METODICKÝ LIST k **DUM 16 – Pointery** ze šablony 10 - Algoritmizace a programování sada 2, tematický okruh sady: Vyšší programovací jazyky

NÁZEV DUM: **Pointery**

Anotace:	V pracovním listu se budeme zabývat adresou proměnné. Ukážeme, jak s touto adresou v programu pracovat a vysvětlíme operátor reference a dereference. Dále se budeme zabývat pointerem jako parametrem funkce a ukážeme, jak ve funkci pracovat s polem i řetězcem definovaným ve funkci main.
Autor:	Mgr. Miroslav Řezáč
Datum vytvoření DUM:	říjen 2012
Klíčová slova:	Pointer, adresa proměnné, reference, dereference
Jazyk:	čeština
Druh učebního materiálu:	Pracovní list
Stupeň a typ vzdělávání:	Střední odborné vzdělávání
Ročník:	2 nebo 3 ročník dle oboru v souladu ŠVP
Typická délka použití:	cca 90 min
Očekávaný výstup:	Studen vysvětlí operátor reference a dereference a dokáže ho použít v programu. Student vytvoří jednoduchou funkci, která pracuje s odkazem na pole nebo řetězec.
Speciální vzdělávací potřeby	žádné

Řešení:

a. **Zjistěte, co dělají následující příkazy. Předpokládejte:** `int i, *p_i;` **Které z nich nedávají smysl nebo jsou syntakticky špatně?**

`i=3;`	správně	do proměnné i se dosadí 3
`*p_i=4;`	správně	na místo kam ukazuje p_i se dosadí 4
`p_i=3;`	špatně	adresa se nastaví na 3
`i=p_i;`	špatně	do i se dosadí adresa
`i=*p_i;`	správně	do i se uloží to, na co ukazuje p_i
`i=&p_i;`	špatně	do i se uloží adresa p_i
`*p_i=i;`	správně	na místo kam ukazuje p_i se dosadí i
`p_i=&i;`	správně	do p_i se uloží adresa i

b. Doplňte tabulku za předpokladu `int i, *p_i1, *p_i2`**.**

	`i` – adresa 10	`p_i1` – adresa 20	`p_i2` – adresa 30
`i=1;`	1	?	?
`p_i1=&i;`	1	10	?
`*p_i1=2;`	2	10	?
`i=*p_i1+1;`	3	10	?
`p_i2=&i;`	3	10	10

c. Najděte chybu:

1) Špatná definice pole
   `void bubble(int *pole){…}`
2) Nesedí datové typy
   `fce(&cislo);`
3) Nesedí datové typy
   `return *x/2;`
4) Dvakrát umocníme to stejné číslo.

d. Co dělají následující funkce:

1) Zjistí délku řetězce
2) Vynuluje pole čísel