АКАДЕМИЈА ТЕХНИЧКО-ВАСПИТАЧКИХ СТРУКОВНИХ СТУДИЈА
ОДСЕК НИШ

# PROGRAMSKI ALATI ZA RAZVOJ SOFTVERA

## Vežba 4: Primer prvog kolokvijuma

Ova vežba obuhvata koncepte koje smo prelazili u prethodnim nedeljama, deo dobijate sa rešenjem, a ostalo radite sami i predajete kao .zip datoteku kada bude odbrana. Na kolokvijumu će biti neki sličan primer, sa malo manjim brojem klasa. Možete da radite na papiru ili laptopu.

### Sistem za upravljanje inventarom prodavnice tehnike

Napraviti sistem za praćenje i upravljanje inventarom prodavnice tehnike, u kojem kupci mogu naručivati proizvode. Sistem treba da obuhvata klase za proizvode, kupce i porudžbine, sa odgovarajućim funkcijama za ažuriranje inventara, dodavanje kupaca, kreiranje porudžbina, i sl.

**1. Dizajn klasa**

**Glavne klase koje treba implementirati:**

- **Device** - Apstraktna klasa koja predstavlja osnovne karakteristike uređaja.
- **MobilePhone, Laptop, Television** - Podklase klase Device, koje nasleđuju zajedničke atribute i ponašanja, a takođe imaju i specifične atribute.
- **Inventory** - Klasa koja sadrži kolekciju Device objekata i može da pruža metode za dodavanje, uklanjanje i pretragu uređaja u inventaru.
- **Customer** – Klasa koja sadrži osnovne podatke o kupcu koji će kreirati porudžbine.
- **Order** – Klasa koja sadrži jednu porudžbinu koju je napravio neki kupac.

**Osnovna struktura klasa:**

**Device** (apstraktna klasa):

- brand: Marka uređaja.
- price: Cena uređaja.
- quantity: Količina na zalihama.

Metode:

- get_details(): Apstraktna metoda koja vraća detalje o uređaju, naravno i sve nasleđene klase je imaju, a sam ispis je prilagođen datom uređajaju (laptopu, telefonu, televizoru).

**MobilePhone**:

- Nasleđuje Device.
- Dodatni atributi: operating_system, storage_capacity.

**Laptop**:

- Nasleđuje Device.
- Dodatni atributi: processor, ram.

**Television**:

- Nasleđuje Device.
- Dodatni atributi: screen_size, is_smart.

**Inventory**:

- Lista objekata klase Device.
- Metode za dodavanje uređaja, brisanje uređaja, ažuriranje uređaja, prikaz celokupnog inventara, kao i promenu statusa poruždbina. Za to će biti zadužen prodavac.

**Order:**

- Klasa za poruždbinu nekog kupca.
- Sadrži instancu tog kupca, listu proizvoda koje je kupio, datum kada je poruždbina napravljena i status (poslata, preuzeta, otkazana).
- Metode dodajete po želji, sve koje smatrate da su neophodne (kreiranje poruždbine, štampanje listinga porudžbine, izračunavanje vrednosti, primena popusta, itd)

**Customer:**

- Sadrži atribute kao što su ime kupca, telefon, adresa, email.
- Takođe sadrži i listu svih porudžbina koje je napravio.
- Metode dodajete po želji, koje smatrate da su neophodne (promena podataka o kupcu, pretraga poruždbina, izračunavanje koliko je do sada potrošio u prodavnici, itd)

**2. Relacije između klasa:**

- Klase MobilePhone, Laptop, i Television nasleđuju Device, što omogućava korišćenje polimorfizma i jedinstvene implementacije za uređaje. Ovo je klasična **generalizacija**.
- Klasa Inventory koristi Device objekte u svojoj listi i omogućava operacije nad kolekcijom. Predlog relacije je ovde **kompozicija,** s obzirom da uređaji postoje kao deo inventara, i kada se inventar uništi ili obriše, svi uređaji u njemu su takođe obrisani.
- Klasa Order sadrži listu proizvoda (instanci klase Device) kao i instancu kupca koji je tu poruždbinu napravio. Predlog relacije je ovde **agregacija** jer ona ovde označava da Order referencira određene uređaje, ali ti proizvodi svakako postoje nezavisno od Order objekta.
- Klasa Customer sadrži listu Order instanci. Kada kupac kreira novu porudžbinu, ona se dodaje u listu porudžbina tog kupca. Predlog relacije je ovde **agregacija** jer ona pokazuje da Order objekti pripadaju određenom kupcu, ali su i dalje nezavisni objekti i mogu postojati izvan Customer klase.

**3. Use Case dijagram** (crtate sami za ovu vežbu)

**Akteri:**

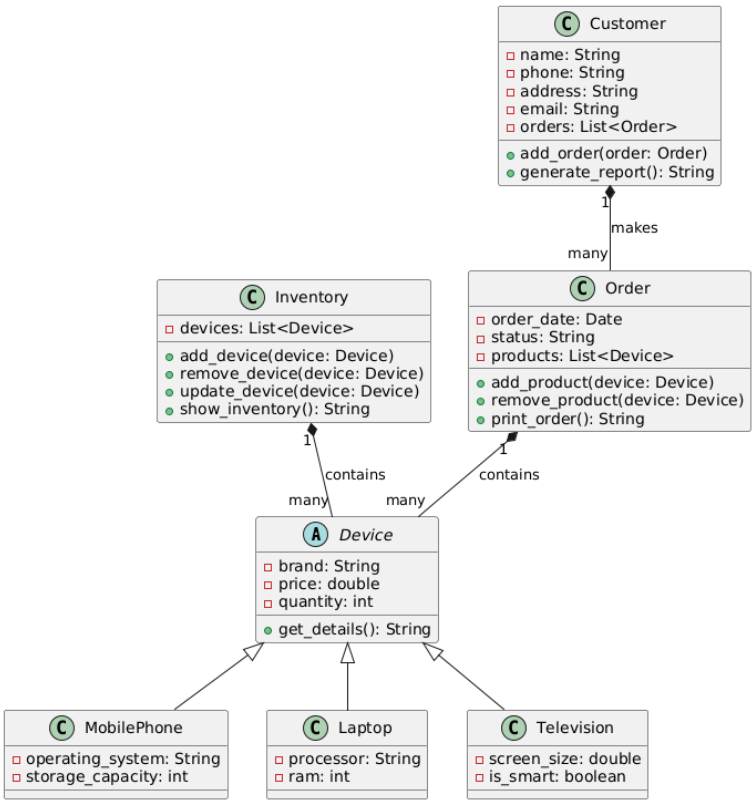
- Kupac: Osoba koja kupuje uređaje.
- Prodavac: Osoba koja upravlja inventarom i obradom porudžbina.

**Predlog operacija:**

- **Kupac:**
    1. **Pretraga proizvoda:** Kupac može pretraživati proizvode u prodavnici.
    2. **Naručivanje proizvoda:** Kupac može dodati proizvode u svoju porudžbinu.
    3. **Pregled porudžbina:** Kupac može pregledati prethodne porudžbine, generisati izveštaje o poručenim proizvodima i dosadašnjoj potrošnji u prodavnici.
- **Prodavac:**
    1. **Dodavanje novih proizvoda**: Prodavac može dodati nove proizvode u inventar.
    2. **Ažuriranje proizvoda:** Prodavac može ažurirati određene informacije o proizvodu (npr. količina, cena).
    3. **Uklanjanje proizvoda:** Prodavac može ukloniti proizvode iz inventara.
    4. **Pregled inventara:** Prodavac može pregledati sve proizvode u inventaru.
    5. **Ažuriranje statusa porudžbine:** Prodavac može promeniti status porudžbine (poslata, preuzeta, otkazana).

**4. UML Klasni dijagram**

Predlog UML klasnog dijagrama je dat u nastavku. Vaš dijagram treba da odgovara metodama koje ste prethodno definisali. U sklopu rešenja ove vežbe predajete svoj originalan dijagram.



**5. jUnit testovi**

U nastavku je dato par osnovnih jUnit testova. Obratite pažnju da uz jUnit test možete dodati i neki komentar o uspehu, kao što je u metodama testAddDevice i testAddOrder navedeno. To nije obavezno, ali može biti jako korisno kada radite sami.

Vaš zadatak je ovde da implementirate još najmanje pet testova, korišćenjem što više poznatih asserta. Možete raditi i u Pythonu ako vam je lakše. Za odbranu vežbe je potrebno da predate kompletan source kod i testove, zajedno sa slikama za dijagrame.

```
@Test
    public void testAddDevice() {
        Inventory inventory = new Inventory();
        MobilePhone phone = new MobilePhone("Samsung", 500.00, 10, "Android", 128);
        inventory.add_device(phone);
        assertEquals(1, inventory.getDeviceCount(), "Novi uredjaj je dodat u inventar!");
    }


@Test
    public void testCalculateInventoryValue() {
        Inventory inventory = new Inventory();
        MobilePhone phone = new MobilePhone("Samsung", 500.00, 10, "Android", 128);
        Laptop laptop = new Laptop("Dell", 1000.00, 5, "Intel i7", 16);
        inventory.add_device(phone);
        inventory.add_device(laptop);
        double expectedValue = 10000.00
        assertEquals(expectedValue, inventory.calculate_inventory_value())
    }


@Test
    public void testAddOrder() {
        Customer customer = new Customer("Petar Petrovic", "0611234567", "Vozda Karadjordja
                                        7, 18000 Nis", "peki@example.com");
        Order order = new Order(customer);
        customer.add_order(order);
        assertTrue(customer.getOrders().contains(order), "Porudzbina je dodata kupcu!");
    }
```