

## UČEBNÉ OSNOVY

Názov predmetu	PROGRAMOVANIE				
Časový rozsah výučby					
Ročník	1.	2.	3.	4.	Spolu
Štátny vzdelávací program					0
Školský vzdelávací program				2	2
<b>Kód a názov odboru štúdia</b>	7902 500 gymnázium				
<b>Stupeň vzdelania</b>	vyššie sekundárne vzdelanie ISCED 3A				
<b>Forma štúdia</b>	denná				
<b>Dĺžka štúdia</b>	štvorročná				
<b>Vyučovacia jazyk</b>	slovenský jazyk				

### Charakteristika predmetu

Algoritmizácia má dôležité postavenie vo vzdelávaní, pretože podobne ako matematika rozvíja myslenie žiakov, ich schopnosť analyzovať a syntetizovať, zovšeobecňovať, hľadať vhodné stratégie riešenia problémov a overovať ich v praxi. Vedie k presnému vyjadrovaniu myšlienok a postupov a ich zaznamenaniu vo formálnych zápisoch, ktoré slúžia ako všeobecný prostriedok komunikácie.

Obsahom povinne voliteľného predmetu: Programovanie vo štvrtom ročníku je učivo tematického celku Postupy, riešenie problémov, algoritmické myslenie v programovacom prostredí Lazarus (DELPHI).

### Ciele vyučovacieho predmetu

Cieľom vyučovania programovania je schopnosť algoritmizovať problém v danom programovacom jazyku a vedieť nájsť jeho optimálne riešenie s prihliadnutím na časovú a pamäťovú zložitosť.

### Základné predmetové kompetencie (spôsobilosti)

- rozvíjať si schopnosť algoritmizovať zadaný problém, rozvíjať si programátorské zručnosti;
- rozvíjať si svoje schopnosti kooperácie a komunikácie (naučiť sa spolupracovať v skupine pri riešení problému, zostaviť plán práce, špecifikovať podproblémy, distribuovať ich v skupine, vysvetliť problém ďalšiemu žiakovi, riešiť podproblémy, zhromaždiť výsledky, zostaviť ich do celkového riešenia, verejne so skupinou o ňom referovať);
- rozvíjať si svoju osobnosť, tvorivosť, logické myslenie, zodpovednosť, morálne a vôľové vlastnosti, húževnatosť, sebakritickosť a snažiť sa o sebazvedelávanie;
- naučiť sa rešpektovať intelektuálne vlastníctvo a autorstvo informatických produktov, systémov a aplikácií (aby chápali, že informácie, údaje a programy sú produkty intelektuálnej práce, sú predmetmi vlastníctva a majú hodnotu), pochopiť sociálne, etické a právne aspekty informatiky.

### Stratégia vyučovania

Pri vyučovaní sa budú využívať nasledovné metódy a formy vyučovania

	<b>Metódy</b>	<b>Formy práce</b>
<b>Postupy, riešenie problémov, algoritmické myslenie</b>	Slovné – vysvetľovanie, práca s textom Názorno – demonštračné – práca a pozorovanie, inštruktáž Praktické – experimentovanie, vytváranie zručností Riešenie problémov	Samostatná práca - programovanie Práca s literatúrou Vyučovanie prostredníctvom IKT

### Učebné zdroje

Na podporu a aktiváciu vyučovania a učenia žiakov sa využijú nasledovné učebné zdroje

	<b>Odborná literatúra</b>	<b>Didaktická technika</b>	<b>Ďalšie zdroje</b>
<b>Postupy, riešenie problémov, algoritmické myslenie</b>	Blaho , Kučera, Takáč: Informatika pre SŠ – Programovanie v Delphi	Počítače Dataprojektor Tabuľa	Internet Knížnica Digitálne učebné texty na serveri

### Spôsoby hodnotenia

V procese diagnostiky a hodnotenia žiakov uplatňujeme rozličné metódy i formy s cieľom poskytnúť žiakovi šancu dosiahnuť úspech. Žiak sa aktívne zapája do procesu hodnotenia.

Výsledná klasifikácia môže byť vyjadrená známku a percentami.

#### 1. Výsledná klasifikácia v predmete informatika zahŕňa nasledovné formy a metódy overovania požiadaviek na vedomosti a zručnosti žiakov:

- písomné – testy, previerky, referáty;
- praktické – projekty, domáce úlohy, praktické cvičenia;
- ústne – ústne prezentovanie osvojených poznatkov, pri ktorom sa kladie dôraz nielen na kvalitu osvojenia, ale aj na spôsob ich prezentácie v logických súvislostiach a ich aplikáciu v praktických súvislostiach.

#### 2. Vo výslednej klasifikácii sa odzrkadľuje:

- sumatívne (súhrnné) hodnotenie, ktoré sa odvíja od základného učiva definovaného v obsahovom a výkonovom štandarde; výsledná klasifikácia závisí od miery jeho zvládnutia;
- hodnotenie projektov a praktických cvičení, ktoré preverujú schopnosť žiaka uplatňovať získané vedomosti a zručnosti pri riešení konkrétnych úloh.  
Pri projektoch sa hodnotí:
  - odborná úroveň;
  - kvalita výstupu, grafická úroveň;
  - úroveň obhajoby;
  - využitie dostupných zdrojov – internet, odborná literatúra;
  - vypracovanie protokolu na požadovanej úrovni.
- reprezentácia školy na konferenciách a podobných akciách, účasť v súťažiach v rámci daného predmetu, tieto aktivity žiaka v predmete môžu výslednú klasifikáciu zlepšiť

Učebný plán predmetu: PROGRAMOVANIE – 4. ročník		2 hodiny týždenne, spolu 60 vyučovacích hodín ročne	
Názov tematického celku Témy	Hodiny	Očakávané vzdelávacie výstupy	Kritériá hodnotenia vzdelávacích výstupov
<b>Úvod</b>	<b>2</b>	<b>Žiak má:</b>	<b>Žiak:</b>
<ul style="list-style-type: none"> <li>Oboznámenie sa s plánom, bezpečnostné predpisy</li> </ul>	1.	Oboznámiť sa s bezpečnostnými predpismi a pravidlami práce v informatickej učebni,	Oboznámil sa s uvedenými predpismi a pravidlami, so štruktúrou siete, pozná svoje prístupové meno a heslo.
<ul style="list-style-type: none"> <li>Práca so školským serverom, dostupná literatúra</li> </ul>	2.	Vyhľadať cieľové požiadavky z informatiky, pracovať s učebnými textami a žiackymi prácami na školskej sieti	Vyhľadal a stiahol z internetu dokumenty súvisiace s maturitnou skúškou, oboznámil sa z podmienkami hodnotenia, na školskej sieti preštudoval dostupné elektronické materiály
<b>Postupy, riešenie problémov, algoritmické myslenie</b>	<b>6</b>	<b>Žiak má:</b>	<b>Žiak:</b>
<ul style="list-style-type: none"> <li>Algoritmizácia</li> </ul>	3.	Objasniť pojmy algoritmus, algoritmizácia, program, programovanie, vlastnosti algoritmov. Demonštrovať rôzne spôsoby zápisu algoritmov (slovný, vývojový diagram, program).	Vysvetlil základné pojmy algoritmizácie, vymenoval vlastnosti algoritmov, uviedol príklady rôzneho zápisu algoritmov
<ul style="list-style-type: none"> <li>Etapy riešenia problému, povolanie „informatik“</li> </ul>	4.	Rozdeliť životný cyklus programu do jednotlivých etáp, charakterizovať etapy, analyzovať prácu v programovom tíme	Popísal prácu pri vytváraní programového produktu (softvéru), charakterizoval jednotlivé etapy životného cyklu programu a analyzoval možné problémy v jednotlivých krokoch, charakterizoval pozície v programovom tíme
<ul style="list-style-type: none"> <li>Tvorba a modifikácia algoritmov</li> </ul>	5.-6.	Rozumieť hotovým algoritmom zapísaným neformálnym jazykom (napr. grafický zápis, prirodzený jazyk, jazyk blízky formálnemu jazyku, programovací jazyk; vie sa zorientovať aj v neštandardných popisoch algoritmov, s ktorými sa priamo nemusel stretnúť; Požadovaným spôsobom modifikovať hotové algoritmy(programy). Testovať výstupy a vyhľadávať chyby	rozumel hotovým algoritmom zapísaným neformálnym jazykom (napr. grafický zápis, prirodzený jazyk, jazyk blízky formálnemu jazyku, programovací jazyk; vedel sa zorientovať aj v neštandardných popisoch algoritmov, s ktorými sa priamo nemusel stretnúť Testoval a ladil hotové algoritmy (trasovanie, krokovanie), modifikoval ich podľa požiadaviek

<ul style="list-style-type: none"> <li>• Vývojové diagramy</li> </ul>	7.-8.	Definovať základné algoritmické štruktúry (postupnosť, vetvenie, cyklus). Rozumieť hotovým algoritmom, vysvetliť jednotlivé zápisy, orientovať sa aj v neštandardných popisoch algoritmov. Vytvárať algoritmus na riešenie zadaného problému, dodržiavať zásady čitateľného zápisu algoritmu	Definoval základné algoritmické štruktúry (postupnosť, vetvenie, cyklus). Vysvetlil hotový algoritmus (aj neštandardný), vytvoril algoritmus na riešenie zadaného problému
<b>Programovanie1</b>	<b>14</b>	<b>Žiak má:</b>	<b>Žiak:</b>
<ul style="list-style-type: none"> <li>• Programovacie jazyky, všeobecne</li> </ul>	9.	Popísať vývoj programovacích jazykov, definovať rozdiely pri programovaní zhora nadol (objektovo orientované) a zdola nahor (strojovo orientované). Demonštrovať rôzne programové prostredia. Vytvoriť aplikácie s použitím základných grafických komponentov na vytvorenie používateľského prostredia, napísať podprogram na obsluhu udalosti	Popísal vývoj programovacích jazykov, definoval rozdiely pri programovaní objektovo a strojovo orientovaných jazykov, ukázal zápis programu v rôznych prostrediach
<ul style="list-style-type: none"> <li>• Programovací jazyk – popis jazyka, chyby v programe</li> </ul>	10.	Vyhľadať, stiahnuť, nainštalovať a spustiť program (programovací jazyk), popísať pracovné prostredie, otvoriť a spustiť existujúci program. Popísať lexikálne prvky jazyka, vysvetliť rozdiel medzi syntaktickými a sémantickými chybami. Rozpoznať a odstrániť syntaktické chyby, identifikovať miesta programu, v ktorých môže dôjsť k chybám počas behu programu a vedieť ich opraviť. Vyhľadať informácie v pomocníkovi	Demonštroval prácu v pracovnom prostredí rôznych programových jazykov (imagine, pascal, lazarus), popísal lexikálne prvky jazyka, vysvetlil rozdiel medzi chybami, vyhľadal požadované informácie v pomocníkovi
<ul style="list-style-type: none"> <li>• Premenné</li> </ul>	11.-12.	Definovať typ a množinu operácie na danom type. Vysvetliť problematiku pretečenia typu. Demonštrovať načítanie a výpis premennej rôzneho typu (číslo, písmeno, text) Používať lokálne a globálne premenné, funkcie na generovanie náhodných hodnôt.	Definoval typ a množinu operácií na ňom. Vysvetlil pojem pretečenie typu. Demonštroval načítanie a výpis premennej rôzneho typu (číslo, písmeno, text) Definoval rozdiel v používaní lokálnych a globálnych premenných Použil generovanie náhodných čísiel v programe.
<ul style="list-style-type: none"> <li>• Procedúry a funkcie</li> </ul>	13.-14.	Používať preddefinované procedúry a funkcie, vysvetliť vstupné parametre Definovať vlastné procedúry a funkcie bez parametrov aj s parametrami	Vysvetlil preddefinované procedúry a funkcie, definoval vlastné procedúry a funkcie
<ul style="list-style-type: none"> <li>• Vetvenie</li> </ul>	15.-18.	Používať a vysvetliť vetvenie úplne, neúplné, vnorené a viacnásobné vetvenie. Vedieť používať logické operátory (and, or, not) a vytvárať logické výrazy.	Vysvetlil a použil vetvenie v programe
<ul style="list-style-type: none"> <li>• Cyklus</li> </ul>	19.-22.	Vytvoriť algoritmus, v ktorom sa používa cyklus s vopred známym počtom opakovaní, s podmienkou na začiatku a na konci. Objasniť problematiku ukončenia cyklu. Používať vnorený cyklus.	Vysvetlil a použil cyklus v programe

<b>Programovanie2</b>	<b>6</b>	<b>Žiak má:</b>	<b>Žiak:</b>
<ul style="list-style-type: none"> <li>Čísla</li> </ul>	23.-24.	Vedieť používať funkcie na generovanie náhodných hodnôt. Vysvetliť obmedzenia vyplývajúce z typu dát a zvoliť vhodné typy dát pri riešení problému.	Používal vhodné typy dát pri riešení problému, vysvetlil obmedzenia vyplývajúce z typu dát, Používal funkcie pre prácu s číslami a generovanie náhodných čísel
<ul style="list-style-type: none"> <li>Reťazce</li> </ul>	25.-26.	Používať procedúry a funkcie, na prácu s reťazcami v programovacom jazyku, ktorý sa učia (napr. dĺžka reťazca, hľadanie pozície znaku alebo podreťazca, vrátenie podreťazca, zmena hodnoty na zadanej pozícii, vloženie a vymazanie podreťazca alebo práca s poľom ako s reťazcom znakov). Používať funkcie a procedúry na prevody medzi reťazcami a číslami.	Aplikoval poznatky o práci s reťazcami na riešenie úloh, vysvetlil parametre príkazov
<ul style="list-style-type: none"> <li>Jednorozmerné pole</li> </ul>	27.-28.	Pracovať s lineárnou štruktúrou napr. s jednorozmerným poľom, riešiť úlohy typu: vyhľadávanie v utriedenej a neutriedenej lineárnej štruktúre (zistenie pozície); posun prvkov štruktúry; hľadanie minima, maxima (aj pozície); výmena prvkov, výber podmnožiny podľa zadaných kritérií, vsunutie - odstránenie prvku podľa zadaných kritérií.	Definoval jednorozmerné pole, efektívne použil cyklus a indexovanú premennú na prístup k jednotlivým prvkom lineárnej štruktúry,
<b>Programovanie3</b>	<b>12</b>	<b>Žiak má:</b>	<b>Žiak:</b>
<ul style="list-style-type: none"> <li>Dvojrozmerné pole</li> </ul>	29.-32.	Vedieť pracovať s dvojrozmerným poľom (napr. do dvojrozmerného poľa kódovať vlastné obrázky ako čísla farieb v štvorcovej sieti, alebo zapísať si vlastný rozvrh do tabuľky, pexeso, ...)	Definoval dvojrozmerné pole, použil vnorený cyklus na prístup k jednotlivým prvkom dvojrozmerného poľa
<ul style="list-style-type: none"> <li>Textový súbor</li> </ul>	33.-36.	Pracovať s textovým súborom - čítať zo súboru, zapisovať do súboru, pridávať do súboru, navrhnuť efektívny a ľahko čitateľný formát zápisu, pracovať s viacerými súbormi naraz (napr. z jedného súboru vytvorte kópiu tak, aby v kópii boli očíslované riadky; v textovom súbore zadané slovo nahraďte zadaným textom).	Aplikoval poznatky o práci so súbormi na riešenie úloh
<ul style="list-style-type: none"> <li>Lineárna štruktúra záznamov</li> </ul>	37.-40.	Vedieť pracovať so záznamom a s lineárnou štruktúrou záznamov (napr. súradnice, smer, farba a popis nejakého útvaru).	Definoval vlastný záznam, použil prvky záznamu v algoritme
<b>Modifikácia programov</b>	<b>20</b>	<b>Žiak má:</b>	<b>Žiak:</b>
<ul style="list-style-type: none"> <li>Modifikácia programov</li> </ul>	41.-60.	Požadovaným spôsobom modifikovať hotové programy, určiť vlastnosti vstupov, výstupov a vzťahy medzi nimi, vedieť ich testovať. Definovať vlastné procedúry a funkcie bez parametrov aj s parametrami, identifikovať chyby v programe	Modifikoval hotové programy, definoval vlastné procedúry a funkcie bez parametrov aj s parametrami, opravil chyby v programe