

	Meno	Priezvisko	Midterm ▼	Domáca úloha
1.	Linda	Jurkasová	25.8	25.9
2.	Dáša	Keszeghová	24.95	25.5
3.	Jana	Oravcová	24.75	23
4.	Michal	Horváth	24.45	18.15
5.	Tatiana	Gyurcsovicsová	23.25	26.5
6.	Matej	Magát	23.15	22.65
7.	Erik	Bíly	23	16.2
8.	Andrea	Hajná	22.85	24.9
9.	Eduard	Krivánek	22.05	24.1
10.	Denis	Čapkovič	21.75	19.05



	Total	1.príklad	2.príklad	3.príklad	4.príklad	5.príklad
<b>AVERAGE</b>	<b>16.27</b>	<b>4.45</b>	<b>2.25</b>	<b>2.36</b>	<b>3.55</b>	<b>3.68</b>
<b>MEDIAN</b>	<b>16.95</b>	<b>5</b>	<b>1.8</b>	<b>2.5</b>	<b>4</b>	<b>4</b>
<b>Maximum</b>	<b>25.8</b>	<b>6</b>	<b>5</b>	<b>4.5</b>	<b>5</b>	<b>6</b>
Midterm 2019				MiroB		
				TatianaG		
				AndreaH		
				NoroJ		
				LindaJ	MatúšG	
				AndrejK	MichalH	
				RichardM	LindaJ	
				JozefM	DášaK	
				ŠimonB		
				LindaJ		
				AdamV	MatejM	
				MartinS	KonrádM	
				JurajV	JanaO	
LindaJ	IvetaB					
DášaK	DenisČ					
JanaO	TomášK					
MichalH	JanaO					
TatianaG	PeterT					
MatejM	BálintZ	MichalH				



	Meno	Priezvisko	Spolu ▼
1.	Dáša	Keszeghová	127.61
2.	Eduard	Krivánek	82.61
3.	Matej	Magát	75.15
4.	Nikola	Horníková	74.49
5.	Matúš	Gál	73.51
6.	Linda	Jurkasová	73.49
7.	Maroš	Malý	72.26
8.	Iveta	Balintová	69.35
9.	Juraj	Vetrák	67.07
10.	Jana	Oravcová	67.04



	Meno	Priezvisko	Domáca úloha ▼
1.	Tatiana	Gyurcovicsová	26.5
2.	Linda	Jurkasová	25.9
3.	Iveta	Balintová	25.8
4.	Dáša	Keszeghová	25.5
5.	Juraj	Vetrák	25.3
6.	Andrea	Hajná	24.9
7.	Andrea	Smiešna	24.8
8.	Eduard	Krivánek	24.1
9.	Miroslav	Baluch	24
10.	Nikola	Horníková	23.49
11.	Jana	Oravcová	23



	Priezvisko	Prémia ▼
	Kerák	61.66
	Keszeghová	59.16
	Horníková	25.4
	Malý	23.96
	Krivánek	22.66
	Vetrák	20.01
	Gál	19.56
	Balintová	19.15
	Kormuth	14.6
	Melišík	14.513333333333333
	Hlávková	13.166666666666666



# Príklad Prvočísla

(Jožo K.)

## a. isPrime(n)

- v zadaní je napísané : "Predpokladajte, že  $n > 0$ , a 1 nie je prvočíslo!" . Napriek tomu až okolo 12 ľudí vôbec neriešilo prípad, že  $n=1$  (a z ich kódu im to na 1 vrátilo true), 3 ľudia riešili aj prípad, že  $n == 0$  a jeden/dvaja riešili aj prípad, že  $n < 0$ .
- jeden student/ka riešil/a prvočísla len tak, že každé nepárne číslo je prvočíslo, a každé párne nie je prvočíslo.

## b. primes(n)

- v zadaní je napísané "vráti zoznam prvých n prvočísel". Asi 6 ľudia však vracali zoznam prvočísel menších ako n, a nie prvých n prvočísel.

## c. sučin+1

- našlo sa zopár ľudí, ktorý nevyužívali ani jednu z predchádzajúcich dvoch funkcií, aj keď je v zadaní napísané, že im k tomu pomôžu funkcie vyššie.

## d. Mersenn

- dvaja ľudia to skúsili riešiť aj s pomocou využitia bitových posunov, jeden úspešne, jeden nie.

# Príklad Stromy

(Lukáš G.)

- a) viacero ľudí písalo referenciu "root". Implementovali ste Node, ktorý má len sam seba, nie Strom s referenciou na koreňový vrchol.
- b) všeobecný strom má list potomkov, žiadne .left, či .right... Stačilo si pozrieť atribúty v zadaní
- c) niektorí prechádzali zoznam potomkov ako for (Integer i : divisors). Znova, pozrieť atribúty! Node mal List<Node>, teda zoznam podvrcholov, nie Integerov...
- d) keď mám metódu depth()/toString() ja, tak ju majú aj všetky podvrcholy, keďže sú rovnakej triedy. Viacerí vyrábali metódy typu depthRek(Node n), ktoré "naštartovali" volaním depthrek(this) a naplnili v nich stack/queue. To samo o sebe nie je zlé riešenie, no netreba sa báť kódu:

```
for (Node child : divisors) {  
    int actual = child.depth(); <- hovorím o tomto volaní  
    if (actual > max) max = actual;  
}
```

- e) konštruktor NIKDY nič nevracia! Iba nastaví atribúty, prípadne vykoná ďalšie operácie, o "návrat" novo-vytvoreného objektu sa stará kľúčové slovíčko new.  
Takže žiadne Node n = new Node(...) ... return n; v konštruktore!

# Príklad Tretí

(Juraj H.)

- a. `System.out.println(null)` vypise "null". Preto v príklade 3a):

```
Integer[][] a = {{1, 2, 3, 4, 5, 6, 7, 8, 9, null }} // vypise  
123456789null ale nespadne
```

```
Integer[][] a = {{1, 2, 3, 4, 5, 6, 7, 8, 9}, null } // vypise  
123456789 a spadne na NullPointerException
```

- b. Caste bolo tvrdenie "Ak dva objekty maju rovnaky hash, tak su to rovnake objekty."  
Toto tvrdenie vo vseobecnosti neplati! (Opacna implikacia by vsak platit mala.) Preto v príklade 3e) bolo potrebne zadefinovat `hashCode()` a aj `equals()`.

```
class Zajac {  
    @Override  
    public int hashCode() { return 1; }  
  
    @Override  
    public boolean equals(Object o) { return true; }  
}
```

# Príklad Streamový

(Peter B.)

- a. Veľa ľudí netuší, koľko a ktoré prvky má `IntStream.range(0,100)`
- b. Niektorí netušia, ako funguje modulo `n`,  
Posledné dve cifry

- sú 99  
`i -> i % 100 == 99`
- niesú 999  
`i -> i % 1000 != 999`

1. `i -> i * 100 - 1`
2. `i -> (i - 99) % 10 == 0`
3. `i -> (i - 99) % 100 == 0`
4. `i -> (i - 99) % 100 == 0 && (i - 9) % 10 == 0`
5. `i -> i % 10 == 9 && (i / 10) % 10 == 9`
6. `i -> ... str.charAt(str.length() - 1) == '9' && ...`
7. `.filter (i -> *99)`

# Projekt

Odovzdávajte **vždy CELÉ zozipované projekty**

Pred ich odovzdávaním si preštudujte pravidlá a podmienky na projekt.

[http://dai.fmph.uniba.sk/courses/JAVA/projekt\\_pravidla.html](http://dai.fmph.uniba.sk/courses/JAVA/projekt_pravidla.html)

24.apríla po prednáške, 11:30 budú zverejnené projekty

31 projektov max. 3 riešitelia.

# Quadterm 2

- 16.5. Quadterm 2 - na cvičeniach, bez unit-testu
- 16.5. aj prvý predtermín, zápis známok tým, čo majú body, a majú hotový (uznaný) projekt
- 27.5. predpokladaný riadny termín – ale ešte nie sú podelené terminálky...

## Quadterm 2

- jednoduchá simulácia/hra s interakciou od užívateľa (myš, klávesnica)
- Čo treba vedieť:
  - kresliť do Pane/Canvasu
  - odchytať udalosti od myši/klavesnice
  - demo: jednoduchá HowTojavaFx aplikácia s Canvasom alebo Pane je tu
  - <https://github.com/Programovanie4/Java2019/tree/master/HowtoWithJavaFx>



# Späť ku concurrency (odpovedaj rýchlo)

- jedna matka porodí dieťa za 9 mesiacov, za koľko dieťa porodí 9 matiek
- vojak vykrváca za 2 hodiny, za koľko hodín vykrváca čata 30 tich vojakov
- 3 mačky zjedia 3 myši za 3 hodiny, za koľko hodín zje 100 mačiek 100 myší