

**CIEĽOVÉ POŽIADAVKY  
NA VEDOMOSTI A ZRUČNOSTI MATURANTOV  
Z INFORMATIKY**

BRATISLAVA 2019

Schválilo Ministerstvo školstva, vedy, výskum a športu Slovenskej republiky  
dňa 12. júna 2019 pod číslom 2019/2049:2-A1020 s platnosťou od 1. 9. 2019

## ÚVOD

Cieľom maturitnej skúšky z informatiky je zistiť u žiakov najmä úroveň:

- schopností riešiť algoritmické problémy,
- schopností zdokumentovať a prezentovať navrhnuté riešenia,
- vedomostí a pochopenia teoretických poznatkov z informatiky.

Podľa ŠVP je vzdelávací (obsahový a výkonový) štandard rozdelený do 5 okruhov:

1. Algoritmické riešenie problémov
2. Reprezentácie a nástroje
3. Softvér a hardvér
4. Komunikácia a spolupráca
5. Informačná spoločnosť

Do maturitnej skúšky sa tieto okruhy premietnu nasledujúcim spôsobom. Každé zadanie na maturitnej skúške je zložené z dvoch úloh, ktoré môžu, ale nemusia na seba nadväzovať:

### **1. Programovanie v konkrétnom programovacom jazyku**

- Algoritmické riešenie problémov

### **2. Základy informatiky:**

- Algoritmické riešenie problémov
- Reprezentácie a nástroje
- Softvér a hardvér
- Komunikácia a spolupráca
- Informačná spoločnosť

## **PODROBNOSTI O SPÔSOBE KONANIA ÚSTNEJ FORMY INTERNEJ ČASTI MATURITNEJ SKÚŠKY**

Každé maturitné zadanie sa skladá z dvoch úloh (úlohy môžu, ale nemusia na seba nadväzovať).

V maturitných zadaniach musia byť zastúpené všetky tematické celky z cieľových požiadaviek.

### **Charakteristika úloh maturitných zadaní**

Žiak na maturitnej skúške nepreukazuje zručnosti v ovládaní digitálnych technológií, ale schopnosti riešiť algoritmické problémy a úroveň vedomostí zo základov informatiky (žiak by mal vysvetliť ideu, princíp fungovania).

#### ***Úloha č. 1 – Riešenie algoritmického problému v konkrétnom programovacom jazyku***

Žiak má vyriešiť zadaný algoritmický problém. V úlohe je definovaný cieľ, ktorý má žiak dosiahnuť, ale nie sú uvedené prostriedky jazyka, pomocou ktorých má dosiahnuť cieľ. Výber prostriedkov je súčasťou hodnotenia. Žiaci majú k dispozícii počítač a vývojové prostredie programacieho jazyka, v ktorom prebiehalo vyučovanie. Odporúčané jazyky sú Pascal alebo Python (prípadne aj C++, Java, C#, a pod.).

**Úloha č. 2 – Riešenie problému z oblasti Základy informatiky** (algoritmické riešenie problémov, reprezentácie a nástroje, softvér a hardvér, komunikácia a spolupráca, informačná spoločnosť)

Úloha má byť prierezová tak, aby zasahovala aspoň do troch oblastí. Väčšinou pozostáva z riešenia problému, ktorý má algoritmický charakter, pomocou rôznych nástrojov informatiky (žiak nemusí použiť programovanie).

### Všeobecné pomôcky

Počítač s možnosťou pripojenia na internet a so softvérovým vybavením, v ktorom prebiehalo vyučovanie.

### Hodnotenie

- a) Každá úloha maturitného zadania sa hodnotí stupňom prospechu 1 až 5.  
 b) Váha hodnotenia jednotlivých úloh je 7 : 3. Pri výpočte váženého priemeru sa používa vzorec

$$z = \frac{7 \cdot z_1 + 3 \cdot z_2}{10},$$

pričom  $z$  je po zaokrúhlení výsledný stupeň prospechu a  $z_i$  je stupeň prospechu za úlohu č.  $i$ .

## 1 ALGORITMICKÉ RIEŠENIE PROBLÉMOV

### 1.1 Typy údajov, premenné a výrazy

#### Obsah

*Pojmy:*

- číslo, znak, logická hodnota, pole, textový reťazec, textový súbor,
- aritmetické operácie: +, -, \*, /, zvyšok po delení,
- operácie porovnania: =, ≠, <, > a pod.,
- logické operácie: negácia, a, alebo,
- konštanta, premenná, štandardná funkcia, generátor náhodných čísel.

*Vlastnosti a vzťahy:*

- typy údajov – ich použitie a obmedzenia,
- pravidlá vyhodnocovania výrazov – priorita operácií, výrazy so štandardnými funkciami,
- premenná – názov, hodnota, vzťah s pamäťou,
- jednorozmerné polia – prvky, indexy, usporiadané hodnoty,
- textový súbor – ako postupnosť znakov a riadkov, koniec súboru,
- zložený a jednoduchý typ údajov.

## Požiadavky na vedomosti a zručnosti

Žiak vie:

- vyhodnotiť a zostaviť výraz s operáciami, premennými, zátvorkami,
- použiť generátor náhodných čísel,
- realizovať prevody – číslo na textový reťazec, textový reťazec na číslo, do/z číselnej sústavy,
- interpretovať výrazy a postupnosti priradovacích príkazov (číselné aj logické),
- identifikovať rozdiely pri práci s celými a desatinnými číslami,
- pristupovať k prvkom polí,
- manipulovať so znakmi textového reťazca, manipulovať s podreťazcom textového reťazca,
- manipulovať s obsahom textového súboru – otvoriť, čítať a zapisovať údaje, zatvoriť, reagovať na koniec súboru.

## 1.2 Programové konštrukcie

### Obsah

*Pojmy:*

- príkaz, postupnosť príkazov, program, programovací jazyk, vývojové prostredie,
- programové konštrukcie – cyklus, vetvenie, podprogram.

*Vlastnosti a vzťahy:*

- syntax jazyka – rezervované slová, identifikátory, konštanty, oddeľovače,
- prostriedky jazyka pre interakciu – získanie vstupu, zrealizovanie výstupu,
- cyklus s pevným počtom opakovaní, cyklus s podmienkou – počítadlo cyklu, telo cyklu, vnorené príkazy,
- vetvenie – podmienený príkaz, podmienka, telo vetvy, vnorené príkazy,
- podprogram – definovanie, hodnotové parametre, nerekurzívne volanie, návratová hodnota, lokálne premenné.

## Požiadavky na vedomosti a zručnosti

Žiak vie:

- sformulovať formálny zápis algoritmu podľa kritérií rôznych vykonávateľov,
- zostaviť príkazy do príkazových konštrukcií podľa pravidiel jazyka,
- rozpoznať a odstrániť chyby vo formálnom zápise algoritmu.

## 1.3 Riešenie problémov

### Obsah

#### *Pojmy:*

- algoritmus,
- interpretácia a kompilácia.

#### *Vlastnosti a vzťahy:*

- chyby: syntaktické, logické, chyby vzniknuté za behu,
- analýza problému – vzťahy medzi vstupom a výstupom,
- vstupné obmedzenia – pre ktoré hodnoty algoritmus správne pracuje.

### Požiadavky na vedomosti a zručnosti

#### Žiak vie:

- analyzovať problém – identifikovať vstup, výstup, vzťahy medzi vstupom a výstupom,
- riešiť algoritmické problémy, vytvárať a čítať zápisy v programovacom jazyku,
- riešiť úlohy, v ktorých sa pracuje s číslami a ciframi čísla:
  - zistiť, či číslo spĺňa zadané kritérium,
  - realizovať výpočty s desatinnými číslami,
  - previesť čísla medzi číselnými sústavami,
- riešiť úlohy, v ktorých sa využíva interakcia:
  - získať vstup (napríklad: čísla, textové reťazce) zo vstupného zariadenia,
  - spracovať vstup, ak je počet vstupných hodnôt dopredu neznámy,
  - vypísať textový a nakresliť grafický výstup,
- riešiť problémy, v ktorých sa využívajú znaky a textové reťazce:
  - zistiť, či je znak písmeno alebo cifra,
  - zistiť výskyt znaku alebo podreťazca v textovom reťazci podľa daného kritéria: prvý, posledný, počet výskytov a pod.,
  - nahradiť alebo odstrániť znak alebo podreťazec v textovom reťazci,
  - zostaviť textový reťazec z podreťazcov podľa daných kritérií (napríklad: striedajú sa čísla a texty a pod.),
  - formátovať výpis,
- riešiť problémy, v ktorých sa využívajú premenné:
  - vymeniť hodnoty dvoch premenných,
- riešiť problémy, v ktorých sa využíva cyklus a vetvenie:
  - zostaviť podmienky pre vetvenie a cyklus podľa zadania úlohy (napríklad: či je hodnota z daného intervalu, či sa bod nachádza v obdĺžnikovej oblasti, či bod leží v kruhu a pod.),
  - prechádzať prvky postupnosti (napríklad: generovanej, zo vstupu, znaky textového reťazca, riadky súboru a pod.),
  - akumulovať výsledky podľa daných kritérií (napríklad: súčet postupnosti čísel a pod.),

- vyberať hodnoty, ktoré spĺňajú dané kritériá (napríklad: z textového reťazca vypísať len všetky spoluhlásky a pod.),
- používať vnorené programové konštrukcie (napríklad: výpis tabuľky hodnôt, spracovanie súboru s riadkami s viacerými hodnotami a pod.),
- rozhodovať sa, kedy stačí použiť cyklus s pevným počtom opakovaní, a kedy treba cyklus s podmienkou,
- riešiť problémy, v ktorých sa využíva generátor náhodných čísel:
  - generovať čísla v danom rozsahu,
  - simulovať danú činnosť (napríklad: zostavovať frekvenčnú tabuľku pri hode dvoma kockami a pod.),
  - generovať náhodnú kresbu podľa stanovených kritérií (napríklad: náhodne rozmiestnené geometrické tvary, kreslenie obdĺžnika z veľkého počtu náhodne zafarbených úsečiek a pod.),
- riešiť problémy, v ktorých sa využívajú podprogramy:
  - definovať vlastné podprogramy s parametrami,
  - definovať vlastné podprogramy s návratovou hodnotou rôznych typov (čísla, texty, logické hodnoty a pod.),
  - používať parametre rôznych typov (čísla, texty a pod.),
  - používať vlastné aj dané podprogramy pri riešení úlohy,
- riešiť problémy, v ktorých sa využíva jednorozmerné pole:
  - generovať obsah poľa podľa daných kritérií (napríklad: vynulovanie poľa, každý nasledujúci prvok je súčtom predchádzajúcich dvoch a pod.),
  - používať pole na uchovanie väčšieho počtu údajov (napríklad: frekvenčná tabuľka výskytu nejakých hodnôt, súradnice bodov v rovine a pod.),
  - hľadať prvky poľa s danými vlastnosťami (napríklad: najmenší, druhý najmenší a pod.),
  - zisťovať, či pole obsahuje hodnoty s danými vlastnosťami (napríklad: či obsahuje číslo 0, prvý výskyt medzery a pod.), resp. ich počet,
  - zisťovať, či pole spĺňa dané kritérium (napríklad: či sú všetky prvky rovnaké, rôzne, či sú hodnoty usporiadané a pod.),
  - modifikovať prvky poľa (napríklad: vsunúť prvok na dané miesto tak, aby ostali prvky usporiadané a pod. – pozor, už nie triediace algoritmy),
  - manipulovať s viacerými poľami podľa daných kritérií (napríklad: kopírovanie časti poľa, otočenie, posunutie, zlučovanie dvoch usporiadaných postupností a pod.),
  - testovať hranice poľa,
  - zobrazovať prvky poľa vypísaním alebo vykreslením (napríklad: stĺpcový graf),
- riešiť problémy, v ktorých sa využíva textový súbor:
  - zobrazíť obsah súboru (napríklad: kreslenie zakódovanej obrázky, naplnenie poľa hodnotami a pod.),
  - používať textový súbor ako vstup alebo výstup (napríklad: čítanie vstupných údajov pre výpočty; zapisovanie výsledkov výpočtov, obsahu polí a pod.),
  - zisťovať štatistické údaje o obsahu textového súboru (napríklad: počet riadkov súboru, počet slov, počet znakov, súčet čísel a pod.),
  - pracovať s viacerými textovými súbormi (napríklad: kopírovanie súborov, zlúčenie

- dvoch súborov, kopírovanie s filtrovaním a pod.),
- opravovať syntaktické chyby,
  - zisťovať, hľadať a opravovať logické chyby v programe:
    - krokovat',
    - používať kontrolné výpisy (napríklad: vložiť kontrolný výpis na správne miesto v programe),
    - odhaľovať kritické miesta v programe (napríklad: v akých situáciách by program spadol, nemusel správne počítať, fungovať),
    - navrhovať vhodné testovacie vstupné údaje,
    - testovať cudzie programy (napríklad: nájsť a opraviť chybu v cudzom programe),
  - intuitívne uvažovať o zložitosti algoritmu (na úlohách primeranej zložitosti):
    - odhadovať počet krokov, ktoré zrealizuje daný algoritmus (napríklad: koľko hviezdíčiek program nakreslí pre dané číslo a pod.),
    - zistiť, pre aký vstup sa zrealizuje daný počet krokov daného algoritmu (napríklad: pre aké číslo sa nakreslí daný počet hviezdíčiek a pod.).

## 2 REPREZENTÁCIE A NÁSTROJE

### Obsah

#### Pojmy:

- bit (b), bajt (B).

#### Vlastnosti a vzťahy:

- $1\text{B} = 8\text{ b}$ ,  $\text{kB} = 1000\text{ B}$ , rýchlosť prenosu informácie b/s,
- kódovanie informácie: logické výrazy, čísla, znaky, textové reťazce, grafická informácia (rastrová, vektorová, farby), zvuk.

### Požiadavky na vedomosti a zručnosti

#### Žiak vie:

- posúdiť možnosti kódovania informácií rôznych typov,
- správne zvoliť reprezentáciu pri riešení algoritmickej úlohy, zdôvodniť voľbu reprezentácie reálnej informácie,
- sformulovať interpretáciu informácie uloženej v údajoch a štruktúrach,
- zakódovať informáciu do konkrétnej digitálnej reprezentácie (napríklad: obrázok ako dvojrozmerná štruktúra, obrázok ako návod na jeho kreslenie, súradnice ako postupnosť čísel alebo dvojíc, čísla v dvojkovej sústave a pod.),
- dekódovať informáciu z jej digitálnej reprezentácie,
- zvoliť si vhodný spôsob výstupu (číselný, textový, grafický),
- opísať, ako sú rôzne typy údajov uložené v počítači,
- ilustrovať vzťah medzi dvojkovou a šestnástkovou reprezentáciou.

### 3 SOFTVÉR A HARDVÉR

#### Obsah

##### *Pojmy:*

- hardvér – procesor, pamäť, zariadenia,
- softvér – operačný systém, ovládače zariadení, procesy.

##### *Vlastnosti a vzťahy:*

- idea fungovania von Neumannovej architektúry počítača,
- procesor – ako vykonávateľ s dobre definovanými inštrukciami,
- úloha operačného systému – ako program pre správu procesov, súborov, pamäte a zariadení.

#### Požiadavky na vedomosti a zručnosti

##### Žiak vie:

- používať prístupové práva, komprimovať a dekomprimovať údaje,
- identifikovať rôzne zariadenia, ktoré obsahujú procesor,
- vysvetliť úroveň hardvéru a softvéru, ktoré zabezpečujú vykonávanie programu,
- získať informácie o súboroch, priečinkoch a pamäťových zariadeniach,
- zistiť parametre bežiackej aplikácie a procesov v systéme; zrušiť ich; riešiť kritické situácie bežiacich aplikácií.

### 4 KOMUNIKÁCIA A SPOLUPRÁCA

#### Obsah

##### *Vlastnosti, vzťahy:*

- počítačová sieť, internet,
- vyhľadávanie, služby internetu, webové aplikácie,
- princíp komunikácie v počítačovej sieti/internete,
- zdieľanie dokumentov, bezpečnosť na internete.

#### Požiadavky na vedomosti a zručnosti

##### Žiak vie:

- porovnať rôzne prístupy k bezpečnému zdieľaniu informácií,
- porovnať rôzne spôsoby vyhľadávania a získavania informácií na internete (napríklad: text, obrázky, mapy, trasy, preklady, prevody jednotiek a pod.),
- porovnať princípy komunikácie prostredníctvom rôznych nástrojov (napríklad: interaktívna/neinteraktívna, výhody/nevýhody, bezpečnosť a pod.),
- navrhnúť postup pri kolaboratívnom riešení úloh.



## 5 INFORMAČNÁ SPOLOČNOSŤ

### Obsah

*Vlastnosti, vzťahy:*

- softvér zadarmo, softvér za poplatok,
- škodlivý softvér, škodlivé stránky, počítačové pirátstvo,
- anonymita na webe, ochrana súkromia.

### Požiadavky na vedomosti a zručnosti

Žiak vie:

- diskutovať o využití nástrojov v rôznych situáciách (napríklad: učenie sa, v práci, pri zábave a pod.),
- diskutovať o vplyvoch informatiky na spoločnosť,
- diskutovať o bezpečnosti (napríklad: údaje, komunikácia a pod.),
- diskutovať o počítačovom pirátstve a kriminalite.