

Programovanie (1) v C/C++ 2026/27

Cvičenia 4, miniprojekt

Okná

Tento **miniprojekt** je väčšia úloha, na ktorú máte viac času ako na bežné príklady z cvičení, ale máte za ňu aj viac bodov. Začnite na nej pracovať v predstihu, nenechávajte si ju na poslednú chvíľu.

Cieľom tejto úlohy je precvičiť si prácu s poliami, structami a s ďalšími základnými stavebnými prvkami jazyka C. Zadanie je o niečo dlhšie, trénujeme teda aj prácu s odborným textom. Zadanie si poriadnej prečítajte a dodržujte všetky požadované detaily.

Na miniprojekte pracujte **samostatne**, neukazujte svoj program spolužiakom a nepoužívajte nástroje umelej inteligencie. V prípade otázok kontaktujte vyučujúcich predmetu.

Študenti, ktorí majú z testu pre pokročilých uznané cvičenia 4, budú mať uznaný aj tento miniprojekt a nemusia ho odovzdávať.

V tejto úlohe budeme uvažovať obrazovku, na ktorej sa nachádza niekoľko obdĺžnikových okien rôznej veľkosti. Najspodnejšie okno predstavuje pozadie, ktoré pokrýva celú plochu obrazovky. Aj ostatné okná sa môžu navzájom rôzne prekrývať, pričom máme dané ich poradie od najspodnejšieho po najvrchnejšie. Na túto plochu kliká používateľ myšou. Každý klik sa môže nachádzať vo vnútri viacerých prekrývajúcich sa okien, ale budeme uvažovať, že používateľ klikol na najvrchnejšie z nich. Váš cieľ je zistiť pre každý klik, na ktoré okno používateľ klikol a tiež pre každé okno, koľkokrát sa naňho kliklo. Pri reálnych oknách by sa po kliknutí do okna toto okno premiestnilo na vrch plochy, ale v našom zjednodušenom scenári zostáva vzájomná poloha okien fixná. Výsledky výpočtu budeme vypisovať na konzolu, ale aj vykresľovať do obrázku pomocou knižnice SVGdraw.

Vstup: Váš program by mal vstup načítať z konzoly. Na prvom riadku budú celé čísla o a k určujúce počet okien a klikov. Predpokladajte, že $1 \leq o, k \leq 100$. Na každom z nasledujúcich o riadkov sú údaje týkajúce sa jedného okna, pričom začneme pozadím a postupujeme smerom k najvrchnejšiemu oknu. Okno je zadané piatimi celými číslami oddelenými medzerami, pričom prvé dve čísla sú súradnice ľavého horného rohu, druhé dve čísla sú súradnice pravého dolného rohu a piate číslo je identifikátor farby okna, ktorý bližšie vysvetlíme neskôr. Za zoznamom okien nasleduje ďalších k riadkov, každý s dvoma celými číslami, ktoré predstavujú súradnice kliku.

Môžete predpokladať, že prvé okno na vstupe, predstavujúce pozadie obrazovky, má ľavý horný roh $(0,0)$ a ak je jeho pravý dolný roh (p, q) , tak všetky x -ové súradnice okien aj klikov budú v rozmedzí medzi 0 a p vrátane a podobne y -ové súradnice budú medzi 0 a q . Žiadne okno ani klik teda nebude mimo hraníc našej obrazovky. Navyše pre každé okno je jeho x -ová súradnica ľavého dolného rohu menšia ako pravého dolného a podobne pre y -ové súradnice.

Výstup: Na výstup najskôr vypíšte zoznam všetkých klikov, v rovnakom poradí ako boli na vstupe. Pre každý klik vypíšte, na ktoré okno používateľ klikol, teda najvyššie položené okno spomedzi tých, ktoré klik obsahujú. Pri testovaní, či je klik v okne, považujeme hranice okna za jeho súčasť. Výstup sformátujte ako v príklade. Pre účely výstupu očísľujeme okná číslami $0, 1, \dots, o-1$ v poradí, ako boli na vstupe.

V druhej časti výstupu vypíšte pre každé okno koľkokrát sa na toto okno kliklo. Okná vypisujte v poradí ako na vstupe a dodržte formát z príkladu.

Vykresľovanie: Obrázok uložte do súboru `okna.svg` (nepridávajte cestu, t.j. meno priečinku). Súradnice bodov zo vstupu použite priamo ako súradnice v obrázku. Veľkosť obrázku nastavte podľa veľkosti prvého okna predstavujúceho pozadie obrazovky.

Každé okno vykreslite ako obdĺžnik, ktorý je vyfarbený bielou farbou, aby sa prekryli okná pod ním a okraj okna je vykreslený farbou, ktorá je zadaná na vstupe ako číslo od 0 po 9. V programe vytvorte pole 10 farieb:

```
const char *farby[10] = {"red", "green", "blue", "black", "gray",  
                        "yellow", "brown", "orange", "purple", "aqua"};
```

Potom oknu s farbou číslo f zvolíte na vyfarbovanie farbu číslo f z tohto poľa príkazom

```
setLineColor(drawing, farby[f]);
```

Kliky zobrazte ako krúžky s polomerom 2, ktoré majú farbu vyfarbovania aj farbu okraja nastavenú na tú istú farbu, akú má okno, v ktorom sa nachádzajú. Prehľad potrebných príkazov z knižnice SVGdraw nájdete na konci zadania.

Ďalšie požiadavky:

- Na ukladanie súradníc klikov použite `struct bod` z prednášky 4b, ale zmeňte typ premenných `x` a `y` na `int`.
- Na ukladanie údajov o okne si spravte `struct okno`, ktorý bude obsahovať súradnice oboch rohov, číslo farby, prípadne aj ďalšie údaje ako napríklad počet klikov.
- Naprogramujte a použite funkciu, ktorá dostane súradnice jedného kliku a jedno okno (t.j. parametre typov `bod` a `okno`) a vráti logickú hodnotu vyjadrujúcu, či je bod vo vnútri okna. Môžete napísať a použiť aj ďalšie funkcie.
- Okrem správnosti budeme pri bodovaní prihliadať aj na pekný programátorský štýl. Váš program by mal byť prehľadný, so správne odsadenými riadkami a rozumnými menami premenných a funkcií.
- Testovač bude kontrolovať iba správnosť výstupu na konzolu, nezabudnite si ručne skontrolovať vygenerovaný obrázok.

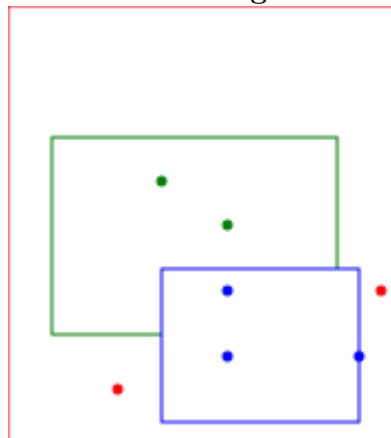
Príklad vstupu:

```
3 7
0 0 180 200 0
20 60 150 150 1
70 120 160 190 2
100 100
50 175
100 130
100 160
170 130
160 160
70 80
```

Príklad výstupu:

```
klik (100,100) je v okne 1
klik (50,175) je v okne 0
klik (100,130) je v okne 2
klik (100,160) je v okne 2
klik (170,130) je v okne 0
klik (160,160) je v okne 2
klik (70,80) je v okne 1
okno 0 pocet klikov 2
okno 1 pocet klikov 2
okno 2 pocet klikov 3
```

Obrázok okna.svg:



Prehľad potrebných príkazov z knižnice SVGdraw

- `#include "SVGdraw.h"`
- `Drawing drawing = drawingInit(velkostX, velkostY, "okna.svg")` inicializuje obrázok zadaných rozmerov
- `drawEllipse(drawing, stredX, stredY, polomer, polomer)` nakreslí kruh s daným stredom a polomerom
- `drawRectangle(drawing, rohX, rohY, sirka, vyska)` nakreslí obdĺžnik s daným ľavým horným rohom a dĺžkami strán
- `setFillColor(drawing, farby[index])` nastaví farbu na vyfarbovanie kruhov a obdĺžnikov
- `setFillColor(drawing, "white")` nastaví farbu na vyfarbovanie na bielu

- `setLineColor(drawing, farby[index])` nastaví farbu čiary na kreslenie kruhov a obdĺžnikov
- `drawingFinish(drawing)` správne ukončí vypisovanie obrázku

Pozor, bod (0,0) je v ľavom hornom rohu obrázku, os y ide nadol.