04 AKO KÓDUJE POČÍTAČ – GRAFICKÁ INFORMÁCIA

Tematický celok / Téma	*Stupeň školy / Odporúčaný ročník / Rozsah*
Reprezentácie a nástroje - informácie	SŠ / 1. ročník / 1 vyučovacia hodina

Požiadavky na vstupné vedomosti a zručnosti

- Informácia, kódovanie informácie
- Prevod čísel do postupnosti bitov (dvojkovej sústavy) a späť

Ciele

Žiakom osvojované vedomosti a zručnosti	*Žiakom rozvíjané spôsobilosti*
Reprezentácie a nástroje – informácie **Výkonový štandard** • posudzovať rôzne reprezentácie pri spracovaní informácií, • kódovať informáciu do konkrétnej digitálnej reprezentácie, • dekódovať informáciu do konkrétnej digitálnej reprezentácie • argumentovať pre voľbu nástrojov. **Obsahový štandard** Pojmy: bit, bajt, kibibajt, mebibajt (resp. bit, bajt, kilobajt, megabajt), digitalizácia Vlastnosti a vzťahy: vzťahy medzi jednotlivými typmi informácií (grafika, čísla) Procesy: kódovanie grafickej informácie	**Informatické myslenie:** Logika • (LOG4) vyvodzovať (logicky zdôvodňovať) závery z pozorovaní a experimentov (aj myšlienkových) • (LOG6) logicky zdôvodniť zmenu algoritmu/programu • (LOG 8) z existujúcich pravidiel logicky odvádzať iné pravidlá Algoritmy • (ALG3) vytvárať vlastné algoritmy riešiace problém (návrh vlastného kódovania textu) Abstrakcia • (ABS3) využiť podstatné prvky problémov (vytvárať model, vyjadriť pomocou tabuľky)

Riešený didaktický problém

Aby si žiaci uvedomili možnosti, ale aj limity či perspektívy súčasných technológií, potrebujú poznať princípy ich fungovania. Je dôležité zvoliť vhodný spôsob reprezentácie grafickej informácie vzhľadom na podmienky použitia.

Dominantné vyučovacie metódy a formy	*Príprava učiteľa a pomôcky*
• Bádateľská metóda (model 5E), • frontálna a individuálna forma.	pre učiteľa • **I_SS_57_Kody_sifry_kompresia_M.pdf** metodika vyučovania vo formáte pdf • **I_SS_57_Kody_sifry_kompresia_PL_riesenie.docx** • **I_SS_57_Kody_sifry_kompresia_PL.docx** pracovný list a pracovný list s riešenými úlohami • **mesto.jpg** pracovný súbor (ukážka) pre žiaka • **I_SS_57_Kody_sifry_kompresia_PL.pdf** pracovný list Použitie digitálnych nástrojov: NUTNÉ

Diagnostika splnenia vzdelávacích cieľov

Sebahodnotiaci test v pracovnom zošite.

EURÓPSKA ÚNIA
Európsky sociálny fond
Európsky fond regionálneho rozvoja
OPERAČNÝ PROGRAM ĽUDSKÉ ZDROJE
MINISTERSTVO ŠKOLSTVA, VEDY, VÝSKUMU A ŠPORTU SLOVENSKEJ REPUBLIKY
it akadémia
Tento projekt sa realizuje vďaka podpore z Európskeho sociálneho fondu a Európskeho fondu regionálneho rozvoja v rámci Operačného programu Ľudské zdroje
www.minedu.sk www.employment.gov.sk/sk/esf/ www.itakademia.sk

ÚVOD

Štvrtá metodiky v sérii Kódy, šifry, kompresia nadväzuje na predchádzajúce. Jej cieľom je ukázať žiakom, akým spôsobom počítač pracuje s grafickou informáciou.

Zameriame sa najmä na bitmapovú (rastrovú) grafiku, keďže nám umožňuje nadviazať na vedomosti a skúsenosti žiakov z predchádzajúcich vyučovacích hodín realizovaných pomocou metodík tejto série. V závere aspoň v krátkosti spomenieme aj vektorovú grafiku a porovnáme ju s bitmapovou.

Keďže pri kódovaní grafickej informácie je potrebné opätovne pracovať s binárnou sústavou, v rámci tejto metodiky si žiaci precvičia zručnosti získané v predchádzajúcej téme – prevody medzi dekadickou a binárnou sústavou. Zároveň si precvičia aj prevody medzi jednotkami informácií (pri výpočtoch veľkostí niektorých grafických súborov).

Táto jedna vyučovacia hodina nepokrýva celú problematiku spracovania grafickej informácie – venuje sa len princípu reprezentácie rastrovej grafickej informácie. Odporúčame venovať nasledujúcu vyučovaciu hodinu grafickým formátom – ich charakteristike, výhodám a nevýhodám ich použitia a podobne. Vhodné je žiakom spomenúť – ak sme žiakov oboznámili s existenciou šestnástkovej sústavy – aj iný spôsob zápisu farieb, a to pomocou práve hexadecimálnej číselnej sústavy. S touto podobou zápisu sa stretnú napr. pri tvorbe webových stránok či pri práci v grafickom editore.

Žiaci majú k dispozícii pracovný list, ktorý obsahuje zadania úloh, miesto na žiacke riešenie a miesto pre poznámky. Odporúčame, aby učiteľ žiakom pri každej fáze vyučovania uviedol zoznam úloh z pracovného listu, ktoré budú aktuálne riešiť. Poslednou časťou je sebahodnotiaci test.

Poznámka:

Pracovný list je jedným z výstupov žiaka. Odporúčame, aby si žiaci jednotlivé vypracované pracovné listy odkladali. Neskôr ich môžu využiť pri opakovaní učiva.

PRIEBEH VÝUČBY

Osnova vyučovacej hodiny (podľa modelu 5E):

- **Zapojenie (5 minút)** – diskusia so žiakmi na tému kódovania textu
- **Skúmanie (8 minút)** – skúmanie princípov zaznamenávania grafickej informácie (úlohy 1 a 2 z pracovného listu)
- **Vysvetlenie (10 minút)** – vysvetlenie predchádzajúcich zistení, riešenie úloh (úlohy 3 až 5 z pracovného listu)
- **Rozpracovanie (12 minút)** – riešenie náročnejších úloh (úlohy 6 až 9 z pracovného listu)
- **Vyhodnotenie (5 minúty)** – vyriešenie sebahodnotiaceho testu, kontrola odpovedí

ZAPOJENIE (CCA 5 MIN)

Na predchádzajúcej vyučovacej hodine sme si ukázali, akým spôsobom je v počítači reprezentovaná textová informácia – použitá znaková sada jednoznačne priradí každému znaku jeho číselnú reprezentáciu (túto hodnotu nazývame aj ordinálna hodnota znaku), a tá je v podobe binárneho kódu spracovaná počítačom.

Akým spôsobom „preložiť“ do binárneho kódu grafickú informáciu? Či už je to fotografia, vlastná kresba vytvorená v grafickom editore či naskenovaná kresba. Už sme niekoľko krát zdôraznili, že počítač „len“ počíta. Tak, ako nevie čítať a písať (ale musíme ho to pomocou znakových sád a príslušných algoritmov naučiť), nevidí. Nemôžeme mu „ukázať“ fotografiu a spoliehať sa, že si ju „nejako“ zapamätá. Opäť mu musíme poskytnúť návod, ktorý ho naučí našu grafickú informáciu previesť do reči binárneho kódu (a naspäť).

Otázka 1	*Pri kódovaní textovej informácie sme využili to, že dokážeme text rozdeliť na jednotlivé znaky – a tým priradíme jednoznačný binárny kód. Nech už ktokoľvek napíše akýkoľvek nový text, román, životopis, správu..., vďaka univerzálne použiteľnej znakovej sade vieme tento text reprezentovať v podobe núl a jednotiek.* *Na aké elementárne (už viac nedeliteľné) prvky vieme „rozobrať“ grafickú informáciu? Ak to dokážeme, stačí týmto prvkov priradiť jedinečný binárny kód, a problém sme vyriešili.*

V reálnom svete odpoveď na uvedenú otázku nehľadáme. Obrazové vnemy vnímame ako celky. Ak sa budeme približovať k nejakému predmetu, uvidíme stále viac detailov, ale stále ten predmet budeme vnímať ako celok. Skúsme sa však takto približovať k otvorenému grafickému súboru v grafickom editore s použitím nástroja lupa. Uvažujme, čo sa bude pri tomto približovaní diať.

Učiteľ otvorí v grafickom editore Skicár grafický súbor (napr. pripravený súbor *mesto.jpg*) a postupne ho zväčšuje. Žiaci si značia svoje postrehy do pracovného listu. Keď už obrázok nie je možné zväčšiť, zosumarizujeme postrehy žiakov.

Možné postrehy žiakov môžu byť:

- obrázok sa bude zväčšovať,
- ak bol vytvorený vo vysokej kvalite (vysokom rozlíšení), budeme môcť vidieť aj jeho detaily,
- pri istom zväčšení sa obrázok začne „rozmazávať“,
- pri maximálnom zväčšení sa obrázok „rozloží“ na štvorčeky,
- každý štvorček je charakterizovaný svojou farbou (teda neobsahuje rôznofarebné plochy, jeho farbu vieme pomenovať – červený, modrý, svetlozelený)...

Počítač má limity, rovnako aj vstupné a výstupné zariadenia, pomocou ktorých s nami komunikuje. Či už je to jeho obmedzenú pamäť, konečný počet zobrazovacích bodov monitora apod. Z toho dôvodu nie je možné donekonečna „vstupovať“ do obrázku.

V závere diskusie by sme mali dospieť k týmto bodov:

- hľadaným elementárnym prvkom grafickej informácie je bod (pixel),
- charakteristikou každého bodu je jeho pozícia a jeho farba.

Ak by bol obrázok tvorený objektami, ktoré vieme jednoznačne opísať, mohli by sme pracovať s tzv. vektorovou grafikou. Vektorový obrázok je tvorený objektami, pričom každý je popísaný svojimi vlastnosťami. Napr. kruh je určený súradnicami svojho stredu, veľkosťou polomeru, farbou a štýlom obrysu, farbou a štýlom výplne, či má tieň, ... Jednotlivým charakteristikám vieme priradiť príslušný binárny kód a takto si obrázok zapamätať. Spätnou interpretáciou kódu počítač dokáže obrázok znova zobraziť (na správnej pozícii, správnej veľkosti či farby). Týmto spôsobom však nedokážeme reprezentovať každú grafickú informáciu – napríklad fotografiu.

Žiakom vysvetlíme, že na dnešnej vyučovacej hodine budeme skúmať tzv. bitmapovú (rastrovú) grafickú informáciu. V tomto prípade sa obrázok rozdelí na štvorcovú mriežku, kde jeden štvorec mriežky reprezentuje spomínaný jeden bod (pixel) obrázka, a tomu sa priradí charakteristická vlastnosť – farba bodu.

SKÚMANIE (CCA 13 MIN)

Úloha 1	*Neznámi hekeri získali jednej z ešte nezverejnených návrhov nového loga známej spoločnosti – dáta zverejnili na internete, pretože napriek svojim schopnostiam neoprávnene vstúpiť do komunikačnej siete nedokážu zistiť, čo sa za získanými dátami skrýva. Ponúkame vám získané dáta a k nim jednu dôležitú informáciu – vieme, že logo je dvojfarebné. Dokážete objaviť tvar loga (farebnosť v tejto chvíli nie je dôležitá, zvoľte si ju podľa vlastného vkusu).*
	
	Získané údaje:
	
	`5`
	`5`
	`10000111001010100111100001`

Riešenie:

Kvôli jednoznačnosti sme v úlohe zvolili „štvorcový“ obrázok (aby nevznikla zbytočná diskusia o tom, či je prvá šírka alebo výška obrázka).

Po predchádzajúcich vyučovacích hodinách je vysoko pravdepodobné, že sa žiakom podarí odhaliť spôsob zakódovania loga  . Pre potreby ďalších úloh sa so žiakmi dohodnime, že prvé číslo bude určovať šírku, druhé výšku obrázka.

Žiaci si vyskúšajú, či porozumeli princípu, v nasledujúcej úlohe:

Úloha 2	*Overme si svoje dekódovacie schopnosti na nasledujúcom súbore, ktorý obsahuje údaje na vykreslenie ďalšieho z návrhov spomínaného loga. Dokážete aj bez akýchkoľvek ďalších informácií toto logo prekresliť?* *Získané údaje:* *8* *6* *1001010101101010100000010101000110010001000100010101000010000001010101010001010101011000100101*

Riešenie:

Postupujeme v jednotlivých krokoch:

- obrázok pozostáva zo 48 bodov,
- počet bitov (číslic 0 a 1 dvojkovej sústavy) je 96,
- z predchádzajúcich dvoch bodov vyplýva, že každý z bodov je zakódovaný pomocou dvoch bitov,
- rozdelíme si celý kód na dvojice, pričom zvýrazníme posledný bod v danom riadku:

```
10|01|01|01|01|10|10|10|10|00|00|01|01|01|00|01|10|01|00|01|00|01|00|01|
01|01|00|00|10|00|00|01|01|01|01|01|00|01|01|01|01|01|01|10|00|10|01|01
```

- v takto zvýraznenom kóde vidíme tri opakujúce sa dvojice: 00, 01 a 10 => obrázok obsahuje tri rôzne farby (zvolíme si ich ľubovoľne):

VYSVETLENIE (CCA 8 MIN)

Čo všetko sme dokázali zistiť z dát súborov v úlohách 1 a 2? Vďaka znalosti rozmerov obrázka vieme vypočítať počet bodov, z ktorých sa obrázok skladá. Následne potrebujeme vedieť, koľko farieb je v obrázku použitých. V úlohe 1 sme videli, že ak ide o dvojfarebný obrázok, na zakódovanie každej z farieb stačí jeden bit – 0 pre jednu farbu, 1 pre druhú. Ak je farieb viac, rastie aj počet bitov potrebných pre zakódovanie jednej farby. V úlohe 2 mal kód dĺžku 2 a použili sme tri farby. Intuitívnym využitím tohto vzťahu (neskôr ho so žiakmi odvodíme a zovšeobecníme) medzi počtom farieb a dĺžkou kódu zistíme, že stačí vziať dĺžku celej postupnosti zapísaného obrázka a vydeliť ju počtom bodov. Získame tak informáciu koľko bitový je kód pre farby obrázka.

Farby sme si zatiaľ volili sami, podľa toho, aké sa nám páčia. Samozrejme, v skutočnosti sú jednotlivým farbám jednoznačne priradené kódy, aby daný obrázok vyzeral farebne rovnako na akomkoľvek počítači kdekoľvek na svete (ekvivalentná „dohoda“, ako pri kódovaní textovej informácie,

kde sme hovorili o znakových sadách). Existenciu rôznych farebných modelov (HSB, HLS, RGB, CMYK) môžeme spomenúť neskôr, či v spojitosti s konkrétnym grafickým editorom, s ktorým budú žiaci neskôr na hodinách inforamatiky pracovať, alebo pri téme periférnych zariadení počítača (monitor, tlačiareň).

Kvalita obrázka je tak určená nielen jeho rozmermi (počet bodov na šírku a na výšku – rozlíšenie obrázka, ale aj počtom farieb, ktoré boli pri jeho vzniku „použité“ (či už pri kreslení a následnom uložení, skenovaní či fotografovaní).

Poznámka:

Aj napriek skúsenostiam z vyučovacej hodiny venovanej kódovaniu textovej informácie je potrebné žiakom opäť zdôrazniť, že dĺžky kódov každej z farieb (a teda každého bodu) majú rovnakú dĺžku (musia mať). Nemôžeme napr. kódovať bielu farbu kódom 1, čiernu farbu kódom 0 a červenú farbu kódom 01. Ak si žiaci tieto poznatky ešte neprepojili, vyzveme ich, aby porozmýšľali, prečo je to tak.

Pri riešení ďalších úloh budeme počítať aj veľkosti súborov s grafickou informáciou. Budeme pri tom brať do úvahy len binárnu postupnosť kódujúcu jednotlivé body obrázka. Reprezentáciu údajov o šírke a výške obrázka pri výpočtoch zanedbáme.

Úloha 3 *V grafickom editore Skicár sme nakreslili obrázok, ktorého rozmery sú 640 x 480 bodov. Pri kreslení sme použili 16 farieb. Koľko bitov zaberie tento obrázok v pamäti počítača?*

Riešenie:

Žiak si musí prepojiť doposiaľ získané vedomosti:

- hľadaný počet bitov je daný dĺžkou postupnosti núl a jednotiek, ktoré reprezentujú celý obrázok,
- počet bodov vypočítame ako výsledok súčinu 640 x 480 bodov = 307 200 bodov,
- počet bitov kódu pre jednu farbu (a teda aj pre jeden bod) súvisí s počtom farieb, teda s hodnotou 16 – pomôžme si tabuľkou:

Počet farieb	Počet bitov kódu
1	1
2	1
3	2
4	2
5	3
6	3
7	3
8	3
9	4
10	4
11	4
12	4
13	4

Počet farieb	Počet bitov kódu
14	4
15	4
16	4
17	5
18	5
...	
31	5
32	5
33	6
...	
255	8
256	8
257	9
...	

Pomocou tejto tabuľky vieme vyriešiť úlohu 3 (následne aj odvodiť všeobecný vzťah medzi počtom farieb a počtom bitov kódu). Každý z 307 200 bodov zaberie v pamäti počítača 4 bity (ak je to potrebné, rozpíšeme so žiakmi kódy každej z týchto farieb), a teda celý obrázok zaberie v pamäti počítača 307 200 bodov x 4 b = **1 228 800 b**.

Aj keď sa úloha pýtala na počet bitov, určite sa neuspokojíme s takouto podobou zápisu množstva informácií – prevedieme ju na kibibajty (alebo na kilobajty – podľa uváženia učiteľa v zmysle metodickej poznámky z predchádzajúcej metodiky): 1 228 800 b = 153 600 B = 150 kiB.

Úloha 4	*Využite skúsenosť z úlohy 3 a formulujte návod, pomocou ktorého dokážeme k danému počtu použitých farieb určiť potrebnú dĺžku kódu pre každú z farieb.*

Riešenie:

Podobnú úlohu sme riešili v metodike venovanej kódovaniu textovej informácie. Pri pohľade do tabuľky z úlohy 3 vidíme, že potrebujeme nájsť najbližšiu mocninu čísla 2, ktorá sa rovná alebo je väčšia ako daný počet farieb. Exponent (stupeň mocniny) nám povie, koľko bitový je kód priradený jednotlivým farbám (bodom) obrázka. Overme si tento postup v úlohe 5.

Úloha 5	*V grafickom editore Skicár sme nakreslili obrázok, ktorého rozmery sú 640 x 480 bodov. Pri kreslení sme použili 256 farieb. Koľko kiB (kibibajtov) zaberie tento obrázok v pamäti počítača?*

Číslo 256 vieme zapísať v tvare 2^8. Na základe predchádzajúcich skúseností a zovšeobecnenia teda môžeme povedať, že na zakódovanie každého bodu obrázka použijeme 8-bitový kód.

Každý z 640 x 480 bodov = 307 200 bodov teda zaberie v pamäti počítača 8 bitov, a teda celý obrázok zaberie v pamäti počítača 307 200 bodov x 8 b = 2 457 600 b = 307 200 B = **300 kiB**.

Poznámka:

Táto situácia je zaujímavá – ak porovnáme vstupné údaje úloh 4 a 5 a ich výsledky, upozorníme žiakov na to, že keď sme zdvojnásobili počet bitov kódu, zdvojnásobila sa veľkosť súboru. Tento fakt je dôležitý na to, aby si žiaci uvedomili, že s nárastom počtu farieb rastie počet bitov kódu a, samozrejme, aj veľkosť výsledného súboru.

ROZPRACOVANIE (CCA 10 MIN)

Vyzveme žiakov, aby nasledujúcu úlohu vypracovali v grafickom editore Skicár.

Úloha 6	*V grafickom editore Skicár nakreslite ľubovoľný obrázok, ktorého rozmery sú 640 x 480 bodov. Použite nielen základné farby – namiešajte si aj rôzne odtiene. Súbor uložte do svojho priečinka pod názvom obr256.bmp vo formáte 256-farebná bitová mapa.* *Zistite skutočnú veľkosť súboru obr256.bmp. Zodpovedá vypočítanej veľkosti z úlohy 5?* *Ako je možné, že každý žiak – aj keď kreslil obrázok odlišný od obrázkov ostatných žiakov – vytvoril súbor s rovnakou veľkosťou ako ostatní žiaci?*

Riešenie:

Zistíme, že skutočná veľkosť súboru je 308 278 B. Ako si vysvetliť (malý, ale predsa len) rozdiel? Skrývajú sa za ním ďalšie údaje potrebné k správnej reprezentácii obrázka, napr. jeho rozmery.

To, že všetky žiakmi vytvorené súbory majú tú istú veľkosť, by žiakov nemalo prekvapiť, ak porozumeli princípu kódovania grafickej informácie, ktorý sme objavovali pri predchádzajúcich úlohách. Ak tomu tak nie je, snažíme sa identifikovať problematické miesta a vyjasniť ich žiakom.

Úloha 7 *V grafickom editore Skicár otvorte svoj súbor obr256.bmp, ktorý ste vytvorili v úlohe 6. Uložte ho ako obr16.bmp, pričom pri jeho ukladaní vyberte uloženie vo formáte 16-farebná bitová mapa.*

Aká je veľkosť súboru obr16.bmp?

Riešenie:

Žiaci pri ukladaní budú editorom upozornení, že pri ukladaní sa môže kvalita farieb znížiť. Výsledok žiaci (ak nevolili pri kreslení obrázka „jednoduché“ farby) uvidia. Opäť by mali byť schopní vysvetliť, prečo k tomuto zníženiu kvality dôjde. Veľkosť súboru bude polovičná v porovnaní so súborom *obr256.bmp*.

Úloha 8 *Kamil si v grafickom editore Skicár nakreslil obrázok s rozmermi 300 x 200 bodov a uložil ho ako obr24.bmp vo formáte 24-bitová mapa*

Aká je veľkosť súboru obr24.bmp? Aký maximálny počet farieb je možné použiť pri tvorbe tohto obrázka?

Riešenie:

V tomto prípade nie je v popise formátu uvedený počet farieb (my vieme, že je, aj keď sprostredkovane). Vieme ale z neho vyčítať informáciu o tom, koľko bitový je kód pre reprezentáciu každého z bodov obrázka – je to 24 bitov. Môžeme teda prejsť k výpočtu:

300 x 200 bodov x 24 b = 1 440 000 b = 180 000 B = 175,78125 kiB.

Čo sa týka počtu farieb, vieme, že každej prislúcha jednoznačný 24-bitový kód. Čiže kód tvorený dvadsiatimi štyrmi nulami a jednotkami. S využitím poznatku z úlohy 4 (a opačným postupom ako v úlohe 4) odvodíme, že počet farieb sa rovná číslu 2^{24}, čo je 16 777 216 farieb.

Úloha 9 *V grafickom editore Skicár otvorte vždy nanovo Váš súbor obr256.bmp a uložte ho vo formáte:*

a) JPEG (získate tak súbor obr256.jpg)
b) GIF (získate tak súbor obr256.gif)
c) PNG (získate tak súbor obr256.png)

(Pozor! Ak ho uložíte v úlohe a) vo formáte JPG a prejdete hneď k riešeniu úlohy b), budete ukladať nie pôvodný súbor bmp, ale už súbor JPEG.)

Porovnajte veľkosti jednotlivých súborov obr256.bmp, obr256.jpg, obr256.gif a obr256.png.

Riešenie:

Získame štyri súbory, ktoré obsahujú ten istý obrázok, no líšia sa v prípone a vo veľkosti. Túto skúsenosť môžeme využiť a nadviazať na ňu nasledujúcou vyučovacou hodinou (nie je súčasťou tejto série metodík), ktorú budeme venovať grafickým formátom.

Otázka 2	*Ako sme si v úvode hodiny povedali, vo všetkých úlohách dnešnej vyučovacej hodiny sme pracovali s tzv. bitmapovou (rastrovou) grafikou. Obrázok sa rozdelí na jednotlivé body (mriežku - raster) a počítač každý bod reprezentuje prostredníctvom farby bodu.* *Ak ste už pracovali s touto bitmapovou (rastrovou) grafikou, aké výhody a aké nevýhody ste si všimli?*

Žiaci môžu spomenúť výhody ako kreslenie podľa vlastných predstáv, úprava fotografií či už hotových kresieb a podobne. Medzi nevýhody patria napr. nemožnosť zmeniť poradie už nakreslených útvarov, deformácie pri zväčšovaní/zmenšovaní/otáčaní častí obrázka alebo celého obrázka apod.

Okrem bitmapovej (rastrovej) grafickej informácie môžeme pracovať s už spomenutou vektorovou grafikou. Používame pri tom špeciálne vektorové grafické editory (napr. InkScape, Zoner Callisto 5 – ten už však nemá podporu...). Každý útvar, ktorý v týchto editoroch nakreslíme, sa reprezentuje ako samostatný objekt s vlastnosťami (napr. šírka, výška, farba výplne, farba obrysu, štýl výplne a obrysu, tieň...) a jeho vykreslenie je priradené k matematickému vzorcu, vďaka čomu je možné tento útvar zmenšovať, zväčšovať, presúvať, otáčať,... bez zhoršenia kvality výsledného útvaru. Samozrejme, neznamená to, že rastrový grafický editor je tým pádom nezaujímavý – každý z nich má svoje špecifiká a je vhodný na iné použitie.

VYHODNOTENIE (CCA 4 MIN)

V záverečnej časti hodiny požiadame žiakov, aby vypracovali sebahodnotiaci test. Odporúčame žiakom vysvetliť a zdôrazniť, že cieľom je zistiť čo a ako si žiak z obsahu hodiny zapamätal a nie klasifikácia známkou. Pre učiteľa a žiaka zvlášť, je cenná pravdivá informácia o úrovni osvojených poznatkov než umelo vylepšená. Odporúčame žiakom poskytnúť spätnú väzbu ohľadom správnosti odpovedí. Problematické odpovede môžeme so žiakmi prediskutovať na konci vyučovacej hodiny.

Sebahodnotiaci test

1.	Obrázok nakreslený v grafickom editore Skicár sme uložili ako 24-bitovú mapu pod názvom *pokus.bmp*. Potom sme ho uložili ako 256-farebnú bitmapu pod názvom *pokus2.bmp*. Veľkosť súboru *pokus1.bmp* je: a) **tretinou veľkosti súboru *pokus.bmp***. b) osminou veľkosti súboru *pokus.bmp*. c) trojnásobkom veľkosti súboru *pokus.bmp*.
2.	Koľko bitové kódovanie grafickej informácie je potrebné zvoliť, ak chceme v rastrovom obrázku použiť 800 farieb? a) 9-bitové kódovanie, b) **10-bitové kódovanie**, c) 11-bitové kódovanie.

Odporúčame, aby učiteľ uviedol správne odpovede a na záver zhrnul nové poznatky v zmysle:

- Podobne ako číselnú či textovú informáciu, aj grafickú kódujeme pomocou binárneho kódu.
- DĺŽka použitého kódu závisí od počtu farieb, ktoré potrebujeme zakódovať.
- Veľkosť rastrového grafického súboru závisí od rozmerov obrázka (a teda počtu bodov) a od počtu farieb.
- Okrem formátu bmp môžeme pri ukladaní rastrovej grafickej informácie použiť aj iné formáty, napr. gif, jpg, png, ktoré v pamäti počítača zaberú menší dátový priestor.
- Okrem rastrovej grafiky môžeme pracovať aj s vektorovou grafikou. Takýto grafický súbor sa neukladá ako množina bodov danej farby (ako je to pri rastrovej, bitmapovej grafike), ale každý vytvorený útvar je samostatným objektom s vlastnosťami popísaný matematickým vzťahom.