



# Midterm

---



- Utorok 16.4. 18:00, posl. A, čas 90-100 min.
- prezencia ISIC
- ide o 25 bodov
- test na papieroch
- 4-5 príkladov na seprarovných papieroch, opravujeme oddelene
- témy až po prednášku StreamAPI
- vzory midtermov 2012-2018 sú na stránke predmetu
- midterm nie je o syntaxi, ale či veciam rozumiete (kódy nekompilujeme)
- syntax či font zaváži len, ak už nie je možné posúdiť, či rozumiete
- ŽIADNE elektronické pomôcky nie sú dovolené, ani mobil na lavici
- akékoľvek vlastné materiály sú povolené, nesmú „kolovať“ lavicou
- nezničte lesy, čo ste nečítali, vám určite nepomôže
- dozor ochotne odpovie na vaše korektné otázky



```
import java.util.ArrayList;  
import java.util.List;
```

```
//https://stackoverflow.com/questions/9591561/java-cartesian-  
product-of-a-list-of-lists
```

```
public class KSucin {  
  
    public static <E> List<List<E>> sucin(List<List<E>> lists) {  
        if(lists == null)  
            return null;  
        ...  
    }  
}
```



# MapSet

---

```
import java.util.*;
```

```
//https://stackoverflow.com/questions/24513400/adding-the-values-of-two-maps-when-same-key
```

```
public class MapSet {  
    public static <K, V> Map<K, Set<V>>  
        union(Map<K, Set<V>> a, Map<K, Set<V>> b) {  
  
        HashMap<K,V> ret =  
            new HashMap<K,V>((Map<? extends K, ? extends V>) a);
```



# ShortestPath

---

[//https://tutorialedge.net/artificial-intelligence/breadth-first-search-java/](https://tutorialedge.net/artificial-intelligence/breadth-first-search-java/)

[//https://stackoverflow.com/questions/16380026/implementing-bfs-in-java](https://stackoverflow.com/questions/16380026/implementing-bfs-in-java)

```
public List<String> findShortestPath(String from, String to) {
```

```
    import static java.util.Collections.swap;
```

[//https://stackoverflow.com/questions/47625101/how-can-you-reverse-an-arraylist-in-java-without-creating-a-new-list-only-use-a](https://stackoverflow.com/questions/47625101/how-can-you-reverse-an-arraylist-in-java-without-creating-a-new-list-only-use-a)

```
    int median = path.size()/2;
```

```
    for (int i = 0, j = path.size()-1; i < median; i++,j--) {  
        swap(path, i, j);  
    }
```



# Podanie vysvetlenia

---

- opravujúci si môže vyžiadať Podanie vysvetlenia k zdroju cudzej myšlienky, ak má pocit, že študent použitému kódu nerozumie. Vtedy je iniciatíva na strane študenta, aby niekomu z tímu cudziu myšlienku vysvetlil.
- Vy sa bojíte, že použijete niečo, čoho pôvod/zdroj ste neuviedli,
- My sa bojíme, že či tomu, čo používate, aj aspoň trochu rozumiete,
- Keď už to nevymyslíte sami (ideál), dajte si tú námahu to nie len obkresliť, ale aj pochopiť, aspoň niečo ste pre seba urobili,
- V opačnom prípade ste získali bezcenný bod.



# Rozklad na palindromy

---

```
public static int minRozklad(String vstup) {
    HashMap<String,Integer> hm = new HashMap<>();
    return = minRozklad_(vstup, hm);
}

public static int minRozklad_(String vstup, HashMap<String,Integer> hm) {
    Integer res = hm.get(vstup);
    if (res != null) return res;
    int min = vstup.length();
    if (jePalindrom(vstup)) min = 1;
    else {
        for(int i = 1; i < vstup.length(); i++) {
            String left = vstup.substring(0, i);
            String right = vstup.substring(i, vstup.length());
            if (jePalindrom(left)) {
                int x = 1+minRozklad_(right, hm);
                if (x < min) min = x;
            }
        }
    }
    hm.put(vstup, min);
    return min;
}
```



# Return True Again

(Jurajov aprílový vtíp na tému Reflexivita)

---

Najčastejšie riešenia:

## **Tatiana G:**

```
Field f = t.getClass().getDeclaredField("random");
f.setAccessible(true);
Field modifiers = Field.class.getDeclaredField("modifiers");
modifiers.setAccessible(true);
modifiers.set(f, f.getModifiers() & ~Modifier.FINAL);
f.set(null, new Random(0));
```

## **Nikola H:**

```
Field testovacíRandom =
TestReturnTrue.class.getDeclaredField("random");
testovacíRandom.setAccessible(true);
((Random) testovacíRandom.get(null)).setSeed(1);
```



# Return True Again

(Matej K.)

```
var f = TestReturnTrue.class.getDeclaredField("random");
f.setAccessible(true);
Random theirRandom = (Random) f.get(null);
myRandom = new Random();

/* chyba 16 bitov z celych 48, co java vygeneruje. nextLongi ch ale zahodi. */
long int1 = theirRandom.nextInt();
long int2 = theirRandom.nextInt();

/* skusme bruteforcnut zvysnych 16 bajtov, ktore nemame */
for (int i = 0; i < 65536; i++) {
    long seed = int1 * 65536 + i;
    /* "generujeme" a ak sa podari mame seed */
    if (((seed * multiplier + addend) & mask) >>> (48 - 32)) == int2) {
        System.out.println("seed: " + seed);
        System.out.println("int1: " + int1);
        System.out.println("int2: " + int2);
        myRandom.setSeed(seed ^ multiplier); // fixneme initialScramble()
                                              //lebo XOR(XOR(x)) je identita
        System.out.println("-----");
        System.out.println("int1: " + myRandom.nextInt());
        break;
    }
}
```



# Return True Again

(Filip K.)

```
if (true) {  
    // ...  
}
```