výška** - t.j. veľkosť písma

**Font.Name := názov-fontu**- napr 'Arial','Times New Roman', ...

**Font.Size:= veľkosť** – veľkosť fonu

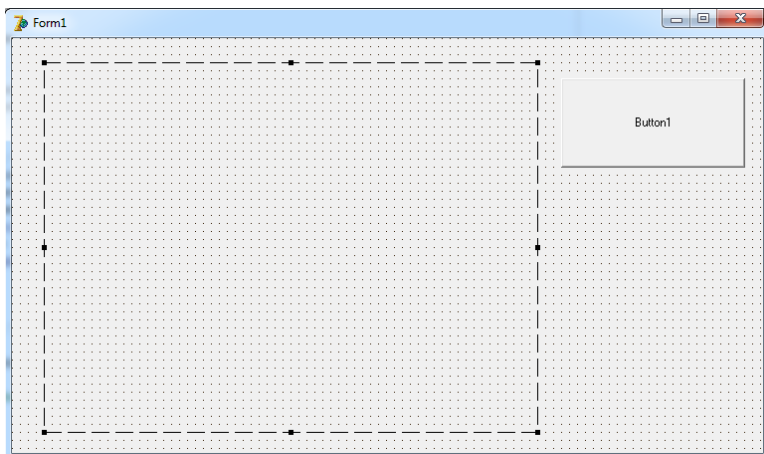
**Font.Style := štýl** - v hranatých zátvorkách je určený nejaký atribút písma, napr. [fsBold, fsItalic, fsUnderline, fsStrikeOut]

### 3.4 Náš prvý program

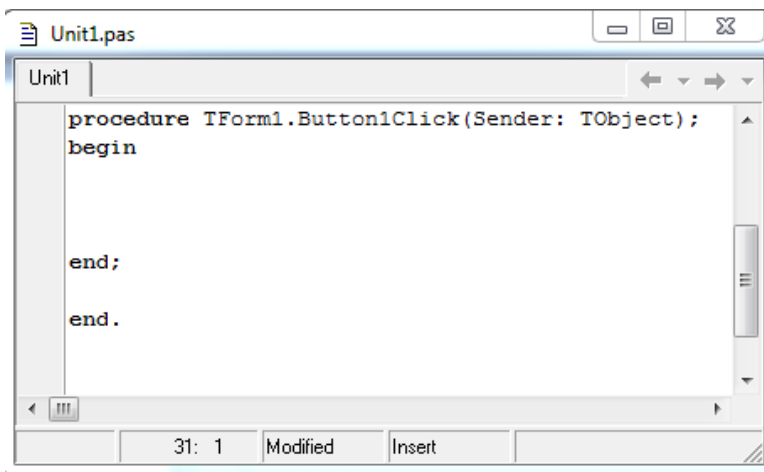
Spustenie prvého programu si ukážeme na jednoduchom príklade kreslenie obdĺžnika. Do prázdneho formulára vložíme *Image* (grafickú plochu) ako bolo vysvetlené predchádzajúcej kapitole. Ak máme grafickú plochu, kresliť do nej budeme pomocou akcie pri stlačení tlačidla **Button**. Dané tlačidlo položíme dvojklikom z palety **Standard**.



Formulár z grafickou plochou a tlačidlom Button bude vyzerat' nasledovne:



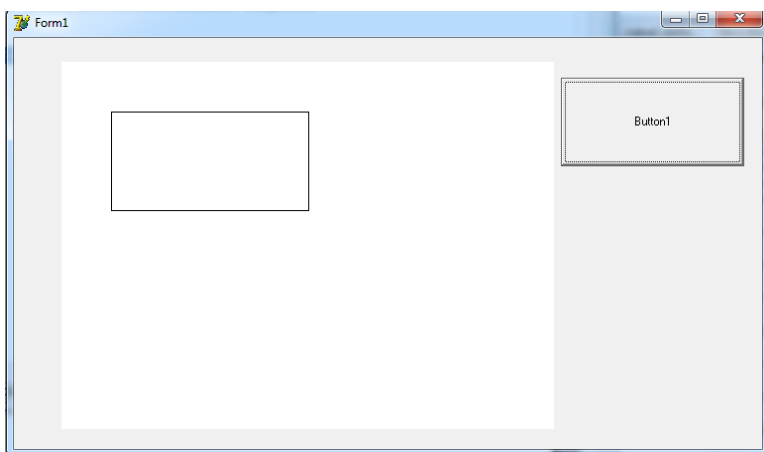
Dvojklikom vo formulári na tlačidlo **Button1** v editovanom okne sa otvorí úsek - procedúra **Button1Click**, do ktorého môžeme medzi **begin** a **end**; zapisovať grafické príkazy.



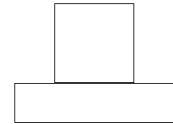
Pridáme grafický príkaz **Rectangle** a nakreslíme obdĺžnik. Zadáme mu súradnice dvoch protíahlých vrcholov rovnobežných s osami.

Napíšeme: `Image1.Canvas.Rectangle(50,50,250,150);` (za každým príkazom nasleduje bodkočiarka).

Po spustení programu (F9) a stlačení tlačidla sa nakreslí obdĺžnik.



Teraz na ukážku nakreslíme jeden obdĺžnik a štvorec, tak aby štvorec bol nad obdĺžnikom v strede:



```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    Image1.Canvas.Rectangle(100,100,200,200); // vykreslí nám štvorec
```

```
    Image1.Canvas.Rectangle(50,200,250,250); // vykreslí nám obdĺžnik
```

```
end;
```

Pomocou Rectangle kreslíme aj štvorce, len sa niekedy musíme so súradnicami "trochu pohrať", aby sme dostali útvary podľa našich predstáv – štvorec je tesne nad obdĺžnikom.

### 3.5 Kreslenie geometrických útvarov

Ukážeme si ako kresliť rôzne geometrické útvary a pracovať s farbami, textom.

Úloha 3.5.1: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Pomocou príkazov LineTo a MoveTo nakreslite domček.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    image1.Canvas.MoveTo(200,400);
```

```
    image1.Canvas.LineTo(200,200);
```

```
    image1.Canvas.LineTo(300,50);
```

```
    image1.Canvas.LineTo(400,200);
```

```
    image1.Canvas.LineTo(200,200);
```

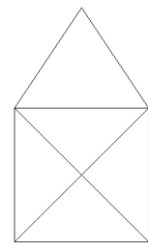
```
    image1.Canvas.LineTo(400,400);
```

```
    image1.Canvas.LineTo(200,400);
```

```
    image1.Canvas.LineTo(400,200);
```

```
    image1.Canvas.LineTo(400,400);
```

```
end;
```



Úloha 3.5.2: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Pomocou príkazu Rectangle a LineTo nakreslite domček, ktorého štvorec bude modrým pero a strieška hrubým červeným perom.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    Image1.Canvas.Pen.Color := clBlue; //nastavenie modrej farby pera
```

```
    Image1.Canvas.Pen.Width := 1; //nastavenie hrúbky pera
```

```
    Image1.Canvas.Rectangle(50,100,150,200);
```

```
    Image1.Canvas.MoveTo(50,100);
```

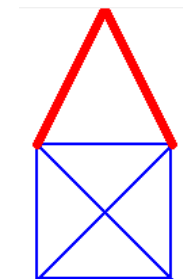
```
    Image1.Canvas.LineTo(150,200);
```

```
    Image1.Canvas.MoveTo(150,100);
```

```
    Image1.Canvas.LineTo(50,200);
```

```
    Image1.Canvas.Pen.Color := clRed; //nastavenie červenej farby pera
```

```
    Image1.Canvas.Pen.Width := 7;
```



```
Image1.Canvas.MoveTo(50,100);
Image1.Canvas.LineTo(100,0);
Image1.Canvas.MoveTo(100,0);
Image1.Canvas.LineTo(150,100);
```

**end;**

Úloha 3.5.3: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Vytvorte pomocou príkazu Rectangle kríž vytváraním štvorčiek ľubovoľnej výplne.

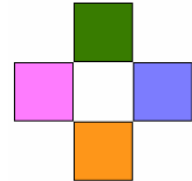
Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
Image1.Canvas.Brush.Color:=RGB(255,124,255);
```

```
Image1.Canvas.Rectangle(50,50,100,100);
Image1.Canvas.Brush.Color:=RGB(124,124,255);
Image1.Canvas.Rectangle(150,50,200,100);
Image1.Canvas.Brush.Color:=RGB(56,124,0);
Image1.Canvas.Rectangle(100,0,150,50);
Image1.Canvas.Brush.Color:=RGB(255,150,25);
Image1.Canvas.Rectangle(100,100,150,150);
```

**end;**



V tomto príklade je názorne ukázané ako môžeme generovať ľubovoľnú farbu podľa našich predstáv pomocou funkcie RGB. Všetky nastavenia pre daný štvorec musíme napísať ešte pred jeho vykreslením. Tieto nastavenia budú platiť pre všetky ďalšie útvary, kým ich nezmeníme.

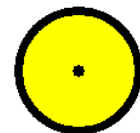
Úloha 3.5.4: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Nakreslite „koleso“ hrubou čiernou farbou a žltou výplňou.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
Image1.Canvas.Pen.Color := clBlack; //nastaví sa farba pera
Image1.Canvas.Brush.Color:=clYellow; //nastaví sa farba výplne
Image1.Canvas.Pen.Width := 7; // hrúbka pera
Image1.Canvas.Ellipse(50,220,150,320); //kreslenie vonkajšieho kruhu
Image1.Canvas.Ellipse(98,268,102,272); //kreslenie vnútorného kruhu
```

**end;**



Úloha 3.5.5: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Nakreslite snehuliaka, ktorý bude mať modrú výplň.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);
begin
```

```
Image1.Canvas.Pen.Color := clBlack;
Image1.Canvas.Brush.Color := clBlue;
```



```

Image1.Canvas.Pen.Width := 1;
Image1.Canvas.Ellipse(240,50,280,90);
Image1.Canvas.Ellipse(230,90,290,150);
Image1.Canvas.Ellipse(220,150,300,230);

```

**end;**

Úloha 3.5.6: Formulár obsahuje tlačidlo Button1 a obrázok Image1. Pomocou príkazu Polygon vytvorte šípku zeleným perom a červenou výplňou ako je na ukážke.

Riešenie:

```

procedure TForm1.Button5Click(Sender: TObject);
begin

```

```

Image1.Canvas.Brush.Color := clRed;
Image1.Canvas.Pen.Width := 2;
Image1.Canvas.Pen.Color := clGreen;
Image1.Canvas.Polygon([Point(100,20),Point(140,100),Point(100,80),Point(60,100
)]);
end;

```



Úloha 3.5.7: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Na vytvorenie loga, ktorý sa skladá z kružnice a trojuholníka použite príkazy Polygon a Ellipse.

Riešenie:

```

procedure TForm1.Button1Click(Sender: TObject);
begin

```

```

Image1.Canvas.Pen.Color := clBlack;
Image1.Canvas.Pen.Width := 2;
Image1.Canvas.Brush.Color := clYellow;
Image1.Canvas.Ellipse(44,120,156,231);
Image1.Canvas.Brush.Color := clYellow;
Image1.Canvas.Ellipse(73,147,127,201);
Image1.Canvas.Brush.Color := clGreen;
Image1.Canvas.Polygon([Point(100,120),Point(150,200),Point(50,200)])

```

**end;**



Pri práci s grafikou je potrebné ešte uviesť mazanie grafickej plochy. Plochu zmažeme, ak pridáme ďalšie tlačidlo napr. **Button2**. Pridáme príkaz **FillRect** a nastavíme farbu výplne akou chceme prefarbiť grafickú plochu:

```

procedure TForm1.Button1Click(Sender: TObject);
begin

```

```

Image1.Canvas.Brush.Color := clWhite; //nastavenie farby výplne
Image1.Canvas.Fillrect (Image1.ClientRect); //mazanie grafickej plochy

```

**end;**

Úloha 3.5.8: Formulár obsahuje tlačidlo Button1 a grafickú plochu Image1. Vypíšte súčasne vedľa seba slova „Informatika“ modrou farbou a slovo „Delphi“ žltou farbou.

Riešenie:

```

procedure TForm1.Button1Click(Sender: TObject);

```

**begin**

```
Image1.Canvas.Font.Size :=18;  
Image1.Canvas.Font.Color:=clBlue;  
Image1.Canvas.TextOut(100,100,'Informatika');  
Image1.Canvas.Font.Color:=clRed;  
Image1.Canvas.TextOut(100,200,'Delphi');
```

Informatika

Delphi

**end;**

**end.**

### 3.6 Deklarácia celočíselných premenných a náhodnosť

Premenná pri programovaní predstavuje určitú oblasť pamäte, v ktorej je uložená hodnota a túto hodnotu môžeme neskôr používať. Väčšinou má premenná svoje meno – *identifikátor premennej*. Ten musí spĺňať určité pravidlá:

- nesmie začínať číslicou (napr. 3krok, 2cislo a pod.);
- nesmie sa zhodovať s *rezervovaným slovom* Pascalu (napr. Do, To, Of, Begin a pod.),
- obsahovať písmena s diakritikou (napr. dĺžka, čas a pod.);
- všetky identifikátory musia byť navzájom rôzne (napr. Abc, aBc označujú tu istú premennú),
- meno premennej má vyjadrovať hodnotu, ktorú do nej ukladáme, napr. názvy, ktoré požívame v matematike alebo vo fyzike (x, y, súčet, výsledok, x1, y2 a pod.)

Ak plánujeme v programe používať premenné, musím ich najprv *zadeklarovať* (oznámiť počítaču, aby vyhradil pamäťové miesto pre naše premenné a zapamätal si ich mená). Deklarácie premenných začínajú slovom **var** (z anglického slova variable t.j. premenná). Za tým nasleduje identifikátor premennej, dvojbodka a slovo **Integer**. Integer označuje, že premenná bude celé číslo. Táto deklarácia sa ukončí bodkočiarkou. Premenné deklarujeme časti programu medzi riadkami **procedure** a **begin**.

Ako zadeklarovať premenné si ukážeme na nasledujúcej úlohe: Je potrebné nakresliť kruh veľkosti  $d = 80$ , ktorého začiatkové súradnice  $x = 150$  a  $y = 100$ .

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
  x, y, d: Integer;
```

```
begin
```

```
  d:= 80;
```

```
  x:= 150;           // x = 150 y = ?
```

```
  y:= 100;          // x = 150 y = 100
```

```
  Image1.Canvas.Ellipse(x, y, x + d, y + d); // Ellipse (150,100,250,200)
```

```
end;
```

Ako ste si všimli, v predchádzajúcej úlohe sme v programe medzi begin a end vložili premenným pomocou prirad'ovacieho príkazu hodnoty. Prirad'ovací príkaz má zápis:

```
premenná := hodnota;
```

Pri riešení rôznych úloh budeme potrebovať vypisovať číselné hodnoty do grafickej plochy.

Na tieto účely nám pomôže pomocná funkcia, ktorá *prekonvertuje* celé číslo na text, aby sa dal vypísať. Funkcia **IntToStr** ľubovoľné číslo prevedie na textový tvar. Takže, ak chceme nejakú číselnú hodnotu (alebo premennú, alebo aritmetický výraz) vypísať pomocou **TextOut**, zapíšeme to takto:

```
Image1.Canvas.TextOut (x,y IntToStr (celočíselná hodnota));
```

*Úloha 1:* Vypíšte na grafickej ploche číslo 112 a to väčším písmom:

*Riešenie:*

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
    x: Integer; // premenná x je celé číslo
```

```
begin
```

```
    x:= 112;           // premennej x priradíme hodnotu 112
```

```
    Image1.Canvas.Font.Height:=40;
```

```
    Image1.Canvas.TextOut (100, 100, IntToStr (112));
```

```
end;
```

Celé čísla (**Integer**) v počítači zaberajú 32-bitov (4 bajty). Celé čísla pokrývajú interval  $< -2\ 147\ 483\ 648, 2\ 147\ 483\ 647 >$  čo je približne  $+ \text{alebo} - 2^{31}$ . Medzi základné operácie s celými číslami patrí:

- **+**, **-**, **\*** sú štandardné operácie pre súčet, rozdiel a súčin - ich výsledok by mal byť opäť v intervale celých čísel,
- **div** je celočíselné delenie, nesmie sa deliť nulou,
- **mod** je zvyšok po celočíselnom delení (napr.  $23 \bmod 7 = 2$ ).

Žrebovanie tomboly, prípadne hry alebo rôzne testovacie programy vyžadujú náhodné hodnoty. Pascal na tento účel používa funkciu **Random**. **Random(n)** vracia náhodné číslo z intervalu  $<0, n-1>$ . Napríklad **random (7)** vyberie náhodne číslo od *nuly* po *šesť*. Aby sme mohli so zvoleným číslom ďalej pracovať, je potrebné ho uložiť do premennej napríklad:

```
x := random (50); // zvolí sa číslo od 0 do 49
```

Ak nechceme mať náhodne číslo od nuly, potom stačí pripočítať spodnú hranicu čísla, ktorú chceme zvoliť napríklad:

```
x := 1 + random (50); // vyberie sa číslo od 1 do 50
```

```
x := 10 + random (51); // vyberie sa číslo od 10 do 60
```

Ak budeme program spúšťať viackrát, môže nastať situácia, že náhodná hodnota sa bude opakovať. Na to nám slúži príkaz **randomize**, ktorý sa umiestni na začiatok procedúry a nastaví generátor náhodných čísel na naozaj náhodnú hodnotu.

*Úloha 3.6.1:* Do formulára vložte tlačidlo, po stlačení sa na náhodnú pozíciu vypíše červené slovo Delphi.

*Riešenie:*

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
begin
```

```
    randomize;
```

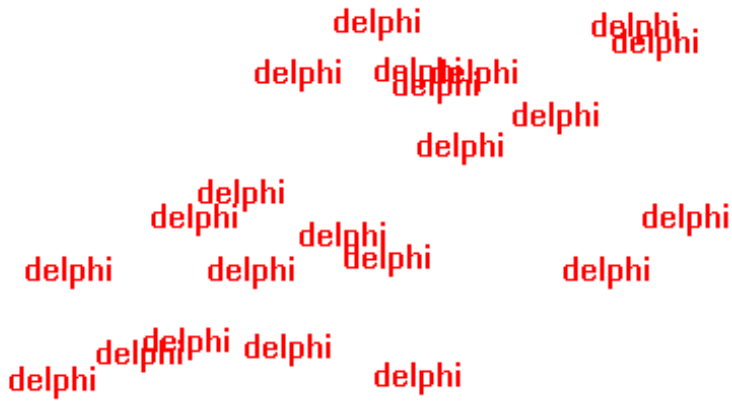
```
    Image1.Canvas.Brush.Style:=bsClear;
```

```
    Image1.Canvas.Font.Height:=20;
```



```
Image1.Canvas.Font.Color:=clred;  
Image1.Canvas.TextOut(random(505),random(433),'delphi');
```

```
end;  
end.
```



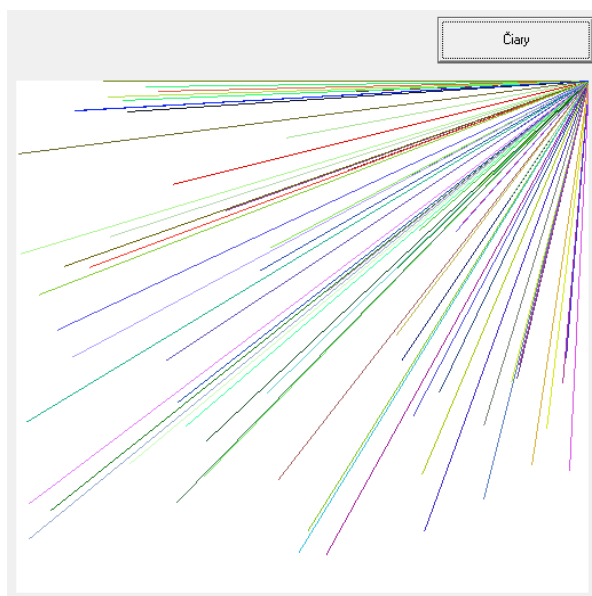
Úloha 3.6.2: Vo formulári máme grafickú plochu a tlačidlo. Vykreslite čiary z pravého horného rohu grafickej plochy náhodnej dĺžky a náhodnej farby. Grafická plocha má súradnice (0,0,500,500)

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin
```

```
    randomize;  
    Image1.Canvas.Pen.Width := 1;  
    Image1.Canvas.Pen.Color:=RGB(Random(256), Random(256), Random(256));  
    Image1.Canvas.MoveTo(500, 0);  
    Image1.Canvas.LineTo(random(500), Random(500));
```

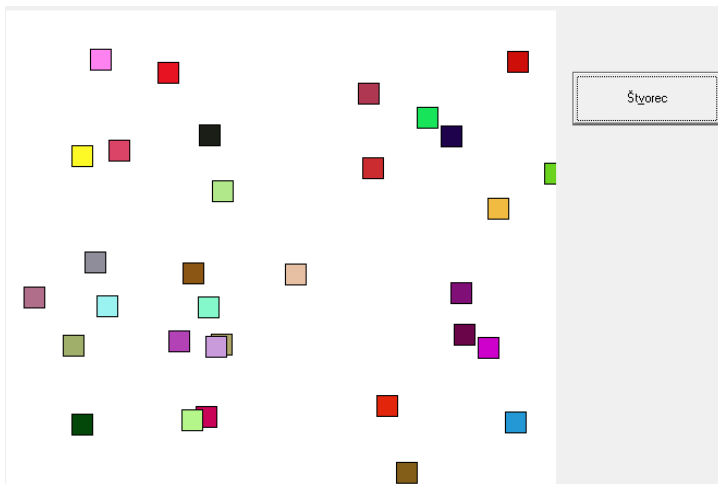
```
end;  
end.
```



Úloha 3.6.3: Vo formulári máme grafickú plochu a tlačidlo. Po stlačení tlačidla sa vykreslí na náhodné miesto do obdĺžnika (grafickej plochy) so súradnicami (0,0,450,400) štvorec so stranou 20. Zároveň štvorce budú mať náhodnú výplň.

Riešenie:

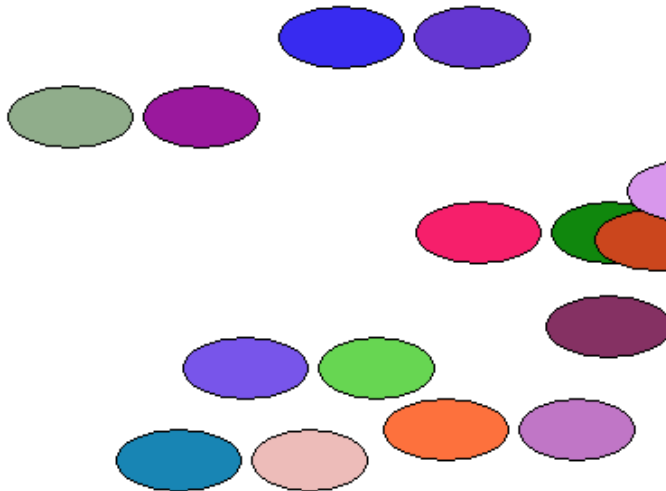
```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  var x,y:Integer;  
  begin  
    randomize;  
    Image1.Canvas.Brush.Color:=RGB(random(256),random(256),random(256));  
    x:=random(450);  
    y:=random(400);  
    Image1.Canvas.Rectangle(x,y,x+20,y+20);  
  end;
```



Úloha 3.6.4: Vykreslite vedľa seba po každom stlačení tlačidla dve elipsy rovnakej veľkosti, rovnakej vzdialenosti od seba, ale náhodnej farby a na náhodnom mieste grafickej plochy.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);  
var x,y:integer;  
begin  
  randomize;  
  Image1.Canvas.brush.Color:=RGB(random(255),random(255),random(255));  
  x:=random(505);  
  y:=random(433);  
  Image1.Canvas.Ellipse(x,y,x+70,y+35);  
  Image1.Canvas.brush.Color:=RGB(random(255),random(255),random(255));  
  Image1.Canvas.Ellipse(x+75,y,x+140,y+35);  
end;  
end.
```



### 3.7 Cyklus For

Napríklad text „Delphi“ by sme na náhodnom mieste grafickej plochy vykreslili takto:

*Image1.Canvas.TextOut (Random(300), Random(300), 'Delphi');*

Keby sme chceli tento text naraz vykresliť napríklad desaťkrát, potom by sme museli tento príkaz opakovať (kopírovať).

Na takéto účely existuje v Pascale pre opakovanie príkaz **cyklu**. Keďže presne vieme, koľkokrát sa má text „Delphi“ zopakovať, použijeme cyklus s pevným počtom opakovaní - **for cyklus**.

Jeho všeobecný zápis v Pascale je nasledovný:

```
for riadiaca premenná := dolná hranica to horná hranica do
  iba jeden príkaz;
```

Náš zápis algoritmu by vyzeral nasledovne:

**begin**

***for** i:=1 **to** 10 **do***

*Image1.Canvas.TextOut (Random(300), Random(300), 'Delphi');*

**end;**

Cyklus začína slovom **for** a za ním nasleduje premenná **i** (*riadiaca premenná*), ktorá slúži ako počítadlo cyklu. Je to premenná, ktorej si program postupne počíta opakovanie cyklu. U nás je to na začiatku **1** (štartová hodnota cyklu), potom sa bude tejto premennej hodnota postupne zväčšovať o jedna, kým nadobudne hodnotu hornej hranice cyklu (hodnota **10**). Pre každú túto hodnotu sa vykoná **telo cyklu**, jeden príkaz *Image1.Canvas.TextOut (Random(300), Random(300), 'Delphi');*

Premenná cyklu **i** musí byť deklarovaná ešte pred prvým **begin** v časti **var**.

Úloha 3.71: Stlačenie tlačidla na náhodne pozície vypíše desaťkrát slovo „Delphi“.

Riešenie:

**procedure** TForm1.Button1Click(Sender: TObject);

**var**

*i:Integer;*

**begin**



```

    for i:=1 to 10 do
      Image1.Canvas.TextOut (Random(300), Random(300), 'Delphi');
end;
end.

```

Úloha 3.7.2: Stlačenie tlačidla nám do radu vedľa seba vykresli 10 štvorčekov zelenej farby.

**Riešenie:**

```

procedure TForm1.Button1Click(Sender: TObject);
var i:integer;
begin
  Image1.Canvas.Brush.Color:=clGreen;
  for i:=1 to 10 do
    Image1.Canvas.Rectangle(30*i,30,30*i+10,40);
end;
end.

```



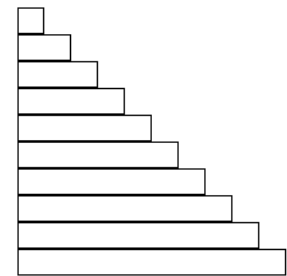
Úloha 3.7.3: Použitím príkazu Rectangle a cyklu for nakreslite schody.

**Riešenie:**

```

procedure TForm1.Button4Click(Sender: TObject);
var
  i:integer;
begin
  for i:=1 to 10 do
    begin
      Image1.Canvas.Rectangle(20, 20*i, 20*i+20, 20*i+20);
    end;
end;
end.

```

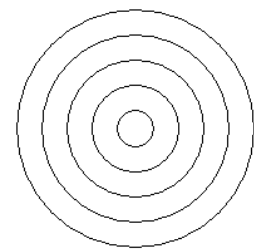


Úloha 3.7.4: Nakreslite terč z piatich kružníc ako je na obrázku.

```

procedure TForm1.Button1Click(Sender: TObject);
var
  i:integer;
begin
  Image1.Canvas.Brush.Style:=bsClear;
  for i:=1 to 5 do
    Image1.Canvas.Ellipse(50+(i-1)*20,50+(i-1)*20,240-(i-1)*20,240-(i-1)*20);
end;
end.

```



Ak potrebujeme opakovať nie jeden príkaz, ale postupnosť viacerých príkazov, na to slúži **zložený príkaz cyklu**. Postupnosť príkazov je uzavretá medzi slovami **begin** a **end**:

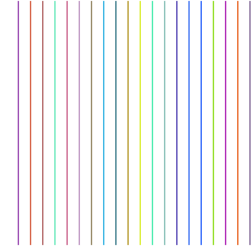
```

for riadiaca premenná := dolná hranica to horná hranica do
  begin
    viac príkazov;
  end

```

```
end;  
end.
```

Úloha 3.7.5: Urobte program, ktorý na stlačenie Button-u vykreslí na Image1 nasledovný obrázok, plot pomocou zvislých rôznofarebných čiar:



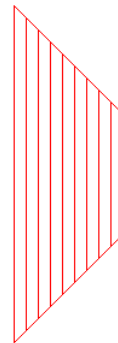
Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
i:integer;  
begin  
  for i:=1 to 20 do  
    begin  
      image1.Canvas.Pen.Color:=RGB(random(255),random(255),random(255));  
      image1.Canvas.MoveTo(10*i,100);  
      image1.Canvas.LineTo(10*i,300);  
    end;  
end;  
end.
```

Úloha 3.7.6: Nakreslite harfu pomocou červených čiar, ktoré sa budú postupne skracovať podobne ako je to na obrázku.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);  
var  
i:integer;  
begin  
  Image1.Canvas.Pen.Color:=clRed;  
  image1.Canvas.MoveTo(110,110);  
  image1.Canvas.LineTo(200,200);  
  image1.Canvas.MoveTo(110,390);  
  image1.Canvas.LineTo(200,300);  
  for i:=1 to 10 do  
    begin  
      image1.Canvas.MoveTo(100+10*i,100+10*i);  
      image1.Canvas.lineto(100+10*i,400-10*i);  
    end;  
end;  
end;  
end.
```



Úloha 3.7.6: Na náhodnom mieste grafickej plochy vykreslite 10 čiar náhodnej farby a náhodnej dĺžky a smeru, ale tak, aby nepresiahli okraje grafickej plochy.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);  
var x,y,i:integer;  
begin
```



```

Image1.Canvas.Brush.Color:=clBtnFace;
Image1.Canvas.Pen.Color:=clBtnFace;
Image1.Canvas.Rectangle(0,0,Image1.Width,Image1.Height);
randomize;
  for i:=1 to 10 do begin
    x:=random(Image1.Width); // premennej x priradíme šírku grafickej plochy
    y:=random(Image1.Height); // premennej y priradíme dĺžku grafickej plochy
    Image1.Canvas.MoveTo(x,y);
    x:=random(Image1.Width);
    y:=random(Image1.Height);
    Image1.Canvas.Pen.Color:=RGB(random(256),random(256),random(256));
    Image1.Canvas.Pen.Width:=2;
    Image1.Canvas.LineTo(x,y);
  end;
end;
end.

```

Telo cyklu môže obsahovať ďalší príkaz cyklu, kedy hovoríme o cykle v cykle, teda o vnorenom cykle.

Vnorený cyklus na poslúži napríklad na vykreslenie štvorcovej siete 10x10 štvorčekov.

```

procedure TForm1.Button1Click(Sender: TObject);

```

```

var

```

```

i,j:integer;

```

```

begin

```

```

  for i:=1 to 10 do

```

```

    begin

```

```

      for j:=1 to 10 do // vnorený cyklus s riadiacou
premenou j

```

```

        begin

```

```

          image1.Canvas.Rectangle(20+20*j,20+20*i,40+20*j,40+20*i);

```

```

        end;

```

```

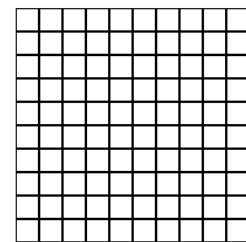
      end;

```

```

end;

```



Úloha 3.7.7: Nakreslite štvorcovú sieť 10x10, štvorčeky budú mať červenú výplň a budú od seba rovnomerne rozmiestnené. Zadeklarujeme premennú počet štvorčekov  $n = 10$  a premennú veľkosť štvorčeka  $a = 30$ .

Riešenie:

```

procedure TForm1.Button1Click(Sender: TObject);

```

```

var

```

```

n,i,j,a:integer;

```

```

begin

```

```

a:=30;

```

```

n:=10;

```

```

Image1.Canvas.Brush.color:=clred;

```

```

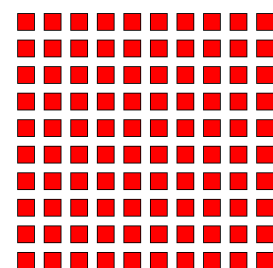
  for i:=1 to n do

```

```

    for j:=1 to n do

```



```

    Image1.Canvas.Rectangle(j*a+10,i*a+10,j*a+a,i*a+a) ;
end;
end.

```

Úloha 3.7.8: Použitím vnoreného cyklu nakreslite postupnosť čísel od 1 – 100 po desiatich riadkoch a desiatich stĺpcoch, ale v opačnom poradí, od najväčšieho čísla po najmenšie.

Riešenie:

```

procedure TForm1.Button25Click(Sender: TObject);
var
  i,j:integer;
begin
    for i:=1 to 10 do
      begin
        for j:=1 to 10 do
          begin
            image1.Canvas.TextOut(222-20*j,222-20*i,inttostr(i*10+j-10));
          end;
        end;
      end;
    end;
end;

```

```

100 99 98 97 96 95 94 93 92 91
90 89 88 87 86 85 84 83 82 81
80 79 78 77 76 75 74 73 72 71
70 69 68 67 66 65 64 63 62 61
60 59 58 57 56 55 54 53 52 51
50 49 48 47 46 45 44 43 42 41
40 39 38 37 36 35 34 33 32 31
30 29 28 27 26 25 24 23 22 21
20 19 18 17 16 15 14 13 12 11
10 9 8 7 6 5 4 3 2 1

```

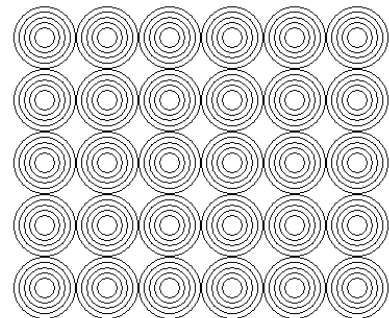
Úloha 3.7.9: Vytvorte kreslením piatich sústredných kružníc sieť veľkosti 6x6 terčov.

Riešenie:

```

procedure TForm1.Button23Click(Sender: TObject);
var i,j,r:integer;
begin
  for i:=1 to 5 do
    begin
      for j:=1 to 6 do
        begin
          for r:=1 to 5 do
            begin
              image1.Canvas.Ellipse(100+5*r+60*j,100+5*r+i*60,170-5*r+60*j,170-5*r+i*60);
            end;
          end;
        end;
      end;
    end;
  end;
end;

```



### 3.8 Podmienený príkaz

Pri programovaní sa môže stať, že náš program a jeho správanie závisí od rôznych okolností. Na základe nejakej podmienky (napríklad výberu nejakej hodnoty, premennej a pod.) sa má vykonať jeden príkaz alebo druhý príkaz, prípadne nejaké ďalšie príkazy. V Delphi sa používa príkaz **if**, podobne ako u iných programovacích jazykoch, ktorý dokáže vetviť program

Neúplná podmienka:

```
if podmienka then príkaz;
```

alebo

```
if podmienka then  
begin  
    príkaz1;  
    príkaz2; ....  
end;
```

Ak je podmienka splnená vykoná sa *príkaz*, respektíve postupnosť príkazov medzi *begin* a *end*.

Použitie neúplnej podmienky si ukážeme na nasledujúcej úlohe. Vypíšeme na grafickej ploche len všetky párne čísla od 1 do 20.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
i: integer;
```

```
begin
```

```
    for i:=1 to 20 do
```

```
        begin
```

```
            if (i mod 2 = 0) then //zvyšok po delení 2 je nula, vtedy je číslo párne
```

```
                Image1.Canvas.TextOut(10*i,20,IntToStr(i));
```

```
            end;
```

```
end;
```

```
end.
```

2 4 6 8 10 12 14 16 18 20
---------------------------

Úplná podmienka:

```
if podmienka then  
    príkaz 1  
    else  
    príkaz 2;  
end;
```

Program má vypísať všetky párne a nepárne čísla do 20 v dvoch radoch na grafickú plochu a to použitím úplnej podmienky. V prvom rade budú párne čísla a v druhom rade všetky nepárne čísla.

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
i:Integer;
```

```
begin
```

```
i:=5;
```

```
for i:=1 to 20 do
```

```
    begin
```

```
        if (i mod 2 = 0) then //zvyšok po delení 2 je 0
```

```
            Image1.Canvas.TextOut(10*i,20,IntToStr(i))
```

```
        else
```

```
            Image1.Canvas.TextOut(10*i,40,IntToStr(i));
```

```
        end;
```

```
end;
```

2 4 6 8 10 12 14 16 18 20
1 3 5 7 9 11 13 15 17 19



**end.**

Keď je podmienka splnená, vykoná sa *príkaz 1*. Ak podmienka nie je splnená, vykoná sa za slovom **else** *príkaz 2*. Príkaz 1 a príkaz 2 môžeme nahradiť aj postupnosťou príkazov, ak ich uzavrieme medzi **begin** a **end**. Je dôležité upozorniť na to, že pred slovom *else* nikdy nesmie byť bodkočiarka.

Pri programovaní máme k dispozícii viacero typov operátorov:

**+** - ...sčítanie, odčítanie, napr. **a - b**  
**\*** / ...násobenie a desatinné delenie, napr. **i\*10**  
**div** ...celočíselné delenie, napr. **x div 2**  
**mod** ...zvyšok po delení, napr. **x mod 2**  
**and or** ...logické spojky "a zároveň" "alebo", napr. **(x mod 2 = 0) and (x < 100)**

Pre porovnávanie dvoch hodnôt ( dvoch premenných) alebo premenných s číslom používame tieto relačné operácie:

**>** ak je väčší ako, napr. **x > 1**  
**>=** ak je väčší alebo rovný ako, napr. **x >= 50**  
**=** ak je rovná sa, rovnaké hodnoty, napr. **x = 10**  
**<>** ak nerovná sa, rôzne hodnoty, napr. **x <> 10**  
**<=** ak je menší alebo rovný ako, napr. **x <= 10**  
**<** ak je menší ako, napr. **i < 100**

*Úloha 3.8.1:* Grafická plocha má súradnice (0, 0, 700, 600). Našou úlohou je vykresliť maličké štvorčeky ( $a = 3$ ) v troch rozdielnych farbách, ktoré budú mať náhodné pozície, ale budú zaplňať len časť grafickej plochy. Plocha bude rozdelená na tri časti, pričom červené štvorčeky budú zaberat' prvú časť, druhú časť žlté štvorčeky a nakoniec modré štvorčeky.

*Riešenie:*

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var
```

```
x,y,i,a:Integer;
```

```
begin
```

```
Image1.Canvas.MoveTo(0,0);
```

```
a:=3;
```

```
for i:= 1 to 1000 do
```

```
begin
```

```
y:=random(600);
```

```
x:=random(700);
```

```
If (x<230) and (x>0) then Image1.Canvas.Brush.Color:=clred
```

```
else Image1.Canvas.Brush.Color:=clyellow;
```

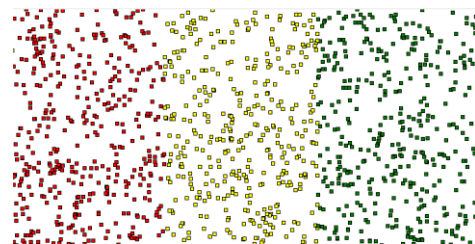
```
If (x>460) and (x<700)then Image1.Canvas.Brush.Color:=clgreen;
```

```
Image1.Canvas.Rectangle(x+a,y+a,x-a,y-a);
```

```
end;
```

```
end;
```

```
end.
```

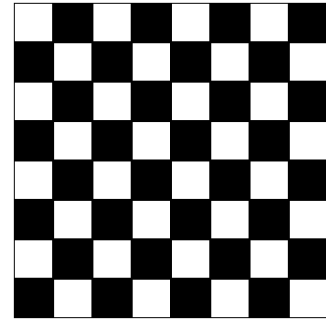


Úloha 3.8.2: Nakreslite šachovnicu 8x8 štvorčekov, veľkosť jedného polička  $a = 40$ .

Riešenie:

```

procedure TForm1.Button1Click(Sender: TObject);
var
  i,j,a:integer;
begin
  a:=40;
  for i:=0 to 7 do
    for j:=0 to 7 do
      begin
        if (i+j) mod 2 = 0 then Image1.Canvas.Brush.Color:=clWhite
        else
          Image1.Canvas.Brush.Color:=clblack;
          Image1.Canvas.Rectangle(400-a*i,400-a*j,400-a*(i+1),400-a*(j+1));
        end;
      end;
end.
  
```

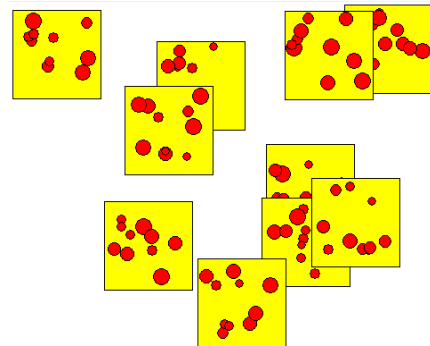


Úloha 3.8.3: Po každom stlačení tlačidla sa na náhodnom mieste grafickej plochy vykreslí štvorec žltej výplne a v ňom 10 kružníc, náhodnej veľkosti a červenej farby.

Riešenie:

```

procedure TForm1.Button1Click(Sender: TObject);
var
  a,b,x,y,i,c:integer;
begin
  randomize;
  a:=random(400);
  b:=random(400);
  Image1.canvas.Brush.color:=clyellow;
  Image1.Canvas.Rectangle(a,b,a+100,b+100);
  for i:= 1 to 10 do
    begin
      x:=a+random(80);
      y:=b+random(80);
      c:=random(10);
      image1.canvas.Brush.color:=clred;
      Image1.Canvas.ellipse(x,y,x+10+c,y+10+c);
    end;
  end;
end.
  
```



### 3.9 Cyklus s podmienkou

Okrem cyklu **for**, ktorý má presný počet opakovaní existuje ešte cyklus s neznámym počtom opakovaní - *cyklus s podmienkou* **while**. Cyklus začína podmienkou a príkazy cyklu sa vykonajú len dovtedy, kým je táto podmienka splnená. Hovoríme tomu podmienka ukončenia cyklu. Podmienka musí obsahovať aspoň jednu riadiacu premennú, ktorej hodnota sa automaticky zväčšuje pri každom prechode cyklu.

Hodnota premennej, ktorá je v podmienke sa musí v príkazoch cyklu meniť, inak by opakovanie cyklu nikdy neskončilo

**While cyklus:**

```
while podmienka do  
  begin  
    príkazy;  
  end;
```

alebo

```
premenná:=pociatocna_hodnota;  
while podmienka pre_premennu do  
  begin  
    príkazy;  
    zmena - premennej;  
  end;
```

Úloha 3.9.1: Pri každom stlačení tlačidla, máme vykresliť z náhodného miesta grafickej plochy, za sebou maximálny počet rovnakých kruhov, ale tak, aby sa posledný ešte zmestil do grafickej plochy. Koľko kruhov nakreslíme bude závisieť od veľkosti grafickej plochy. Kruh bude mať polomer  $r = 10$  a grafická plocha má rozmery 500x500. Úlohu naprogramujeme pomocou *while*:

Riešenie:

```
procedure TForm1.Button5Click(Sender: TObject);
```

```
var
```

```
x,y,r:Integer;
```

```
begin
```

```
r:=10;
```

```
x:=random(500);
```

```
y:=random(500);
```

```
Image1.Canvas.Brush.Color:=clYellow;
```

```
  while (x+r)<500 do
```

```
  begin
```

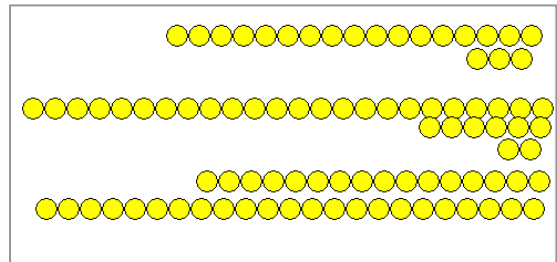
```
    image1.Canvas.Ellipse(x-r,y-r,x+r,y+r);
```

```
    x:=x+2*r;
```

```
  end;
```

```
end;
```

```
end.
```



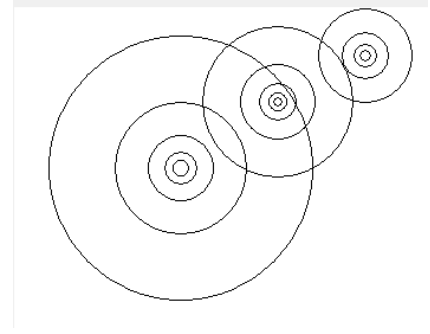
Úloha 3.9.2: Po každom stlačení tlačidla vykreslite na náhodnom mieste sústredné kružnice (max. 5), ale tak, aby posledná kružnica nepresiahla okraj grafickej plochy. Preto polomer kružníc závisí od náhodného miesta a okraja grafickej plochy.

Riešenie:

```
procedure TForm1.Button1Click(Sender: TObject);
```

```
var x,y,r:integer;
```

```
begin  
x:=random(image1.Width);  
y:=random(image1.Height);  
r:=random(9)+1;  
image1.Canvas.Brush.Style:=bsclear;  
  while ((x+r)< image1.Width) and ((y+r)< image1.Height) and ((x-r)> 0) and ((y-  
  r)>0) do  
    begin  
image1.Canvas.Ellipse(x-r,y-r,x+r,y+r);  
r:=2*r;  
    end;  
end;  
end.
```



## ZÁVER

Používanie počítača ako jedného z moderných prostriedkov IKT prináša so sebou postupnú zmenu tradičných metód a foriem vyučovacieho procesu, ktoré rozvíja samostatnú a tvorivú prácu žiakov. Tým sa vo vyučovaní mení tradičné postavenie učiteľa na úlohu radcu, konzultanta, ktorý je schopný a ochotný pomáhať žiakom pri hľadaní a spracovaní informácií. Správnym využívaním IKT v edukačnom procese sa stáva vzdelávací proces efektívnejší, pričom zohľadňuje rôzne štýly učenia sa a prispieva k rozvoju osobnosti žiaka.

IKT pomáhajú rozvíjať názornosť, praktickosť a zážitkovosť vyučovania žiakov, ale i učiteľov. Vnásajú do výučby nové námety, metódy a formy práce. Zároveň prinášajú širokú ponuku nástrojov k získavaniu nových poznatkov a vytvárajú prostredie, v ktorom prebieha efektívnejšie, kvalitnejšie a atraktívnejšie vzdelávanie.

Vyučovací predmet Informatika má v edukačnom procese významné postavenie, lebo podobne ako matematika podporuje myslenie žiakov, ich schopnosť rozoberať a spájať, zovšeobecňovať, dospieť k správnosti riešenia problémov a preveriť ich v praxi. Vyučovanie informatiky vedie žiakov k presnému vyjadrovaniu myšlienok, postupov a techník, ktoré slúžia ako všeobecný prostriedok komunikácie. S vyučovaním informatiky úzko súvisí výučba programovania. Základy znalosti programovania pomáhajú pochopiť, ako rôzne programy fungujú. Žiakom to umožní lepšie ovládať informačno-komunikačné technológie, napr. presnejšie formulovať svoje požiadavky na program.

## ZOZNAM BIBLIOGRAFICKÝCH ZDROJOV

1. Andrejková, G., Galčík, F., Guniš, J., Šnajder, L., 2012. Učebné štýly u študentov informatiky a podporné a učebné štýly. Didinfo, Banská Bystrica. 2012  
<[http://di.ics.upjs.sk/gunis/praca/prezentacie/didinfo\\_2012\\_andrejkovalgalcik\\_gunis\\_snajder\\_ucebne\\_a\\_vyucbove\\_styly.pdf](http://di.ics.upjs.sk/gunis/praca/prezentacie/didinfo_2012_andrejkovalgalcik_gunis_snajder_ucebne_a_vyucbove_styly.pdf)>
2. Bačová, D. Informatika a výpočtová technika. Delphi.  
<<http://www.spsepn.edu.sk/bc/index.php?stranka=programovanie&kap=delphi>>
3. Blaho, A., Kubincová, Z., Salanci, L., 2009. Ďalšie vzdelávanie učiteľov základných a stredných škôl v predmete informatika. Programovanie 2. Štátny pedagogický ústav, Bratislava. 2009. ISBN 978-80-8118-007-1
4. Blaho, A. 2006. Informatika pre stredné školy. Programovanie v Delphi. Slovenské pedagogické nakladateľstvo, Bratislava. 2006. ISBN 80-10-00421-9
5. Burgerová, J., 2001. Internet vo výučbe a štýly učenia. Prešov. 2001
6. Býna, M., 2009. Programujeme v Delphi,  
<<http://absolventi.gymcheb.cz/2010/mibyna/septima/>>
7. Fandelová, E., 1999. Psychológia rizika a zavádzanie multimediálnych prostriedkov vzdelávania. UKF, Nitra. 1999
8. Fialová, I., Študenti a počítač. In.: Technológia vzdelávania tretieho tisícročia MEDACTA 95. Zborník 2. Slovidiac, Nitra. 1995
9. Delphi - prednášky na FMFI UK, <<http://www.pascal.input.sk/>>
10. Homola, M., 1971. Úvod do psychológie osobnosti. SPN, Praha. 1971
11. Hrabal, V., Man, F., Pavelková, I. 1989. Psychologické otázky motivácie ve škole. SPN, Praha. 1989.
12. Hynek, J., 2002. Myslíme v jazyku Delphi 6, knihovna programátora. Grada Publishing a.s., Praha. 2002. ISBN 80-247-0334-3
13. Kabátová, M., Kubincová, Z., 2011. Kurz programovania pre študentov učiteľstva informatiky. Fakulta matematiky, fyziky a informatiky, Univerzita Komenského v Bratislave. Katedra základov a vyučovania informatiky,  
<<http://edi.fmph.uniba.sk/~kabatova/kurz/index.htm>>
14. Kadlec, Václav., 2001. Učíme se programovat v Delphi a jazyce Object Pascal. Vydavatelství a nakladatelství Computer Press, Praha. 2001. ISBN 80-7226-245-9
15. Kalaš, I., Vaníček, J., Mikolajová, K., Kabátová, M., Pekárová, J., 2010. Ďalšie vzdelávanie učiteľov základných a stredných škôl v predmete informatika. Štátny pedagogický ústav, Bratislava. 2010. ISBN 9788081180323
16. Kulič, V., 197. Chyba v učení. SPN, Praha. 1971.
17. Machová, J., 2004. Začínáme s Delphi. Metodicko-pedagogická centrum, Prešov. 2004. ISBN 80-8045-353-5
18. Machová, J., 2004. Delphi pre začiatočníkov.  
<<http://server.gphmi.sk/machova/delphi/>>
19. Mareš, J. 1998. Styly učení žaku a studentu. Portál, Praha. 1998

20. Nákonečný, M., 1972. Psychologie osobnosti. SPN, Praha. 1972
21. Novák, M. 2000. Úvod do používania Pc a informačných technológií. Metodické centrum, Prešov. 2000
22. Ondrušová, J. 1995. Motivačné činitele formujúce žiakovu osobnosť a kreatívne myslenie. MEDACTA 95. Zborník 1. Technológia vzdelávania tretieho tisícročia. Slovidac, Nitra. 1995.
23. Pardel, T., Boroš, J., 1979. Základy všeobecnej psychológie. SPN, Bratislava. 1979
24. Skolowsky, P., Šedivá, Z., 1994. Multimédia súčasnosť budúcnosť. Grada, Praha. 1994. ISBN 8071690813
25. Zelina, M., 1994. Stratégia a metódy rozvoja osobnosti dieťaťa. Iris, Bratislava. 1994. ISBN 8096701347