Kapitola 3. Pole

ÚLOHA 3-J-01 (VŠECHNA ÁČKA)

- V testu znaku vůči `'a'` je možno operátorem `||` (alespoň jeden) přidat testy vůči dalším variantám – A, á, Á.
- Počet variant lze snížit na polovinu, když se výchozí řetězec voláním metody `ToLower` celý převede na malá písmena.

ÚLOHA 3-J-02 (POČET DISKOVÝCH JEDNOTEK)

- Počet se zjistí jednoduše dotazem na hodnotu `Length` pole vráceného metodou `GetLogicalDrives`.

ÚLOHA 3-J-03 (V OPAČNÉM POŘADÍ)

- Cyklus `for` je třeba změnit tak, aby běžel od posledního indexu k prvnímu. Změní se všechny tři jeho části (inicializace, podmínka vstupu, aktualizace). Tělo cyklu zůstane nedotčeno.

ÚLOHA 3-J-04 (ÚPRAVY *EVIDENCE NÁJEMNÍKŮ*)

- Vystěhuj přízemí – přiřazení `null` do buňky `[0]` pole.
- Test obsazenosti – porovnáním příslušné buňky pole vůči hodnotě `null`.
- Blokace tlačítka „Vystěhovat“ – přiřazením do `Enabled` tlačítka v obsluze `TextChanged` příslušného textového pole.
- Vystěhování všech – cyklem se přiřadí `null` do všech buněk pole.

ÚLOHA 3-K-00 (CO TO DĚLÁ?)

- Představte si, že jste počítač, a vykonávejte přesně to, co je zapsáno v kódu.
- `*=` je operátorem složeného přiřazení, `a *= b;` znamená totéž co `a = a * b;`

ÚLOHA 3-K-01 (INDEX A HODNOTA)

- `a[b[2]]` se vyhodnocuje postupně, nejprve se zjistí hodnota `b[2]`, která se následně použije jako index do pole `a`.

ÚLOHA 3-K-02 (KDE JE CHYBA?)

- Jak přesně se vytváří nové inicializované pole?

ÚLOHA 3-K-03 (KDE JE DALŠÍ CHYBA?)

- Jaká je první a jaká poslední hodnota řídící proměnné zapsaného cyklu? Je to tak správně?

UKÁZKA

ÚLOHA 3-K-04 (V ČEM JE ROZDÍL?)

- Představte si to na konkrétním případě, např. pro `index` rovný 5.

ÚLOHA 3-A-00 (DATUM SLOVY)

- V programu budou dvě pole, v jednom bude 31 názvů dní, ve druhém 12 názvů měsíců. Čísla zadaná uživatelem pak poslouží jako indexy do těchto polí.

ÚLOHA 3-A-01 (KULOVÝ BLESK)

- Všichni kromě nájemníka bytu 0 se stěhují vždy do bytu o jedno číslo nižšího. Jelikož to jsou téměř stejné operace, provedou se hromadně cyklem.
- Při tomto většinovém stěhování se do každé buňky pole (vyjma poslední) přiřadí obsah jejího pravého souseda. Chcete-li v jednom okamžiku pracovat se dvěma sousedními buňkami pole, odkážete se na tyto buňky pomocí indexů `[i]` a `[i+1]`, kde `i` je v obou případech jedno a totéž číslo (jedna proměnná).
- Nájemník bytu 0 se vymyká všeobecnému schématu, nejde o byt níže, naopak jde na nejvyšší index. Není ovšem možno ho přestěhovat hned na začátku, neboť by se „přepsal“ poslední nájemník. Není ani možno čekat až na konec, neboť by sám již byl „přepsán“ nájemníkem bytu 1. Proto je na začátku třeba ho někam schovat (do proměnné pojmenované např. `skrýš` nebo `sklep`) a na konci ho odtamtud přesunout do nejvyššího bytu.

ÚLOHA 3-A-02 (ZPĚTNÝ KULOVÝ BLESK)

- Hlavním rozdílem oproti Úloze 3-A-01 (Kulový blesk) bude změna pořadí – je třeba stěhovat odshora, od bytu 12 a skončit u bytu 0. Současně se také do sklepa bude schovávat nájemník bytu 12, nikoli nájemník bytu 0.

ÚLOHA 3-A-03 (UBYTOVNA)

- V programu *Evidence nájemníků* z učebnice jsme měli pole řetězců. Pro každý byt jsme ukládali jeden řetězec. Nyní se má pro každý pokoj uložit jedno číslo (počet aktuálně obsazených lůžek), použije se tedy pole celých čísel.
- Při nastěhování se číslo v příslušné buňce pole zvětší o 1, nesmí ovšem přesáhnout trojku (počet lůžek na pokoji).
- Při vystěhování se číslo v příslušné buňce pole zmenší o 1, nesmí ovšem klesnout pod nulu.
- V náročnější variantě jsou kapacity jednotlivých pokojů (počty lůžek) uloženy v samostatném poli, do kterého se při nastěhování vždy sahá a počet lůžek na požadovaném pokoji zjišťuje.

UKÁZKA

ÚLOHA 3-A-04 (DOSTIHY PRO DESET KONÍ)

- Hlavní proměnnou programu bude pole deseti pozic jednotlivých koní. Chcete-li, aby každý kůň měl svůj obrázek, je třeba mít k dispozici také pole deseti obrázků.
- Tlačítko „Hraj“ bude pouze jedno, po jeho stisku hodí kostkou všichni koně a následně se posunou.
- Při vyhodnocení vítěze se hledá maximální číslo v poli pozic. Tato dílčí úloha se řeší klasicky procházením všech hodnot a udržováním průběžně dosaženého maxima.

ÚLOHA 3-A-05 (ODHALENÍ FALEŠNÉ KOSTKY)

- Tabulka četností se bude vytvářet stejně jako v programu *Kontrola kostky*. Jediné, co je oproti tomuto programu potřeba změnit, je naprogramovat falešnou kostku.
- Kostka falešná podle zadaných kritérií se asi nejlépe naprogramuje generováním pomocného čísla z 31 čísel. Vždy pět hodnot pomocného čísla se pak přetransformuje na jedno výsledné (hozené) číslo s tím, že na šestku zbude pomocných čísel šest ($31 - 5 \cdot 5$). Její pravděpodobnost pak bude $6/5 = 1{,}2 = 120\ \%$ pravděpodobnosti kteréhokoli jiného čísla.
- Zmíněná transformace se nejsnáze provede za pomoci celočíselného dělení.

ÚLOHA 3-A-06 (HISTOGRAM)

- Histogram se nakreslí cyklem od 1 do 6 podle hodnot z pole tabulky četností. Výška sloupce histogramu v pixelech bude rovna počtu, kolikrát dané číslo padlo.
- Detaily použití metody `DrawString` najdete ve dvanácté kapitole učebnice v části „Otáčkoměr“.
- Detaily použití přepínače z knoflíků „RadioButton“ najdete v páté kapitole učebnice, jakožto i v rozšiřujících úlohách druhé kapitolu sbírky úloh k učebnici pro začátečníky.

ÚLOHA 3-T-00 (TABULKA ČETNOSTÍ OD NULY)

- Kdykoli chcete pracovat s počitadlem zaznamenávajícím, kolikrát padlo číslo `hozenéČíslo`, obrátíte se na buňku pole počitadel s indexem `[hozenéČíslo-1]`.

ÚLOHA 3-T-01 (DRAWLINE A POINT)

```
Point bodA = new Point(100, 20);
```

UKÁZKA

ÚLOHA 3-T-02 (POLE LOGICKÝCH HODNOT)

- Z uživatelského rozhraní vypadne pole pro zadání jména nájemníka (rodiny). Bude tam pouze tlačítko „Nastěhuj“ (nějakého anonymního nájemníka).
- Do buněk pole `bool[]` se přiřazují hodnoty `true` a `false`.

ÚLOHA 3-R-00 (DVOJITÁ ČÁRA)

- Téměř vše je v zadání.
- `DrawLines` kreslí lomenou čáru, `DrawPolygon` lomenou čáru uzavřenou do výchozího bodu

ÚLOHA 3-R-01 (STRINGBUILDER)

- Nejdříve se vytvoří prázdná instance `StringBuilder`u voláním konstruktoru, poté se v cyklu volá `AppendLine` a nakonec se na výsledek zavolá `ToString`.

ÚLOHA 3-R-02 (PARAMETRY PROGRAMU)

- Získání pole parametrů:

```
string[] parametry = Environment.GetCommandLineArgs();
```

- Z hodnoty `parametry[1]` se načte obrázek. Když načítání selže (nebo když pole `parametry` má méně než dva prvky), použije se vestavěný obrázek.
- Z hodnoty `parametry[2]` se načte velikost kroku. Když načítání selže (nebo když pole `parametry` má méně než tři prvky), nastaví se velikost kroku na standardní hodnotu 5.

UKÁZKA