

1. Opis i sadržaj vežbe

- Rad sa Stack memorijom
 - Skladištenje vrednosti promenljivih primitivnih tipova – Rad sa stack memorijom,
- Rad sa Heap memorijom i Objekti u javi
 - Upoznavanje sa objektima u Javi i rad sa heap memorijom,
- Rad sa nizovima primitivnih tipova
 - Kreiranje i korišćenje nizova – jednodimenzionalni i višedimenzionalni,
 - ispis i obrada nizova kroz for petlju,
- Metode (funkcije), matematičke funkcije - klasa Math
 - Čitanje args main metode,
 - Rad sa funkcijama (ključna reč return, povratni tipovi i tipovi ulaznih parametara, castovanje povratne vrednosti, prosleđivanje parametara koji sadrže primitivnu vrednost na steku vs prosleđivanje parametara koji sadrže referencu na objekat/niz koji se nalazi na heapu),
 - Matematički proračuni,
- Klasa String,
 - Rad sa stringovima, gde se skladište, klasa String, metode String klase,
 - Splitovanje stringova i leksičko poređenje stringova,
- Wrapper klase za primitivne tipove,
 - Klase za predstavu primitivnih tipova kao Java objekata,
 - Dobijanje vrednosti primitivnih tipova iz teksta korišćenjem Wrapper klase za primitivne tipove,
 - Boxing i Unboxing,
- Klasa ArrayList i objekti Wrapper klasa,
 - Kolekcije - Liste koje sadrže objekte Wrapper klase za primitivne tipove,
 - Poređenje rada sa nizom,
 - Metode add(Object), add(index, Object), remove(index) i get(index).

Dodatni materijal:

- Upoznavanje sa formatima ispisa,
 - Pozivanje različitih formalizovanih ispisa na ekran,
 - Pisanje ćirilčnih i latiničnih slova u Java fajlu, Unicode predstava
 - Ispis ćirilčnih i latiničnih slova na konzolu, podešavanje pokretanja aplikacije ili eclipse alata da rade sa UTF-8 enkodingom
- Klase StringBuilder, StringBuffer,
 - Korišćenje,
 - Prednosti u odnosu na običnu kontaktenaciju stringova
- Klasa StringTokenizer,
- Podela memorije u Javi 1.8.,
 - Objašnjavanje Memorije za niti, Non-hep memorije i Heap memorije,
- Oslobođanje neiskorišćene memorije u Javi,
 - Garbage collector
- Razlike između Heap i Stack memorije.

2. Literatura

materijali/slajdoviTermin2.pptx.pptx
/funkcijeDodatno.doc

3. Primeri

primer01 - Nizovi
primer02 - Kreiranje funkcije koja računa hipotenuzu pravouglog trougla.
primer03 - Stringovi i splitovanje teksta.
primer04 - Wrapper klase za primitivne tipove.
primer05 - ArrayList, parsiranje teksta i preuzimanje podataka

Dodatni materijal:

primerDodatnoFormatiranjeIspisa – Primeri formatizovanog ispisa na ekran
primerDodatnoKonkatenacijaTeksta – Razlika konkatenacije teksta upotrebom klasa String, StringBuilder i StringBuffer

4. Zadaci na času

Zadatak01 - Implementirati aplikaciju koja će za kupljene artikle omogućiti prikaz fiskalnog isečka (računa).

Na račun prvo ispisati ime prodavnice (npr. Megamarket Metro), a zatim stavke računa. Na kraju računa ispisati ukupnu cenu bez pdv-a i sa pdv-om (pdv 20%).

Ime prodavnice se prosleđuje prilikom pokretanja programa kroz *args main* metode. Stavka računa definisana sa nazivom artikla, količinom, cenom po komadu.

Ukupna cena se računa, a ostalo je uskladišteno pomoću lokalnih promenljivih tipa niz.

Zadatak02 - Prebaci Zadatak01 tako da se umesto nizova koriste liste

5. Domaći zadaci

Dom01 -
Proširiti zadatak sa časa tako da postoji i informacija o tome u čemu je izražena količina (kilogramima, litrama, komadima).
Neki artikli mogu biti na akciji od 10%.
Neki proizvodi imaju 10%, a neki 20% pdv-a, a neki su oslobođeni pdv-a.
Prilikom obračuna ukupne cene (bez pdv-a i sa pdv-om) voditi računa o akcijama i različitim stopama pdv-a.

Dom02 –

Pročitajte zatatak naveden u fajlu „studentska_sluzba_dijagram_klasa.pdf“ (tumačenje celokupnog modela podataka biće u narednom terminu).

Kombinujući zadatak Strings02 i Liste uradi sledeće.

Implementirati aplikaciju koja omogućuje prikaz određenih entiteta Studentske Službe.

Za entitete potrebno je definisati *liste* ili *nizove* odgovarajućih tipova čije će se vrednosti popuniti parsiranjem teksta. Potrebno je za svaki tip entiteta napraviti funkcije

1. Učitaj podatke
2. Ispiši sve podatke
3. Ispiši podatak u odnosu na identifikator (predmet_id, nastavnik_id,...)

Entiteti nastavnika su opisani *id* (identifikatorom), *imenom*, *prezimenom* i *zvanjem*. Tekst koji se parsira:

"1,Petar,Petrović,Docent\n2,Jovan,Jovanović,Docent\n3,Marko,Marković,Asistent\n4,Nikola,Nikolić,Redovni Profesor\n5,Lazar,Lazić,Asistent"

Entiteti predmeta su opisani tekstem:

"1,Matematika\n2,Fizika\n3,Elektrotehnika\n4,Informatika"

Entiteti studenta su opisani *id* (identifikatorom), *indeksom*, *prezimenom*, *imenom* i *gradom* iz koga dolaze. Tekst koji se parsira:

"1,E1 01/2011,Srđanov,Konstantin,Loznica\n2,E1 02/2012,Baki,Strahinja,Novi Sad\n3,E1 03/2013,Trajković,Nebojša,Indija\n4,E2 01/2011,Sekulić,Miloš,Vršac\n5,E2 02/2012,Askin,Vuk,Novi Sad\n6,E2 03/2013,Klainić,Marko,Sombor\n7,E2 04/2011,Marko,Panić,Zrenjanin"

Dom03 -

Proširiti zadatak Dom02, tako da se kreira funkcija koja omogućuje ispis svih studenta koji dolaze sa smera E2 (Računarstvo i Automatika)

Dom04 -

Proširiti zadatak Dom03, tako da se kreira funkcija koja omogućuje ispis statistike broja upisanih studenata po godinama upisa.

Ispis bi trebao da bude nalik:

U 2011 je upisalo fakultet 3 studenta

U 2012 je upisalo fakultet 2 studenta

U 2013 je upisalo fakultet 2 studenta

Dom05 -

U *xls* fajlu *rasporedKolizije* prikazana je inicijalna verzija rasporeda predavanja.

Raspored sadrži kolizije koje se ogledaju u preklapanju zauzeća kabineta, preklapanju angažmana predavača.

Potrebno je detektovati i ispisati sve kolizije ako je pretpostavka da su termini sortirani po početku održavanja.

1. Sve kolizije preklapanja zauzeća kabineta
2. Sve kolizije preklapanja zauzeća predavača

3. Sve kolizije preklapanja zauzeća predavača (Voditi računa da između termina predavanja mora postojati 15 minuta pauze)
4. Sve kolizije preklapanja zauzeća predavača ako se razmatra i kartica SIT - Lozica, pri čemu put Novi Sad - Loznica traje 3 h

Za kreiranje zadatka koristiti matrice ili liste koje u sebi sadrže liste.

Sugestija: Vrste neka predstavlja dane u nedelji, dok će kolone predstavljati termin u rasporedu.

Termin u rasporedu mogao bi se opisati osmorkom u formatu: Kabinet, Predavac, odSati, odMinuta, doSati, doMinuta, Predmet, Mesto