

³Broj indeksa: _____, Ime i prezime: _____

Završni ispit iz predmeta Napredne tehnike programiranja 2023/2024.

Poznavanje teorijskih koncepata iz programskih jezika *Python* i *Rust*

Grupa 3

19 bodova

1. (7) Dopuniti sledeće rečenice. (Odgovore upisati na odgovarajuće linije):
 - a) U programskom jeziku *Python*, podrška za funkcije višeg reda je _____ modul.
 - b) U programskom jeziku *Rust*, jedan od najvažnijih koncepata jezika je skup pravila koja omogućavaju upravljanje memorijom i zove se _____.
 - c) U programskom jeziku *Rust*, greška koja se desi prilikom izvršavanja programa se zove _____.
 - d) U programskom jeziku *Rust*, mehanizam za eliminaciju dupliranja koda se realizuje pomoću _____.
 - e) U kontekstu *Rust* programskog jezika, pravila vlasništva su:
 - 1) _____
 - 2) _____
 - 3) _____
2. ([-12, 12]) Da li su navedene tvrdnje tačne (zaokružiti)?
 - a) U programskom jeziku *Rust*, pametni pointer `RefCell<T>` omogućava više vlasnika.
•Da •Ne •Ne znam
 - b) Svaki generator je i iterator u programskom jeziku *Python*.
•Da •Ne •Ne znam
 - c) Programski jezik *Rust* nema mogućnost enkapsulacije.
•Da •Ne •Ne znam
 - d) Klase su objekti u programskom jeziku *Python*.
•Da •Ne •Ne znam
 - e) U kontekstu programskog jezika *Rust*, iterator adapteri su naziv za metode koje konzumiraju iteratore.
•Da •Ne •Ne znam
 - f) Ciklične reference se mogu rešiti konverzijom iz `Rc<T>` u `Weak<T>` u programskom jeziku *Rust*.
•Da •Ne •Ne znam
 - g) U programskom jeziku *Rust*, svaka referenca ima svoj životni vek.
•Da •Ne •Ne znam
 - h) U programskom jeziku *Rust*, generički tip koji mora zadovoljiti određena ograničenja se naziva „ograničenje kroz osobine“.
•Da •Ne •Ne znam
 - i) U programskom jeziku *Rust*, generički tipovi u strukturama i metodama moraju biti isti.
•Da •Ne •Ne znam
 - j) U kontekstu programskog jezika *Rust*, paket može imati više bibliotečkih sanduka.
•Da •Ne •Ne znam
 - k) U programskom jeziku *Rust*, ruke u *match* izrazu se podudaraju u redosledu navođenja.
•Da •Ne •Ne znam
 - l) U programskom jeziku *Rust*, varijanta u *Enum*-u može da sadrži strukturu.
•Da •Ne •Ne znam

³Broj indeksa: _____, Ime i prezime: _____

Završni ispit iz predmeta Napredne tehnike programiranja 2023/2024.

Praktični zadaci - Python

Grupa 3

9 bodova

1. (3) Na osnovu navedene hijerarhije klasa, odrediti *MRO* lanac koji odgovara klasi A.

```
1  class D():
2      pass
3
4  class F():
5      pass
6
7  class B(D, F):
8      pass
9
10
11 class C(F):
12     Pass
13
14
15
16 class A(B, C):
17     pass
18
19
20
```

```
print(A.mro()) # Output: _____.
```

(Odgovor upisati na liniju iznad!)

2. (6) Pogledati sledeći isečak koda:

```
1
2 from itertools import takewhile
3 from functools import reduce
4
5
6 print(
7     reduce(
8         lambda x, y: x + y,
9         map(
10            lambda x: str(x),
11            filter(
12                lambda x: x % 5 == 2,
13                dropwhile(
14                    lambda x: x < 80,
15                    range(100)
16                )
17            )
18        )
19    )
20 )
```

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

```
# Output: _____.
```

(Odgovor upisati na liniju iznad!)

³Broj indeksa: _____, Ime i prezime: _____

Završni ispit iz predmeta Napredne tehnike programiranja 2023/2024.

Praktični zadaci - Rust

Grupa 3

26 bodova

1. (8) Odredi rezultat izvršavanja sledećih isečaka koda. Ukoliko isečak izaziva grešku, sažeto pojasniti i upisati odgovor na odgovarajuću liniju.

- a) (4) Dat je sledeći isečak koda. Potrebno je odrediti njegov rezultat.

```
1 let mut s = String::from("hello");
2 let s2 = &s;
3 let s3 = &mut s;
4 s3.push_str(" world");
5 println!("{s2}");
```

Rezultat prethodno navedenog isečka koda je: _____

- b) (4) Dat je sledeći isečak koda. Potrebno je odrediti njegov rezultat.

```
1 fn get_str(s: &str) -> &str {
2     let part = &s[0..2];
3     part
4 }
5 fn main() {
6     let d;
7     let s;
8     {
9         s = String::from("hello");
10        d = get_str(&s);
11    }
12    println!("str: {}", d);
13 }
```

Rezultat prethodno navedenog isečka koda je: _____

³Broj indeksa: _____, Ime i prezime: _____

2. $([-1, 4])$ U isečku koda ispod dato je zaglavlje (potpis) funkcije. Utvrdi da li je zaglavlje funkcije tačno napisano i odgovori na pitanja u vezi sa životnim vekom funkcije ispod.

```
1 fn overlap(s: &str, t: &str) -> &str;
```

- Da li će Rust automatski obaviti eliziju životnog veka prethodne funkcije?
 - Da • Ne • Ne znam
- Sledeće eksplicitne anotacije životnog veka su ispravne (zaokruži tačne odgovore):
 - 1) `fn overlap<'a>(s: &'a str, t: &str) -> &'a str;`
 - 2) `fn overlap<'a, 'b>(s: &'a str, t: &'b str) -> &'b str;`
 - 3) `fn overlap<'a, 'b, 'c>(s: &'a str, t: &'b str) -> &'c str;`
 - 4) `fn overlap<'a>(s: &'a str, t: &'a str) -> &'a str;`

3. (8) Odredi rezultat izvršavanja sledećih isečaka koda. Ukoliko isečak izaziva grešku, sažeto pojasniti i upisati odgovor na odgovarajuću liniju.

- a) (4) Dat je sledeći isečak koda. Potrebno je odrediti njegov rezultat.

```
1 let mut s = String::from("Hello");
2 {
3     let r = &mut s;
4     let mut add_suffix = || r.push_str(" world");
5 }
6 println!("{s}");
```

Rezultat prethodno navedenog isečka koda je: _____

- b) (4) Dat je sledeći isečak koda. Potrebno je odrediti njegov rezultat.

```
1 struct Guard;
2 impl Drop for Guard {
3     fn drop(&mut self) {
4         print!("1");
5     }
6 }
7 fn main() {
8     let _guard = Guard;
9     print!("3");
10    let _ = Guard;
11    print!("2");
12 }
```

Rezultat prethodno navedenog isečka koda je: _____

³Broj indeksa: _____, Ime i prezime: _____

1. (6) Sledeći kod se ne kompajlira/izvršava jer ima greške. Pogledati sledeći kod i u tabeli ispod navesti broj reda u kojem je greška i kratko opisati koja je greška. Ukoliko više redova ima istu grešku, grupisati ih u jednom redu date tabele.

```
1 trait Figure {
2     fn area(self) -> i32;
3     fn circumference(self) -> i32;
4 }
5 struct Rectangle<'a> {
6     width: &'a u32,
7     height: &'a u32,
8 }
9 impl<'a> Figure {
10    fn area(self) -> u32 {
11        self.width * self.height
12    }
13    fn circumference(self) -> u32 {
14        2 * (*self.width + *self.height)
15    }
16 }
17 fn main() {
18     let width = 2;
19     let height = 3;
20     let rect = Rectangle { width, height };
21     println!("Area: {}", rect.area());
22     println!("Circumference: {}", rect.circumference());
23     let fig = Figure { width, height };
24     println!("Area: {}", fig.area());
25 }
```

Broj reda	Kratak opis greške