

obsahuje 5 príkladov, spolu $5+6+6+5+6 = 28$ bodov**1) [5 bodov]** Zistite, o po íta nasledujúca funkcia `foo` pre $n \geq 0$. *Hint: `foo(1000) = 1`.*

```
static long foo(long n) {
    return rekurzia(n, 0);
}
static long rekurzia(long n, long r) {
    if (n == 0)
        return r;
    else
        return rekurzia(n / 10, 10 * r + n % 10);
}
```

[1 bod] Pre istotu to vyjadrite pár slovami, a Oe tomu rozumiete, tak vypo ítajte `foo(911775039) = ?`**[2 body]** Napízte nerekurzívnu definíciu pre funkciu `rekurzia`, aby po ítala to isté, teda:

```
static long nerekurzia(long n) { // píšte rovno sem...
```

}

Zistite, o po íta nasledujúca funkcia `goo` pre $n \geq 0$. *Hint: `goo(13) = 11`, `goo(11) = 13`.*

```
static long goo(long n) {
    long vysledok = 0;
    while (n != 0) {
        vysledok <<= 1;
        vysledok += n & 1;
        n >>= 1;
    }
    return vysledok;
}
```

[1 bod] Vyjadrite pár slovami, a Oe tomu rozumiete, tak vypo ítajte `goo(31) = ?`**[1 bod]** Nájdite najvä zie 3-ciferné íslo také, Oe `goo(x) = x`:*Hint:*

$2^0 =$	1,	1
$2^1 =$	2,	10
$2^2 =$	4,	100
$2^3 =$	8,	1000
$2^4 =$	16,	10000
$2^5 =$	32,	100000
$2^6 =$	64,	1000000
$2^7 =$	128,	10000000
$2^8 =$	256,	100000000
$2^9 =$	512,	1000000000
$2^{10} =$	1024,	10000000000

..... riezenia príkladu 1 pízte na tento list (aj z druhej strany) alebo si vypýtajte alzí list

2) [6 bodov] Fibonacciho

Trieda `Fib` definuje uzol binárneho stromu, ktorý okrem ľavého a pravého podstromu obsahuje Fibonacciho číslo.

```
public class Fib {  
    Fib ľavy;           // ľavý podstrom  
    long kluc;         // hodnota fib.čísla  
    Fib pravý;        // pravý podstrom  
    public Fib(Fib ľavy, long kluc, Fib pravý) { ... } // konštruktor
```

Trieda `Fib` má automaticky vygenerovaný konštruktor. Fibonacciho binárny strom n -tej úrovne má ako ľavý podstrom Fibonacciho strom úrovne $n-1$, ako pravý podstrom je strom $n-2$ úrovne. Hodnotou kľuča je súčet kľučov v koreni ľavého a pravého podstromu. Lepšie si pozrite obrázok. Strom úrovne 0 a 1 je list obsahujúci 1 (ľavy=null=pravý).

V triede `Fib` definujte:

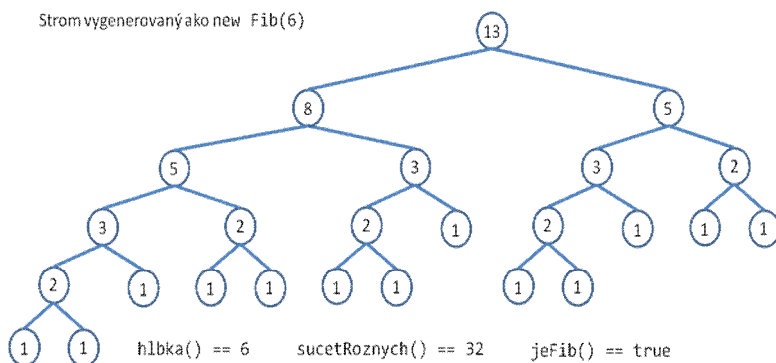
[1 bod] metódu `public Fib(int n)`, ktorá vytvorí Fibonacciho strom úrovne n .

[1 bod] metódu `int hlbka()`, ktorá vráti hĺbku stromu. Hĺbka listu (ak ľavý aj pravý sú null) je 1.

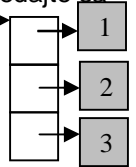
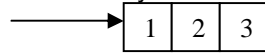
[1 bod] metódu `int listy()`, ktorá vráti počet listov Fibonacciho stromu, teda tých, kde je hodnota 1. [riešenie $O(1)$ +0.5 bodu]

[1 bod] metódu `int sucetRoznych()`, ktorá vráti súčet rôznych hodnôt Fibonacciho stromu. [riešenie $O(1)$ +0.5 bodu]

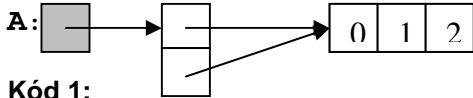
[2 body] metódu `boolean jeFib()`, ktorá vráti `true`, ak je to Fibonacciho strom nejakej úrovne.



3) [6 bodov = 0.5 bodu za každú správnu odpoveď] Ku každému obrázku napíšte kód, ktorý vytvorí znázornené pole v pamäti. Pre prázdna bunka predstavuje hodnotu null. Ak taký kód neexistuje, napíšte **NEEXISTUJE**. Nedajte sa nachyť! Uvedomte si, že Integer je referenčný typ, takže new Integer[]{1,2,3} je pole troch pointrov, ale new int[]{1,2,3} je pole troch 32-bitových čísel. Prvok pola je znázornený bielym ztvorením bez pozadia.

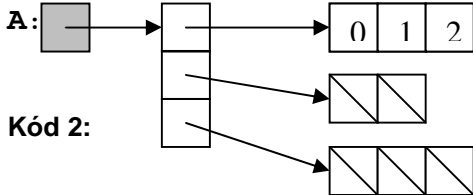


Obrázok 1:



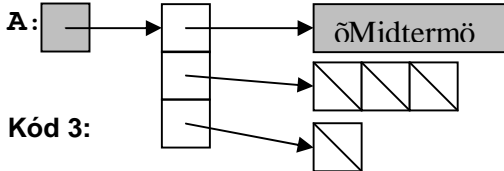
Kód 1:

Obrázok 2:



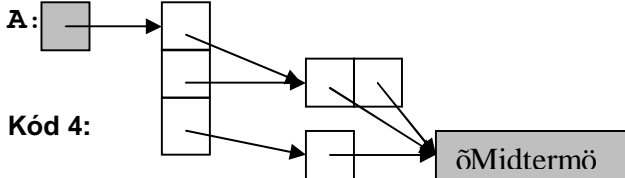
Kód 2:

Obrázok 3:



Kód 3:

Obrázok 4:



Kód 4:

Kód 5: dvojrozmerný clone s následným priradením

```
Integer[][] p = {{}, {1}, null, {3,3,3}};
p=new Integer[][]{{}, {1}, null, {3,3,3}};
Integer[][] clon = (Integer[][]).p.clone();
p[1] = p[3];
p[2] = new Integer[]{2,2};
    čo by vypísali nasledujúce printy?
System.out.println(clon[0].length);
System.out.println(clon[1][1]);
System.out.println(clon[2].length);
```

Kód 6: definujte triedy Hruska a Jablko tak, aby ste vedeli vytvoriť dvojprvkové pole s hruzkou a jablkom. Ak to nejde, napíšte **NEEXISTUJE**.

Kód 7: definujte triedu Pomelo tak, aby platilo

```
new Pomelo().equals(new Pomelo())
HashSet<Pomelo> pp = new HashSet<>();
pp.add(new Pomelo());
pp.add(new Pomelo());
pp.size() == 1
```

V pravej časti strany pre každú riadku, ktoré sú:

- syntakticky nesprávne - tie pre každú označte **S**,
- spôsobia chybu počas behu programu - tie označte **R**,
- to, čo vypíše System.out.println napíšte vedľa

Kód 8: pole s null-mi

```
String[] pole = {"java", null, "kava"};
for (String p:pole)
    System.out.println(p+p);           //?
```

Kód 9: priradenie po a rovnosť zov

```
String[] pole1 = { "Jezibaba" };
String[] pole2 = pole1;
pole2 = new String[]{"Janko", "Marienka"};
System.out.println(pole1[1]);           //?
```

```
pole1 = pole2;
pole2[1] = "Saxana";
```

```
System.out.println(pole1[1].equals("Janko"));           //?
System.out.println(pole1[1] == new String("Janko"));           //?
```

Kód 10: HashSet pre každú zlé možnosti

```
Set<String> ms = new HashSet<String>();
Set<String> ns = new Set<String>();
HashSet<String> hs = new Set<String>();
HashSet<String> gs = new HashSet<String>();
```

Kód 11: pole/list generických tried

Ak class MySet<E> extends HashSet<E>, pre každú zlé:

```
MySet<String>[] pole5 =
    new HashSet<String>[5];
MySet<String> set = new HashSet<String>();
ArrayList<MySet<String>> al =
    new ArrayList<HashSet<String>>();
```

Kód 12: prepíšte kód bez použitia (% , /)

```
int a = ľubovoľná int hodnota;
if (a % 8 > 0) a++;
int b = (a / 256) % 256;
```

3) PAPIER NA RIEŠENIA PRÍKLADU 3)

. riezenia príkladu 3 pízte na tento list, aj zozadu, alebo si vypýtajte alzí list

4) [5 body] Sie vlakových spojov reprezentujeme ako orientovaný graf, teda ako zobrazenie/dictionary z vrcholu do množiny vrcholov, kam sa z neho môžeme dostať. Ak sa z mesta nedá nikam dostať, takéto mesto do zobrazenia neuvádzame. Nech vrcholy grafu sú reálne iniciály miest, potom zruktúra je napr.:

```
Map<String, Set<String>> vlak = Map.of(
    "ZA", Set.of("PP", "BA", "KE", "PO"), // zo Ziliny idu vlaky do Popradu...
    "BA", Set.of("ZA", "PB", "BB", "KE"), ...); // z Bratislavy do Ziliny, Považskej
```

Ako sme uviedli hore, mesto, ktorého obraz by bola prázdna množina (`Set.of()`), vôbec v zobrazení nenájdete.

[1 body] Napíšte kus kódu, ktorý vytvorí zruktúru `Set<String> zeleznicna` obsahujúcu všetky mestá v hodnote premennej `vlak`. V grafovej terminológii, množina všetkých vrcholov železničného grafu.

[2 body] Napíšte kus kódu, ktorý vytvorí zruktúru `Map<String, Set<String>> vlakyOpatne`, a jej hodnotou sú opačne idúce vlaky. V grafovej terminológii, toto je orientáciu žípiek v grafe vlakových spojení.

[2 body] Podobnú zruktúru majú aj autobusové spojenia `Map<String, Set<String>> bus`. Napíšte kus kódu, ktorý vytvorí zruktúru `Map<String, Set<String>> vlakBus`, o sú spojenia s práve jedným prestupom, najprv vlakom, potom autobusom. Vlastne ide o zloženie zobrazení.

5) [6 bodov] Streamový

ítanie s porozumením:

[1bod] Nájdite aspo jedno íslo patriace do tohoto zoznamu

```
IntStream.range(1,30).filter(n -> IntStream.range(1,n).filter(i->n%i == 0).sum() == n)
    .boxed().collect(Collectors.toList())
```

Odpove :

V zozname sa nachádza číslo: ...

[1bod] Ko ko prvkov má výsledný zoznam, a ktoré íslo sa tam vyskytuje naj astejýie ?

```
Stream.of(1,2,3).flatMap(i -> Stream.of(3,2,1).map(j -> i+j)).collect(Collectors.toList())
```

Odpove :

Počet prvkov: ...

Najčastejšie: ...

[1bod správna odpove , -1 bod nesprávna odpove]

Ktoré íslo je vä ýie ?

```
int A = IntStream.range(0,100).map(i -> i*2).filter(i -> i % 6 == 0).count();
```

```
int B = IntStream.range(0,100).filter(i -> i % 6 == 0).map(i -> i*2).count();
```

Odpove :

Väčšie je: ...

Písanie s porozumením, ale **bez for/while-cyklu**:

Bez pouítia cyklu definujte zoznam typu `List<Integer>` ísel menších ako `MAX` (pouíte ako parameter, napr. `int MAX = 1000`), ktoré sú:

[1bod] delite né 2, alebo 3, alebo 5

Odpove :

[1bod] delite né 2 aj 3 aj 5

Odpove :

[1bod] ktorých ciferný sú et sa rovná íslu samotnému (napríklad 13 tam nepatrí, lebo $1+3 \neq 13$):

Odpove :

Hint: akisto viete (a my vás nechceme nachyta), 0e

- `IntStream.range(0,100)` je stream ísel 0, 1, ..., 99, ale 100 tam nepatrí,
- `.boxed()` je len konverzia z `IntStream` do `Stream<Integer>`
- rovnica $10*a+b = a+b$ má len zopár riešení, a prstoch vazich dlaní by ste ich zráтали...

..... riezenia príkladu 5 pízte na tento list (aj z druhej strany) alebo si vypýtajte alzí list