

## Programiranje 2: 1. pisni izpit

26. april 2024

Čas reševanja je 60 minut. Veliko uspeha!



Sedež (3.11)

--	--	--	--	--	--	--	--

Vpisna številka

Ime in priimek

### 1. naloga (10 točk)

Za vsakega izmed spodnjih programov prikažite vse spremembe sklada in kopice, če poženemo funkcijo main. Za vsako spremembo označite, po kateri vrstici v kodi se zgodi.

a)

```
1 fn f(c: u32) -> u32 {
2     c + c
3 }
4 fn main() {
5     let a = 10;
6     let b = 20;
7     let d = f(b) + a;
8     println!("{d}");
9 }
```

b)

```
1 fn f(c: u32) -> u32 {
2     println!("{c}");
3     c * 10
4 }
5 fn g(d: u32) -> (u32, u32) {
6     (f(10), d)
7 }
8 fn main() {
9     let a = 10;
10    let b = 20;
11    let mut c = 3;
12    if a > b {
13        c = 10;
14    }
15    let e = g(c);
16    println!("{:?}", e);
17 }
```

c)

```
1 fn g(a: &String) -> usize {
2     a.len()
3 }
4 fn f(mut b: String) -> String {
5     println!("{}", g(&b));
6     b.push('!');
7     b
8 }
9 fn main() {
10    let m = String::from("Abeceda");
11    let n = f(m);
12    println!("{n}")
13 }
```

## 2. naloga (10 točk)

Definirajmo tip sklada `Stack<T>`, ki ob dodajanju in jemanju elementov *prevzame in vrne lastništvo*, hkrati pa omogoča vpogled, ki ne prenese lastništva.

Dopolnite signaturo spodnje implementacije. Če v dani prostor ni treba dopisati ničesar, ga prečrtajte.

```
impl<T _____> Stack<T> {

    fn contains(_____ self, elt: _____) _____ {
        // preveri, ali sklad vsebuje dani element
    }

    fn elements(_____ self) _____ {
        // vrne vektor referenc na elemente sklada
    }

    fn push(_____ self, elt: _____) _____ {
        // na vrh sklada doda element
    }

    fn pop(_____ self) _____ {
        // z vrha odstrani element in ga vrne
    }

    fn peek(_____ self) _____ {
        // na vpogled vrne zgornji element na skladu
    }

}
```

### 3. naloga (30 točk)

Za vsakega izmed spodnjih programov:

1. razložite, zakaj in s kakšnim namenom Rust program zavrne;
2. program popravite tako, da bo veljaven in bo učinkovito dosegel prvotni namen.

a)

```
fn main() {  
    let a = 0;  
    for x in 1..10 {  
        a += x;  
    }  
}
```

b)

```
fn main() {  
    let trojica = (String::from("1"), String::from("2"), String::from("3"));  
    let zadnji = trojica.2;  
    println!("{:?}", trojica);  
    println!("{zadnji}");  
}
```

c)

```
fn sestej(a: Option<u32>, b: Option<u32>) -> u32 {  
    match (a,b) {  
        (Some(a), Some(b)) => a + b,  
        (Some(a), None) => a,  
        (None, Some(b)) => b  
    }  
}
```

d)

```
fn vecji(b1: &Box<u32>, b2: &Box<u32>) -> &Box<u32> {  
    if b1 > b2 {  
        return b1;  
    } else {  
        return b2;  
    }  
}
```

e)

```
fn stevilo_manjsih<T>(v: &Vec<T>, x : &T) -> u32 {  
    let c = 0;  
    for y in v {  
        if x > y {  
            c += 1;  
        }  
    }  
    c  
}
```