

Zápočtový test IDT

14. 4. 2023

Příjmení a jméno:

Osobní číslo:

Body:

1	2	3	4	5	6	7	8	celkem

Úloha 1 (6 bodů)

Zapište hlavičku a tělo metody `LCM`, která pro dvě kladná celá čísla předaná jako parametry vypočte a jako návratovou hodnotu vrátí jejich **nejmenší společný násobek**. Algoritmus nemusí být efektivní, musí však vždy vrátit **správnou** odpověď. Zvolíte-li pro řešení nějaký specifický efektivní algoritmus, zapište i jeho jméno jako komentář metody.

Úloha 2 (6 bodů)

Co vypíše následující program do konzole?

```
class MyClass {
    int number = 0;

    public static void shuffle(MyClass a, MyClass b, MyClass c) {
        a = b;
        b = c;
        c = a;
        a.number += 1;
        b.number += 2;
        c.number += 3;
    }

    public static void Main(String[] args) {
        MyClass x = new MyClass();
        MyClass y = new MyClass();
        MyClass z = new MyClass();
        shuffle(x, y, z);
        Console.WriteLine(x.number);
        Console.WriteLine(y.number);
        Console.WriteLine(z.number);
    }
}
```

Úloha 3 (6 bodů)

Do následující neúplné implementace ADT Fronta doplňte metodu `Remove()` pro odebrání prvku.

```
class Link {
    public Link next;
    public int data;
}

class LinkQueue : IQueue{
    Link first;
    Link last;

    public int Get(){
        if (first!=null)
            return first.data;
        else throw new Exception();
    }

}
```

Úloha 4 (6 bodů)

Neznámá funkce $f(n)$ **patří** do množiny $\Omega(n^2+3n+1)$. **Zakroužkujte**, do kterých dalších množin tato funkce ještě také **určitě patří**, **přeškrtněte** množiny, do kterých **určitě nepatří**:

$O(n)$	$O(n^2)$	$O(n^3)$	$O(n^4)$
$\Omega(n)$	$\Omega(n^2)$	$\Omega(n^3)$	$\Omega(n^4)$
$\Theta(n)$	$\Theta(n^2)$	$\Theta(n^3)$	$\Theta(n^4)$

Úloha 5 (7 bodů)

Následující program byl spuštěn bez parametrů příkazové řádky a zastavil se na vyznačené řádce, která dosud nebyla vykonána. Zapište celý obsah paměťové oblasti zásobník a halda. U všech pojmenovaných entit uveďte datový typ, jméno a hodnotu, u nepojmenovaných zapište třídu a všechny atributy (datové typy, názvy a hodnoty). Pokud je hodnotou reference, pak zakreslete šipku k instanci, na kterou reference odkazuje. Vyznačte zásobníkové rámce. Odpověď zapište na následující stranu.

```
class Green {
    Yellow next;
    public void Attach(int n) {
        if (n>0) {
            this.next = new Yellow();
            this.next.Attach(n-1);
        }
        else
            Console.WriteLine("Mamba is complete!");
    }
}

class Yellow {
    Green next;
    public void Attach(int n) {
        if (n>0) {
            this.next = new Green();
            this.next.Attach(n-1);
        }
        else
            Console.WriteLine("Mamba is complete!");
    }
}

public class Snake {
    Green first;
    public Snake(int n) {
        this.first = new Green();
        this.first.attach(n-1);
    }

    public static void Main(String[] args) {
        Snake s = new Snake(3);
    }
}
```


Úloha 6 (7 bodů)

Je dán následující program:

```
int n = ...;
for (int i = 0; i < n; i++) {
    M(i, 0);
    for (int j = i; j < n; j++) {
        M(i, j);
    }
}
```

Zapište počet volání metody $M(\dots)$ jako funkci proměnné n . Proved'te důkaz, že funkce patří/nepatří do množiny $\Theta(n^2)$.

Úloha 7 (6 bodů)

Uvažte následující rekurzivní program:

```
static void Hanoi(char s, char t, char m, int c) {
    if (c == 0)
        return;
    else {
        hanoi(s, m, t, c-1);
        Console.WriteLine(s + "->" + t + ", ");
        hanoi(m, t, s, c-1);
    }
}
```

Následující program představuje neúplnou nerekurzivní variantu tohoto programu. Doplňte chybějící řádky.

```
class Task {
    public char s, t, m;
    public int c, segment;
    public Task(char s, char t, char m, int c) {
        this.s = s;
        this.t = t;
        this.m = m;

        _____
        segment = 0;
    }
}

static void HanoiNR(char s, char t, char m, int c) {
    Stack<Task> tasks = new Stack<Task>();
    tasks.Push(new Task(s, t, m, c));
    while(tasks.Count>0) {
        Task task = tasks.Peek();
        switch (task.segment) {
            case 0:
                if (task.c == 0)
                {
                    tasks.Pop();
                    break;
                }
                tasks.Push(new Task(task.s, task.m, task.t, task.c-1));
                _____
                break;
            case 1:
                Console.WriteLine(task.s + "->" + task.t + ", ");
                _____
                task.segment += 1;
                break;
            case 2:
                tasks.Pop();
                break;
        }
    }
}
```

Úloha 8 (6 bodů)

Zapište kompletní zdrojový kód třídy `Tuple`, reprezentující dvojici prvků stejného datového typu. Konkrétní datový typ zvolí uživatel jako typový parametr třídy `Tuple`. Přístup k jednotlivým prvkům dvojice bude zajištěn metodami `GetFirst` a `GetSecond` (zjištění hodnoty) a `SetFirst` a `SetSecond` (nastavení hodnoty).

```
class Tuple
```

Zapište hlavičku a tělo metody `Swap` třídy `Tuple`, která prvky dvojice zamění.

```
public void Swap(
```