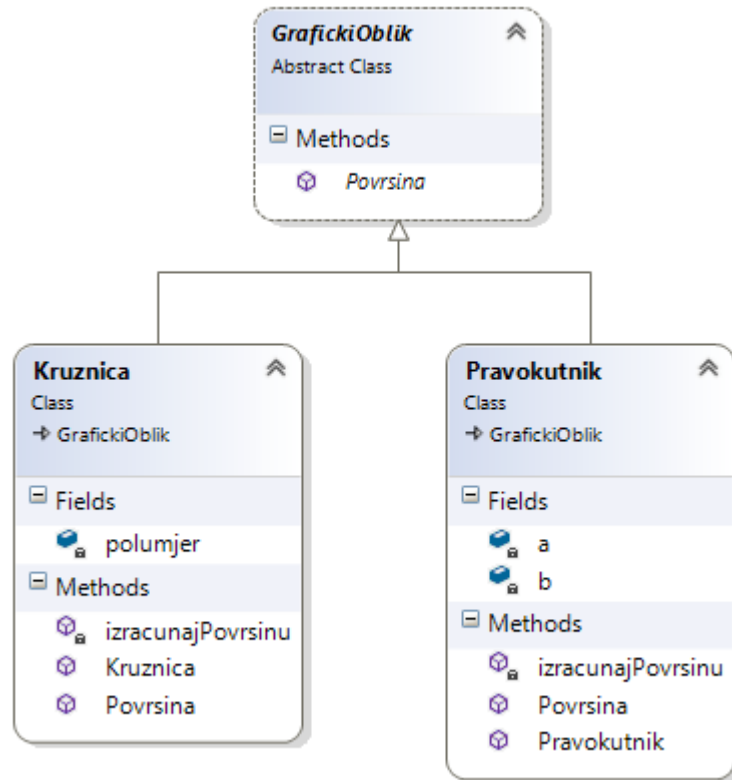


Ime i prezime studenta:		Matični broj:		Ukupno bodova	/100 A
-------------------------	--	---------------	--	---------------	---------------

1. [Najviše 25 bodova]

Razraditi razrede **GrafickiOblik**, **Kruznica** i **Pravokutnik** tako da je razreda **GrafickiOblik** bazni razred za ostala dva. U razredu **GrafickiOblik** deklarirati apstraktnu metodu naziva „Povrsina“ koja će biti implementirana u razredima **Kruznica** i **Pravokutnik**. Pokazati znanje implementiranja nasljeđivanja i dinamičkog povezivanja.



Koristeći programski jezik C# i razvojnu okolinu Visual Studio izradite projekt, konzolnu aplikaciju koja sadržava:

- 1.1. Razred **GrafickiOblik**. Razred **GrafickiOblik** biti će bazni razreda u stablu nasljeđivanja, te se sastoji od:
 - 1.1.1. Metode dostupne izvan razreda, naziva „Povrsina“ koja daje vrijednost tipa „double“. Metoda ne prima parametre, njezinim pokretanjem izračunava se površina grafičkog oblika. U razredu **GrafickiOblik** metoda nije implementirana, nema tijelo. Dodatni Bodovi se dodjeljuju ukoliko se razred i metoda implementiraju ako apstraktni.
- 1.2. Razred **Kruznica** koji je naslijeđen od razreda od **GrafickiOblik** i sastoji se od:
 - 1.2.1. Privatnog podatkovnog člana „polumjer“ tipa double.
 - 1.2.2. Onemogućiti stvaranje objekta pomoću podrazumijevanog konstruktora.
 - 1.2.3. Stvoriti konstruktor koji prima polumjer kao parametar.
 - 1.2.4. Izraditi privatnu metodu „izracunajPovrsinu“ kojom se izračunava površina.
 - 1.2.5. Izraditi javnu metodu „Povrsina“ koja daje površinu (vraća podatak izračunat prethodno izrađenom metodom). Osigurati dinamičko povezivanje za metodu.
- 1.3. Razred **Pravokutnik** koji je naslijeđen od razreda **GrafickiOblik** koji se sastoji od:
 - 1.3.1. Privatnih podatkovnih članova za pohranu dimenzija pravokutnika
 - 1.3.2. Konstruktora kojim se upisuju podaci o dimenzijama (onemogućiti stvaranje pravokutnika pomoću podrazumijevanog konstruktora).
 - 1.3.3. Privatne metode „izracunajPovrsinu“ kojom se izračunava površina pravokutnika, povratni tip je „double“
 - 1.3.4. Izraditi javnu metodu „Povrsina“ koja daje površinu (vraća podatak izračunat prethodno izrađenom metodom). Osigurati dinamičko povezivanje za metodu.
- 1.4. Glavni dio programa u kojem je potrebno implementirati:

- 1.4.1. Listu naziva „oblici“ koja prima tip podatka GrafickiOblik u koju se pohranjuju objekti razreda Kriznica i Pravokutnik.
- 1.4.2. Stvoriti tri kružnice i tri pravokutnika, te u jednoj „foreach“ petlji izračunati njihovu ukupnu površinu i ispisati je na zaslone.

[Najviše 25 bodova]

2. Koristeći programski jezik C# i razvojnu okolinu Visual Studio nadogradite projekt iz prethodnog zadatka na način da se doda razred **Proizvod** i razred **Tvornica**.
 - 2.1. Razred **Proizvod** izraditi na ovaj način:
 - 2.1.1. Onemogućiti stvaranje objekta podrazumijevanim konstruktorom.
 - 2.1.2. Implementirati javno svojstvo Naziv tipa string. Omogućiti samo dohvat podataka za svojstvo, ali ne i postavljanje vrijednosti.
 - 2.1.3. Implementirati javno svojstvo Cijena tipa int. Omogućiti samo dohvat podataka za svojstvo, ali ne i postavljanje vrijednosti.
 - 2.1.4. Implementirati jedan konstruktor koji kao parametre prima naziv proizvoda (string) i njegovu cijenu (int).
 - 2.1.5. Implementirati sučelje IComparable za redanje (sortiranje) proizvoda po cijeni, rastuće.
 - 2.1.6. Implementirati metodu „string ToString()“ kojom se ispisuje: "Proizvod: " + Naziv + " Cijena: " + Cijena. Osigurati dinamičko povezivanje za metodu.
 - 2.2. Razred **Tvornica** izraditi na ovaj način:
 - 2.2.1. Definirati privatnu varijablu tipa „int“ naziva „NAJVECI_KAPACITET“ u koju se pohranjuje podatak o najvećem kapacitetu tvornice. Dodatni bod: učiniti varijablu statičkom i dozvoliti samo čitanje za varijablu.
 - 2.2.2. Definirati privatnu varijablu tipa „int“ naziva „kapacitet“ za pohranu kapaciteta tvornice.
 - 2.2.3. Definirati privatnu varijablu tipa „string“ naziva „ime“ za pohranu naziva tvornice.
 - 2.2.4. Deklarirati i stvoriti listu naziva „proizvodi“ u koju se pohranjuju objekti tipa **Proizvod**.
 - 2.2.5. Onemogućiti stvaranje objekta tipa Tvornica pomoću podrazumijevanog konstruktora.
 - 2.2.6. Stvoriti konstruktor koji prima parametre: „kapacitet“ tipa int i „naziv“ tipa „string“. Pohraniti primljene parametre u privatne podatkovne članove te prethodno provjeriti:
 - 2.2.6.1. Da li je primljeni kapacitet veći od vrijednosti u NAJVECI_KAPACITET te ukoliko je stvoriti i „baciti“ iznimku tipa „ArgumentOutOfRangeException“ s porukom „Najveći dopušteni kapacitet je premašen“.
 - 2.2.6.2. Da li je podatak o nazivu tvornice prazan tj. ima duljinu niza nula. Ukoliko je stvoriti i „baciti“ iznimku tipa ArgumentException s porukom „Naziv tvornice ne može biti prazan niz“.
 - 2.2.7. Stvoriti metodu „proizvedi“ koja daje podatak tipa **Proizvod** a prima naziv i cijenu, prototip metode je: „public Proizvod proizvedi(string naziv, int cijena)“. Metoda stvara novi proizvod pomoću predanog naziva i cijene te ga dodaje u listu „proizvodi“ (točka zadatka 2.2.4.). Na kraju vraća stvoreni proizvod pomoću naredbe „return“.
 - 2.2.8. Stvoriti javnu metodu naziva „najsкупljі“ koja vraća podatak tipa **Proizvod**. Podatak koji se vraća mora biti najskuplji proizvod iz liste svih proizvoda. Pomoć: Razred Proizvod ima implementirano sučelje IComparable pa kako bi se dobio najskuplji proizvod potrebno je sortirati listu svih proizvoda te dohvatiti prvi ili zadnji, ovisno i načinu kako je izvedena metoda CompareTo u razredu **Proizvod**.
 - 2.2.9. Stvoriti javno svojstvo naziva „Najjeftiniji“ koja vraća podatak tipa **Proizvod**. Podatak koji se vraća mora biti najjeftiniji proizvod iz liste svih proizvoda. Vidi pomoć u prethodnoj točki.

[Najviše 25 bodova]

3. Koristeći programski jezik C# i razvojnu okolinu Visual Studio nadogradite projekt iz prethodnog zadatka na način da se doda razred **Osoba**. Ostvariti jednostruko povezanu listu razredom Osoba te ispisati podatke o svim članovima liste. Koristiti svojstvo jezika C# za referentne tipove pri čemu se može smatrati da identifikator referenca (uvjetno rečeno pokazivač).
 - 3.1. Izraditi razred **Osoba** koji sadrži:
 - 3.1.1. Javno svojstvo naziva „Slijedeca“ tipa „Osoba“, implementirati dohvat i postavljanje. Podrazumijevana vrijednost svojstva je „null“.

- 3.1.2. Javno svojstvo naziva „Ime“ tipa „string“, implementirati dohvat i postavljanje.
- 3.1.3. Onemogućiti stvaranje pomoću podrazumijevanog konstruktora.
- 3.1.4. Stvoriti konstruktor u kojem se predaje ime osobe te se pohranjuje u svojstvo „Ime“. U konstruktoru postaviti svojstvo „Slijedeca“ na vrijednost „null“.
- 3.2. U glavnom dijelu programa:
 - 3.2.1. Stvoriti četiri osobe s različitim imenima te ih povezati u listu. Pomoć: nakon prve osobe u listi je druga, treća itd. pa će tako prve dvije u listi biti povezane ovako: „prva.Slijedeca = druga“.
 - 3.2.2. Jednom „for“ petljom ispisati imena sve četiri osobe.

[Najviše 25 bodova]

- 4. Koristeći programski jezik C# i razvoju okolinu Visual Studio izradite razred **Valjak** kojim je ostvareno trodimenzionalno tijelo
 - 4.1. Tako da razred sadrži:
 - 4.1.1. Podatak polumjeru baze (float) koji nije vidljiv izvan razreda
 - 4.1.2. Podatak o visini (float) koji nije vidljiv izvan razreda
 - 4.1.3. Podrazumijevani konstruktor koji sve podatkovne članove postavlja na vrijednost „1“
 - 4.1.4. Konstruktor kojim se postavlja polumjer baze i visina
 - 4.1.5. Metodu za postavljanje polumjera baze naziva „postaviPolumjerBaze“
 - 4.1.6. Metodu za dohvat polumjera baze naziva „dohvatiPolumjerBaze“
 - 4.1.7. Metodu za postavljanje visine
 - 4.1.8. Metodu za dohvat visine
 - 4.1.9. Svojstvo PolumjerBaze
 - 4.1.10. Svojstvo Visina
 - 4.1.11. Metodu za izračun površine baze (PovrsinaBaze)
 - 4.1.12. Metodu za izračun površine plašta (PovrsinaPlasta)
 - 4.1.13. Svojstvo za izračun ukupne površine valjka (UkupnaPovrsina)
 - 4.1.14. Svojstvo za izračun volumena (Volumen)
 - 4.1.15. Metodu za ispis objekta (toString)
 - 4.2. Stvorenom projektu (solution) dodati programski kôd za testiranje funkcionalnosti razreda **Valjak**
 - 4.2.1. Ostvariti test metodu „ProvjeraPostavljanjaPolumjeraUKonstruktoru“ kojom se provjerava da li je vrijednost polumjera predana u konstruktoru jednaka onoj koja se dohvaća metodom „dohvatiPolumjerBaze“.
 - 4.2.2. Ostvariti test metodu „ProvjeraPostavljanjaPolumjeraMetodom“ kojom se provjerava da li je vrijednost polumjera baze predana metodom „postaviPolumjerBaze“ jednaka onoj koja se dohvaća metodom „dohvatiPolumjerBaze“.