

<sup>4</sup>Broj indeksa: \_\_\_\_\_, Ime i prezime: \_\_\_\_\_

## Završni ispit iz predmeta Napredne tehnike programiranja 2022/2023.

### Poznavanje teorijskih koncepata iz programskih jezika *Python*, *Pharo* i *Rust*

Grupa 4

20 bodova

1. (7) Dopuniti sledeće rečenice. (Odgovore upisati na odgovarajuće linije):
  - a) U programskom jeziku \_\_\_\_\_, magične metode često služe za implementaciju \_\_\_\_\_. Jedan od njih je iterator.
  - b) Programski jezik *Rust* ima komponentu pod nazivom \_\_\_\_\_ koja proverava da li su sve pozajmice validne.
  - c) U kontekstu *Pharo* programskog jezika, *super* se određuje \_\_\_\_\_, u vremenu \_\_\_\_\_ (engl. \_\_\_\_\_ *time*).
  - d) U kontekstu *Rust* programskog jezika, dve vrste putanja unutar stabla modula su:
    - 1) \_\_\_\_\_
    - 2) \_\_\_\_\_
2. ([-13, 13]) Da li su navedene tvrdnje tačne (zaokružiti)?
  - a) *Python* podržava *first-class* funkcije.  
•Da    •Ne    •Ne znam
  - b) U programskom jeziku *Python*, dekoratori predstavljaju funkcije prvog reda.  
•Da    •Ne    •Ne znam
  - c) Konkurentno programiranje je podskup paralelnog programiranja.  
•Da    •Ne    •Ne znam
  - a) U kontekstu programskog jezika *Rust*, upotreba generičkog koda u većini slučajeva ne dovodi do degradacije performansi.  
•Da    •Ne    •Ne znam
  - d) U programskom jeziku *Rust*, metoda ne može uzeti vlasništvo nad *self*.  
•Da    •Ne    •Ne znam
  - e) U kontekstu programskog jezika *Rust*, navođenje tipa promenljive nije uvek obavezno.  
•Da    •Ne    •Ne znam
  - f) Programski jezik *Pharo* ne podržava leksička zatvorenja (engl. *lexical closures*).  
•Da    •Ne    •Ne znam
  - g) U kontekstu programskog jezika *Rust*, možemo istovremeno imati promenljivu i promenljivu referencu na element vektora.  
•Da    •Ne    •Ne znam
  - h) U programskom jeziku *Rust*, životni vek varijable ne mora da sadrži životni vek reference pozamljene od te varijable.  
•Da    •Ne    •Ne znam
  - i) U kontekstu programskog jezika *Rust*, osobine ne mogu da implementiraju metodu.  
•Da    •Ne    •Ne znam
  - j) U kontekstu programskog jezika *Pharo*, blok se može evaluirati najviše jednom.  
•Da    •Ne    •Ne znam
  - k) U kontekstu programskog jezika *Pharo*, povratna vrednost kaskade je povratna vrednost poslednje poruke (metode) u kaskadi.  
•Da    •Ne    •Ne znam
  - l) U kontekstu programskog jezika *Pharo*, pretraga metoda (engl. *method lookup*) u hijerarhiji metaklasa je znatno drugačija od pretrage metoda u hijerarhiji klasa.  
•Da    •Ne    •Ne znam

**Završni ispit iz predmeta Napredne tehnike programiranja 2022/2023.**

**Praktični zadaci - Python**

Grupa 4

7 bodova

1. (2) Data je hijerarhija klasa:

```
1  class C:
2      pass
3
4  class E(C):
5      def pozdrav(self):
6          print("Pozdrav iz klase E")
7
8  class A(C):
9      def pozdrav(self):
10         print("Pozdrav iz klase A")
11
12
13
14  class B(A):
15      pass
16
17  class D(B, E):
18      pass
19
20
21
22
23
24
```

Šta će biti prikazano na standardnom izlazu nakon izvršavanja sledećeg iskaza?

**D().pozdrav()** # **Output:** \_\_\_\_\_.

(Odgovor upisati na liniju iznad!)

2. (5) Pogledati sledeći isečak koda:

```
1
2  from itertools import dropwhile
3  from functools import reduce
4
5  val = reduce(lambda x, y: x + y,
6              map(lambda c: 2 ** c,
7                  filter(lambda b: b % 2 == 1,
8                          dropwhile(lambda a: a > 10,
9                                      range(100, 0, -1)))), -5)
10 print(val)
11
```

Šta će biti prikazano na standardnom izlazu nakon izvršavanja navedenog isečka koda?

# **Output:** \_\_\_\_\_.

(Odgovor upisati na liniju iznad!)

<sup>4</sup>Broj indeksa: \_\_\_\_\_, Ime i prezime: \_\_\_\_\_

## Završni ispit iz predmeta Napredne tehnike programiranja 2022/2023.

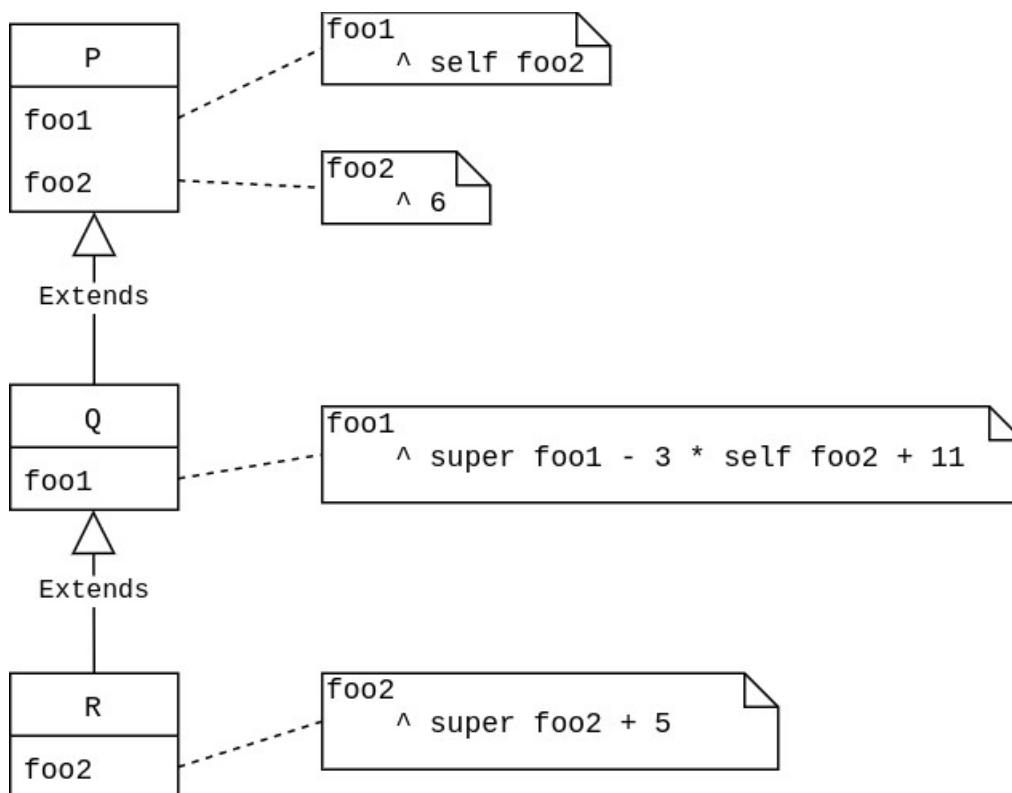
### Praktični zadaci - *Pharo*

Grupa 4

7 bodova

1. (3) Data je hijerarhija klasa prikazana na dijagramu ispod. Odrediti vrednosti sledećih izraza. (Odgovore upisati na linije navedene pored izraza):

P new foo1	_____
Q new foo1	_____
R new foo1	_____



<sup>4</sup>Broj indeksa: \_\_\_\_\_, Ime i prezime: \_\_\_\_\_

2. (4) Odrediti vrednosti sledećih izraza:

a) (1) Dat je sledeći isečak koda. Potrebno je odrediti njegov rezultat.

```
[ 2 raisedTo: 3 + 2. 3 - 4 * 3 ].
```

Rezultat prethodno navedenog isečka koda je: \_\_\_\_\_.

b) (1) Dat je sledeći isečak koda. Potrebno je odrediti njegov rezultat.

```
##(11 12 13) 3.0 10 "string" #simbol "jos jedan string") at: 0.
```

Rezultat prethodno navedenog isečka koda je: \_\_\_\_\_.

c) (2) U isečku koda ispod data je implementacija metode `withValue:` na nivou klase `YetAnotherCounter`. Zadatak ove metode je da kreira instancu klase `YetAnotherCounter`. Prethodno navedena klasa ima promenljivu instance `count`, koja predstavlja celobrojnu vrednost. Promenljiva `count` se ažurira pomoću metode `instance count:`, koja kao rezultat vraća ažuriranu vrednost promenljive `count`. Potrebno je odrediti vrednost navedenog izraza (napisanog podebljanim slovima):

```
YetAnotherCounter class >>>
  withValue: anInteger [
    ^ self new count: anInteger; yourself
  ]
```

**YetAnotherCounter withValue: (3 raisedTo: 2 + 1).**

Rezultat prethodno navedenog izraza je: \_\_\_\_\_.

<sup>4</sup>Broj indeksa: \_\_\_\_\_, Ime i prezime: \_\_\_\_\_

## Završni ispit iz predmeta Napredne tehnike programiranja 2022/2023.

### Praktični zadaci - Rust

Grupa 4

18 bodova

1. Šta će biti prikazano na standardnom izlazu nakon izvršavanja sledećih programa?

1.1 (4) Zaokružiti i upisati odgovor na odgovarajuću liniju.

```
1 fn main() {  
2     use std::collections::HashMap;  
3     let mut map = HashMap::new();  
4     map.insert(String::from("Red"), 20);  
5     map.insert(String::from("Green"), 100);  
6     let team_name = String::from("Red");  
7     let score = map.entry(team_name).or_insert(100);  
8     let score = map.get(&team_name).copied().unwrap_or(0);  
9     print!("{score}")  
10 }
```

○ Program se neće kompajlirati, zbog: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

○ Program zagantovano ispisuje: \_\_\_\_\_

1.2 (5) Zaokružiti i upisati odgovor na odgovarajuću liniju.

```
1 fn main() {  
2     use std::collections::HashMap;  
3     let mut map = HashMap::new();  
4     let red = {String::from("Red"); "Red" };  
5     let twenty = 20;  
6     let green = { String::from("Green"); "Green" };  
7     let hundred = 100;  
8     map.insert(twenty, red);  
9     map.insert(hundred, green);  
10    println!("{twenty}, {red}; {hundred}, {green}")  
11 }
```

○ Program se neće kompajlirati, zbog: \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

○ Program zagantovano ispisuje: \_\_\_\_\_

<sup>4</sup>Broj indeksa: \_\_\_\_\_, Ime i prezime: \_\_\_\_\_

2. (9) Dopisati programski kod koji nedostaje na donjim crticama, tako da se sledeći program uspešno izvrši.

```
1 fn main() {
2     let v1: _____ = vec!['a', 'b', 'c', 'd'];
3     let v2: _____ = vec!['e', 'f', 'g', 'h', 'i'];
4
5     let s1 = &v2[..];
6     let s2 = &v1[2..];
7     let s3 = &s1[1..3];
8     let s4 = &s3[1..];
9     let s5 = &s4[..1];
10
11     assert_eq!(s1, &[_____]);
12     assert_eq!(s2, &[_____]);
13     assert_eq!(s3, &[_____]);
14     assert_eq!(s4, &[_____]);
15     assert_eq!(s5, &[_____]);
16
17     let s6: _____ = &v2[_____];
18     let s7: _____ = &v1[_____];
19
20     assert_eq!(s6, &['i']);
21     assert_eq!(s7, &['c']);
22 }
```