

Rodjendan2021

December 14, 2021

1 Rodjendan 2021

1.1 Pozvani

```
[55]: #[derive(Debug, Clone)]
struct Posetilac {
    pub ime: String,
    pub pozvan: bool,
    pub dolazi: bool,
    pub osoba: u32,
    pub vegan: bool,
}

impl Posetilac {
    fn new(ime: &str) -> Self {
        Self { ime: ime.to_owned(), pozvan: false, dolazi: false, osoba: 1,
        ↪vegan: false }
    }
    fn pozvan(self) -> Self {
        Self { pozvan: true, ..self }
    }
    fn dolazi(self) -> Self {
        Self { dolazi: true, ..self }
    }
    fn dodaj(self, dodatno: u32) -> Self {
        Self { osoba: self.osoba + dodatno, ..self }
    }
    fn vegan(self) -> Self {
        Self { vegan: true, ..self }
    }
}

let pozvani = vec![
    Posetilac::new("Ja")
        .pozvan()
        .dolazi(),
    Posetilac::new("Milena")
        .pozvan()
```

```

        .dolazi(),
    Posetilac::new("Mk")
        .pozvan()
        .dolazi(),
    Posetilac::new("Jelena")
        .dodaj(1)
        .vegan()
        .pozvan()
        .dolazi(),
    Posetilac::new("Maksa"),
    Posetilac::new("Laza")
        .pozvan(),
    Posetilac::new("Nina")
        .pozvan(),
    Posetilac::new("Olja")
        .dodaj(1)
        .pozvan(),
    Posetilac::new("Marinko")
        .dodaj(1),
    Posetilac::new("Lero")
        .dodaj(1),
    Posetilac::new("Rista"),
    Posetilac::new("Desa"),
    Posetilac::new("Marin"),
    Posetilac::new("Guda"),
    Posetilac::new("Buda")
        .pozvan(),
    Posetilac::new("Vlajko"),
    Posetilac::new("Marko")
        .pozvan()
        .dolazi(),
];

let ukupno_ljudi = pozvani.iter().fold(0, |acc, p| {
    acc + if p.dolazi { p.osoba } else { 0 }
});
println!("Ukupno ljudi: {}", &ukupno_ljudi);
let vegana = pozvani.iter().fold(0, |acc, p| {
    acc + if p.dolazi && p.vegan { p.osoba } else { 0 }
});
println!("Ukupno vegana: {}", &vegana);
println!("Jos uvek nepozvanih: {}", pozvani.iter().fold(0, |acc, p| {
    acc + if p.pozvan { 0 } else { 1 }
}));

```

Ukupno ljudi: 6

Ukupno vegana: 2

Jos uvek nepozvanih: 8

1.2 Posluženje

```
[56]: #[derive(Debug, Clone)]
struct Posluženje {
    pub ime: String,
    pub url: String,
    pub cena: u32,
    pub komada: u32,
    pub ljudi: f64,
    pub vegansko: bool,
}

impl Posluženje {
    fn new(ime: &str) -> Self {
        Posluženje{ ime: ime.to_owned(), url: String::new(), cena: 0, komada: 1, ljudi: 0., vegansko: false }
    }
    fn url(self, url: &str) -> Self {
        Self { url: url.to_owned(), ..self }
    }
    fn cena(self, cena: u32) -> Self {
        Self { cena, ..self }
    }
    fn komada(self, komada: u32) -> Self {
        Self { komada, ..self }
    }
    fn za(self, ljudi: f64) -> Self {
        Self { ljudi, ..self }
    }
    fn vegansko(self) -> Self {
        Self { vegansko: true, ..self }
    }
}

// http://www.lavcakes.rs/vesti/clanak/
// ketering-za-vas-dogadjaj-koliko-hrane-naruciti daje reference za koliko ljudi koliko cega
let posluzenje = vec![
    Posluženje::new("Prasetina kg")
        .url("https://pecenjaramarinkovic.rs/")
        .cena(1200)
        .za(2.)
        .komada(5),
    Posluženje::new("Skordalja, kilo i po")
        .vegansko()
        .za(4.),
    Posluženje::new("Spanakorizo, kg")
```

```

        .vegansko()
        .za(3.),
    Posluzenje::new("Ruska salata")
        .za(4.),
    Posluzenje::new("Pita sa sirom, 250g")
        .url("Oklagija")
        .cena(180)
        .za(0.5)
        .komada(4),
    Posluzenje::new("Pita sa pecurkama, 250g")
        .url("Oklagija")
        .cena(150)
        .za(0.5)
        .vegansko()
        .komada(4),
    Posluzenje::new("Masline sa paprikama, kutija")
        .url("Aroma")
        .cena(300) // nemam pojma, recimo 300
        .za(0.5)
        .vegansko()
        .komada(3),
    Posluzenje::new("Urnebes, 500g")
        .url("Zlatiborski specijaliteti")
        .cena(500) // nemam pojma
        .za(0.5),
    Posluzenje::new("Kiseli kupus, kg")
        .url("Aroma")
        .cena(200) // nemam pojma
        .vegansko()
        .za(0.5),
    Posluzenje::new("Paket ljutih papricica")
        .url("Aroma")
        .cena(260)
        .vegansko()
        .za(2.),
];

let para_za_posluzenje = posluzenje.iter().fold(0, |acc, p| acc + p.cena*p.
    ↪komada);
println!("Ukupno para: {} din", &para_za_posluzenje);
let hrane_za_ljudi = posluzenje.iter().fold(0., |acc, p| acc + p.ljudi*p.komada,
    ↪as f64);
println!("Za {} ljudi", &hrane_za_ljudi);
let veganske_hrane = posluzenje.iter().fold(0., |acc, p| {
    acc + if p.vegansko { p.ljudi*p.komada as f64 } else { 0. }
});
println!("Za {} vegana", &veganske_hrane);

```

Ukupno para: 9180 din
Za 29.5 ljudi
Za 13 vegana

1.3 Pice

```
[57]: #[derive(Debug, Clone)]
struct Pice {
    ime: String,
    komada: u32,
    ljudi: f64,
    cena: u32,
}

impl Pice {
    fn new(ime: &str) -> Self {
        Self { ime: ime.to_owned(), komada: 0, ljudi: 0., cena: 0 }
    }
    fn komada(self, komada: u32) -> Self {
        Self { komada, ..self }
    }
    fn ljudi(self, ljudi: f64) -> Self {
        Self { ljudi, ..self }
    }
    fn cena(self, cena: u32) -> Self {
        Self { cena, ..self }
    }
}

let pice = vec![
    Pice::new("Banjalucko")
        .ljudi(0.25)
        .cena(62)
        .komada(48),
    Pice::new("Belo vino")
        .ljudi(0.5)
        .cena(600) // odokativno
        .komada(3),
    Pice::new("Rakija")
        .ljudi(3.0)
        .cena(1500) // odokativno
        .komada(1),
    Pice::new("Viski iz tim99")
        .ljudi(3.0)
        .cena(1500) // cini mi se
        .komada(1),
    Pice::new("Koka-kola")
```

```

        .ljudi(1.5)
        .cena(100)
        .komada(3),
    Pice::new("Kisela voda")
        .ljudi(1.5)
        .cena(100)
        .komada(3),
];

let para_za_pice = pice.iter().fold(0, |acc, p| acc + p.cena*p.komada);
println!("Ukupno para: {} din", &para_za_pice);
let pice_za_ljudi = pice.iter().fold(0., |acc, p| acc + p.ljudi*p.komada as f64);
println!("Za {} ljudi", &pice_za_ljudi);

```

Ukupno para: 8376 din
Za 28.5 ljudi

1.4 Summa summarum

```

[58]: let hrana = hrane_za_ljudi / ukupno_ljudi as f64;
if hrana > 1. {
    println!("Suviše hrane ({}× više)", hrana);
} else {
    println!("Premalo hrane ({}× manje)", 1./hrana);
}

let veg_odnos = veganske_hrane / vegana as f64;
if veg_odnos > 1. {
    println!("Suviše veganske hrane ({}× više)", veg_odnos);
} else {
    println!("Premalo veganske hrane ({}× manje)", 1./veg_odnos);
}

let pice = pice_za_ljudi / ukupno_ljudi as f64;
if pice > 1. {
    println!("Suviše pica ({}× više)", pice);
} else {
    println!("Premalo pica ({}× manje)", 1./pice);
}

let ukupan_trosak = para_za_posluzenje + para_za_pice;
println!("Ukupan trosak za rođendan: {} din", &ukupan_trosak);

```

Suviše hrane (4.916666666666667× više)
Suviše veganske hrane (6.5× više)
Suviše pica (4.75× više)
Ukupan trosak za rođendan: 17556 din