

DUM č. 7 v sadě

28. Inf-4 Jednoduchá hra Had ve Flashi (ActionScript)

Autor: Robert Havlásek

Datum: 14.03.2013

Ročník: 5AV

Anotace DUMu: Flash - teorie: Zavedení struktury pole. Jeho naplnění. Přístup k prvkům pole.

Materiály jsou určeny pro bezplatné používání pro potřeby výuky a vzdělávání na všech typech škol a školských zařízení. Jakékoliv další využití podléhá autorskému zákonu.



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Pole

Používáme v případech, kdy si potřebujeme systematicky pamatovat více hodnot. Flash umí pole vyrábět i dynamicky (přidávat mu další hodnoty do nových míst i za chodu programu).

Deklarovat pole můžeme jako proměnnou: `var promenna:Array`
nebo lépe s přiřazeným prázdným polem `var promenna:Array = Array()`
nebo s přiřazeným prázdným polem dané délky `var promenna:Array = Array(delka)`
nebo rovnou s hodnotami: `var promenna:Array = Array(hodnota, hodnota, hodnota)`
K jednotlivým prvkům pole přistupujeme jako `promenna[cislo]`, kde `cislo` je tzv. index, pozice prvku uvnitř pole; začíná od 0.

Na celé pole lze aplikovat funkci `trace`, která jej vypíše s čárkami.

Například:

```
var pole:Array = Array(1, 1, 2, 3, 5, 8, 13);  
pole[8]=34;  
trace(pole);
```

vypíše `1, 1, 2, 3, 5, 8, 13, undefined, 34`. Neboli sedmý prvek je nedefinován, do osmého jsme přiřadili 34.

Na poli lze aplikovat následující metody:

`pop()` ... vrátí hodnotu posledního prvku (s nejvyšším indexem) a ten z pole smaže

`push(hodnota)` ... na konec pole přidá další prvek s danou hodnotou

`shift()` ... vrátí hodnotu prvního prvku a ten z pole smaže

`unshift(hodnota)` ... na začátek pole přidá další prvek s danou hodnotou

a další, např. třídící, rozdělovací nebo naopak slučovací, uříznutí části pole, atd. – studentům netřeba zmiňovat všechny.

Z vlastností je nejzajímavější `length` ... udává délku pole (tedy nejvyšší index mínus jedna) – bez ohledu na to, kolik z těchto prvků má hodnotu `undefined`.

Pedagogická poznámka: Proměnná typu `Array` neobsahuje přímo vlastní pole, ale pouze odkaz na něj. Přístupovat k celému poli (např. přiřazením) nelze, jen k jednotlivým jeho položkám. Napíšeme-li například

```
var pole:Array = Array(1, 1, 2, 3, 5, 8, 13);  
pole2=pole;  
pole2[6]=113;  
trace(pole);
```

vypíše `1, 1, 2, 3, 5, 8, 113`. Neboli `pole` a `pole2` ukazují do stejného místa v paměti. Chceme-li pole zduplikovat, musíme vzít cyklicky každou položku zvlášť a zkopírovat ji do druhého pole.

Cyklus for

Celý for-cyklus se všemi detaily rozebereme v DUMu č. 9 této sady, v tuto chvíli předvedeme studentům pouze použití: napíši-li

```
for (var i=0;i<10;i=i+1) trace(i);
```

cyklus běží od 0, pokaždé zvětší `i` o jedna, až do okamžiku, kdy podmínka `i<10` přestane být splněná. Tento for-cyklus tedy vypíše čísla `0123456789` (oddělené konci řádků).

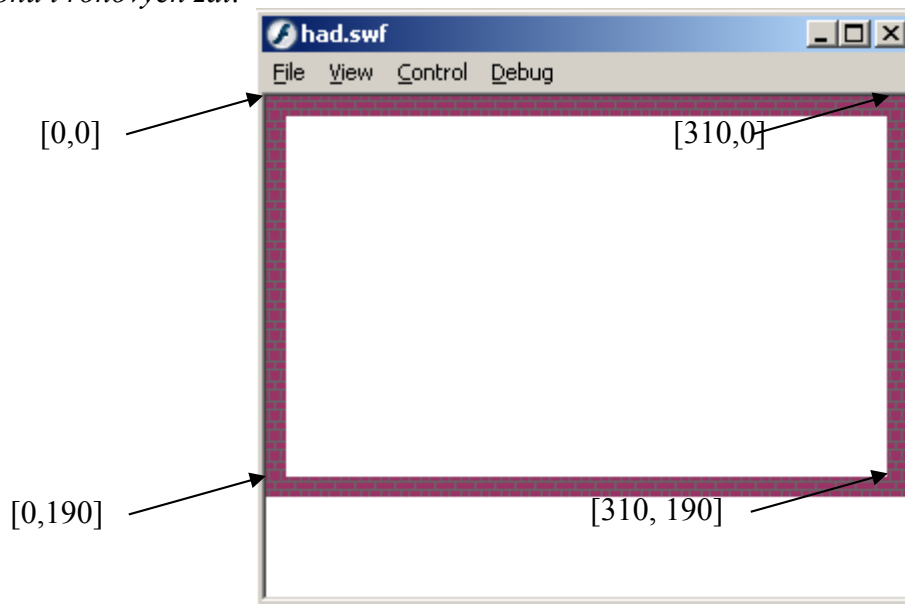
Praktický úkol: Naplňte pole hodnotami od 310 do 0 sestupujícími po deseti. Vypište jej pomocí trace.

Řešení:

```
var pole:Array = Array();
for (i=310; i>=0; i=i-10) pole.push(i);
trace(pole);
```

Praktický úkol: Naplňte dvě pole se jmény zdix, zdiy hodnotami takovými, aby každá dvojice obsahovala vždy xovou a yovou souřadnici zdi, které v souhrnu orámují obrazovku. Pole vypište pomocí trace a nechte si jej zkontrolovat vyučujícím.

Pedagogická poznámka: Na tabuli nakreslíme schéma obrazovky i souřadnice levého horního rohu i rohových zdí:



Řešení:

```
var zdix:Array = Array();
var zdiy:Array = Array();
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(0);} // horni rada
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(190);} // spodni rada
for (i=0; i<=190; i=i+10) {zdix.push(0);zdiy.push(i);} // leva rada
for (i=0; i<=190; i=i+10) {zdix.push(310);zdiy.push(i);} // prava rada
```

K variantě, kdy namísto čtyř cyklů použijeme dva, tedy:

```
for (i=0; i<=310; i=i+10) {zdix.push(i);zdiy.push(0); // horni rada
                          zdix.push(i);zdiy.push(190);} // spodni rada
for (i=0; i<=190; i=i+10) {zdix.push(0);zdiy.push(i); // leva rada
                          zdix.push(310);zdiy.push(i);} // prava rada
```

bych se nepřikláněl, připadá mi méně přehledná (cihličky se sázejí „napřeskáčku“).

Pedagogická poznámka: Obvykle je tento příklad poměrně časově náročný, studentům se pletou hodnoty, znaménka < a <=, zapisují do polí špatně. V méně šikovných skupinách je vhodné první cyklus napsat a vysvětlit.

Pedagogická poznámka: Studenti, řešení si uložte, v další hodině (viz DUM č. 8) budeme pomocí tohoto pole kreslit zdi.