

IZP – vzorové řešení

Příklad 1: [2 body] Vysvětlete pojmy syntaxe a sémantika programovacích jazyků. (1-3 věty).

Řešení

Syntaxe: Soubor pravidel udávající přípustné konstrukce programů.

Popisuje formální strukturu programu. Definuje klíčová slova, identifikátory, čísla a další programové entity a určuje způsob, jak je lze kombinovat na základě syntaktických pravidel lze posoudit, zda určitý text je či není korektním zápisem programu v daném jazyce.

(1b)

Sémantika: Určuje logický význam jednotlivých výrazů jazyka. Ze sémantiky plyne, jaký má daná konstrukce význam. (1b)

IZP – vzorové řešení

Příklad 2: [2 body] Co je výsledkem aplikace operátoru sizeof?

Výsledkem aplikace operátoru sizeof je: *velikost jeho operandu v bytech.*

Příklad 3: [2 body] Vysvětlete 2 základní typy analýz programů. U každé uveďte příklad alespoň jedné techniky ladění programů.

Statická analýza: *zkoumá zdrojové kódy bez jejich spouštění, metoda code review.*

příklad: code review, (yellow/rubber duck debugging)

Dynamická analýza: *zkoumá průběh programu.*

příklad: krokování, ladící výpisy/záznamy

IZP – vzorové řešení

Příklad 4: [4 body] Uvedte příklady datových struktur. U každé struktury také uveďte deklaraci proměnné pro uvedený typ datové struktury.

Řešení

Homogenní: *pole*

```
int pole [10];
```

Heterogenní: *záznam*

```
struct miry {int vyska; float vaha;} jana, pavel;
```

Statická: *pole*

```
float teplota [10] [12] [31];
```

Dynamická: *lineární seznam*

```
typedef struct {char surname[15];int pay;}tdata;
```

```
typedef struct item titem;
```

```
struct item {tdata data;titem *next;};
```

```
typedef struct {titem *head;} tlist
```

```
tlist list;
```


IZP – vzorové řešení

Příklad 5: [4 body] Uvedte minimálně 4 výrazy, které se budou lišit druhem operátorů. Není nutné deklarovat použité proměnné.

Výraz s unárním a binárním operátorem: $++i * a / b$

Výraz s ternárním operátorem: $(a > b) ? a : b$

Výraz s logickým operátorem: $a \&\& b || c$

Výraz s relačním operátorem: $x < z$

IZP – vzorové řešení

Příklad 6: [6 bodů] Je dáno:

```
int x;
```

Napište **výraz**, který nabývá hodnoty true, když hodnota uložená v proměnné x **není dělitelná** žádným číslem z množiny {3,5,7} (použijte pouze multiplikativní a logické operátory!). **Nepoužívejte žádné příkazy** (ani if-else).

Řešení:

```
((x % 3 != 0) && (x % 5 != 0) && (x % 7 != 0))
```

nebo také:

```
((x % 3 ) && (x % 5 ) && (x % 7 ))
```


IZP – vzorové řešení

Příklad 7: [8 bodů] Upravte následující **dvě** funkce tak, aby každá zobrazila všechny prvky **na hlavní diagonále** čtvercové matice řádu n **po řádcích v opačném pořadí** uložené ve dvojrozměrném poli daném parametrem array.

```
void printMainDiagRev_A (int n, int array[n][n])
{
    n-1 >=0 i--
    for (int i = 0; i < n; i++) // xx
        printf ("%d \n", array[i][i]);
}

void printMainDiagRev_B (int n, int array[n][n])
{
    for (int i = 0; i < n; i++) [n-i-1][n-i-1]
        printf ("%d \n", array[i][i]); // xx
}
```

IZP – vzorové řešení

Příklad 8: [10 bodů] Je dán typ:

```
typedef struct {  
    int rows, cols; // rozměry  
    int **matrix; // 2D pole  
} Tmatrix;
```

Implementujte funkci `initMatrix`, která alokuje pole a vrátí inicializovanou hodnotu typu `TMatrix`

Řešení:

```
TMatrix initMatrix(int rows, int cols, int *errCode)  
{  
    *errCode = 0;  
    TMatrix m = { .rows = rows, .cols = cols };  
    m.matrix = malloc(rows * sizeof(int *));  
    if (m.matrix == NULL) { *errCode = -1; return m; }  
    for (int r = 0; r < rows; r++)  
    {  
        m.matrix[r] = malloc(cols * sizeof(int));  
        if (m.matrix[r] == NULL) { *errCode = r + 1; return m; }  
    }  
    return m;  
}
```


IZP – vzorové řešení

A, Příklad 9: [12 bodů] Co se zobrazí po provedení následujícího programu? Uvedte přesně výsledek, který se zobrazí na standardní výstup.

```
void zobrazeni (unsigned long long cislo, unsigned short arg_bitu) // Místo A
{

    if (arg_bitu > 2)

        zobrazeni (cislo >> 2, arg_bitu - 2);

    printf ("%llu%llu ", cislo & 1, (cislo >> 1) & 1);

}

int main (void)
{
    unsigned short pom = 121;
    zobrazeni (pom, sizeof (pom) * 4); // Místo B
    // předpokládejte, že velikost typu short je 2 bajty

    return 0;
}
```

Řešení: (co se zobrazí?): 10 11 01 10 (6b)

Význam kódu v místě A: hlavička funkce zobrazeni, jsou specifikovány dva formální parametry pro předávání hodnotou. (3b)

Význam kódu v místě B: volání funkce zobrazeni se dvěma argumenty. (3b)

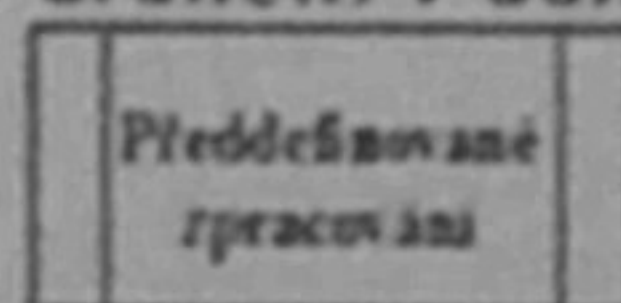
IZP – vzorové řešení

Příklad 10: [6 bodů] Uvedte základní řídicí struktury v programování (3 struktury). Každou strukturu stručně charakterizujte. Pro každou strukturu uveďte minimálně 2 příklady (příkazy), kterými lze danou strukturu v programu realizovat a jeden příklad symbolu, kterým lze daný příkaz struktury vyjádřit ve vývojovém diagramu.

Řešení:

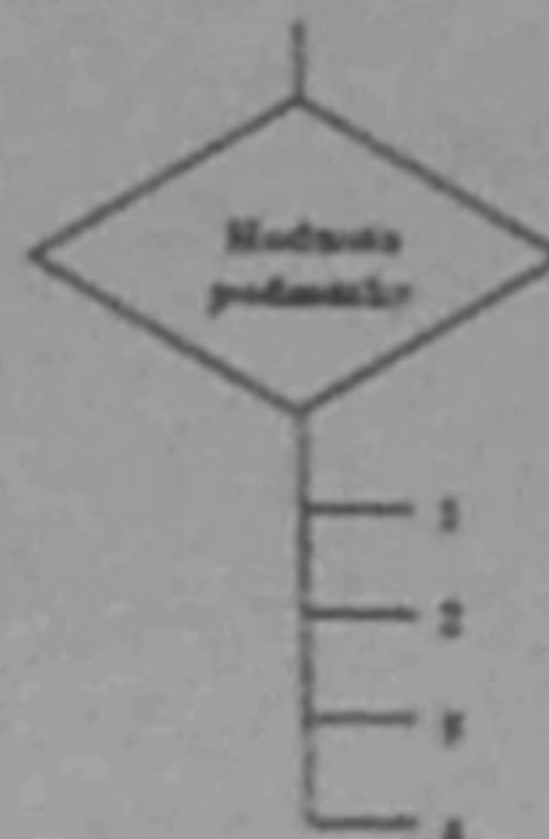
Sekvence: všechny příkazy se postupně provedou jeden po druhém v daném pořadí.

Příklad: přiřazení, volání funkcí



Selekce: v závislosti na splnění podmínky se určitý příkaz buď provede, nebo neprovede.

Příklad: větvení if-then, if-then-else, switch



Iterace: v závislosti na splnění podmínky se část programu může vykonat vícekrát.

Příklad: cykly for, while, do-while

