Rodjendan2022

December 13, 2022

1 Rođendan 2022

1.1 Pozvani

[10]:
```rust
#[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 pozvana(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")
        .pozvana()
        .dolazi(),
    Posetilac::new("Mk")
        .pozvan()
        .dolazi(), // najverovatnije
    Posetilac::new("Jelena")
        .dodaj(1)
        .vegan()
        .pozvana()
        .dolazi(),
    Posetilac::new("Maksa")
        .pozvan()
        .dolazi(),
    Posetilac::new("Laza")
        .pozvan()
        .dolazi(),
    Posetilac::new("Nina")
        .pozvana(),
    Posetilac::new("Olja")
        //.dodaj(1)
        .pozvana()
        .dolazi(),
    Posetilac::new("Marinko")
        .dodaj(1)
        .pozvan(), // Ne moze da dodje
    Posetilac::new("Lero")
        .dodaj(1),
    Posetilac::new("Rista")
        .pozvan()
        .dolazi(),
    Posetilac::new("Desa")
        .pozvan(), // Na Kopu
    Posetilac::new("Marin")
        .pozvan()
        .dolazi(),
    Posetilac::new("Guda")
        .pozvan()
        .dolazi(),
    Posetilac::new("Buda"),
    Posetilac::new("Vlajko")
        .pozvan(),
    Posetilac::new("Marko")
        .pozvan()
        .dolazi(),
    Posetilac::new("Isis")
```

```
        .pozvana()
        .dolazi(),
    Posetilac::new("Marija")
        .pozvana()
        .dolazi(),
    Posetilac::new("Jovana")
        .pozvana()
        .dolazi(),
    Posetilac::new("Jovan")
        .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: 16
Ukupno vegana: 2
Jos uvek nepozvanih: 2
```

1.2 Posluzenje

[11]:
```
#[derive(Debug, Clone)]
struct Posluzenje {
    pub ime: String,
    pub url: String,
    pub cena: u32,
    pub komada: u32,
    pub ljudi: f64,
    pub vegansko: bool,
}

impl Posluzenje {
    fn new(ime: &str) -> Self {
        Posluzenje{ 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![
    Posluzenje::new("Rebarca kg")
        .url("https://youtu.be/jfAe3Axrw7I")
        .cena(1000) // makar u aromi
        .za(2.)
        .komada(3),
    Posluzenje::new("Skordalja, kg")
        .vegansko()
        .za(4.)
        .komada(2),
    Posluzenje::new("Spanakorizo, kg")
        .vegansko()
        .za(3.),
    Posluzenje::new("Ruska salata")
        .za(4.), // ? (turci dodaju pavlaku uz majonez, mozda ima smisla)
    Posluzenje::new("Sendvici")
        .url("Stankovi neki sa fakulteta")
        .cena(10500)
        .za(10.),
    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(0.5),
];

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: 15360 din
Za 34 ljudi
Za 13.5 vegana
```

1.3 Pice

[12]:
```
#[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("Badvajzer")
        .ljudi(0.25)
        .cena(86)
        .komada(24),
    Pice::new("Amstel")
        .ljudi(0.25)
        .cena(59)
        .komada(24),
    Pice::new("Belo vino")
        .ljudi(0.5)
        .cena(600) // odokativno
        .komada(3),
    Pice::new("Rakija") // popadic dunja sa kalenica
        .ljudi(3.0)
        .cena(1000)
        .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(108)
        .komada(2),
    Pice::new("Kisela voda")
        .ljudi(1.5)
        .cena(55)
        .komada(5),
];

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: 8271 din
Za 30 ljudi
```

1.4 Summa summarum

[13]:
```rust
let hrana = hrane_za_ljudi / ukupno_ljudi as f64;
if hrana > 1. {
    println!("Suvise hrane ({}× vise)", hrana);
} else {
    println!("Premalo hrane ({}× manje)", 1./hrana);
}

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

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

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

```
Suvise hrane (2.125× vise)
Suvise veganske hrane (6.75× vise)
Suvise pica (1.875× vise)
Ukupan trosak za rodjendan: 23631 din
```