

Ime i prezime studenta:		Matični broj:		Ukupno bodova	/50 A
-------------------------	--	---------------	--	---------------	--------------

1. [Najviše 9 bodova]

Koristeći programski jezik C# i razvojnu okolinu Visual Studio izradite projekt, konzolnu aplikaciju koja sadržava razred **Osoba**. Razred **Osoba** se sastoji od:

- 1.1. Privatnih podatkovnih članova OIB, Ime, Prezime. Odabrati odgovarajuće tipove podataka.
- 1.2. Onemogućiti stvaranje razreda pomoću podrazumijevanog konstruktora. Objekt razreda **Osoba** mora biti moguće stvoriti isključivo predajom tri parametra: OIB-a, imena i prezimena.
- 1.3. Izraditi metodu vidljivu isključivo u okviru razreda **Osoba** i njegovih naslijeđenih razreda naziva **provjeriOib**. Prije upisa podataka u privatni podatkovni član „oib“ razreda **Osoba** u konstruktoru je metodom potrebno provjeriti valjanost podatka na način:
 - 1.3.1. Predani podatak mora sadržavati isključivo brojeve. Ukoliko uvjet nije zadovoljen, grešku dojaviti iznimkom tipa `DataMisalignedException` porukom "Predani podatak sadržava znakove koji nisu brojevi".
 - 1.3.2. Predani podatak mora sadržavati točno 11 znakova. Ukoliko uvjet nije zadovoljen, grešku dojaviti iznimkom tipa `ArgumentOutOfRangeException` s porukom "Predani podatak sadržava previše ili premalo znakova". Ukoliko je uvjet zadovoljen metoda ne baca iznimku te daje predani podatak kao povratnu vrijednost koja se operatorom pridruživanja vrijednost upisuje u podatkovni član „oib“. (`this.oib = provjeriOib(oib);`).
- 1.4. Javne metode naziva „ispisiPodatke“ koja daje znakovni niz na početku kojeg stoji „Osoba – „, te nakon toga slijede podaci osobe, oib, prezime i ime. Metoda sama ne ispisuje na konzolu, NE KORISTI `Console.WriteLine` već vraća podatak tipa `string`.
- 1.5. Implementiranog sučelja `IEquatable`. Omogućiti provjeru istovjetnosti dva objekta razreda **Osoba** usporedbom prezimena i imena.
- 1.6. U glavnom dijelu programa izraditi programski kôd:
 - 1.6.1. Izraditi programski kod koji će nakon stvaranja razreda **Osoba** provjeriti valjanost upisanog OIB-a te obraditi iznimke koje s pri tome mogu pojaviti. Ukoliko je OIB valjan, u glavnom dijelu programa se ispisuje "Podatak je dobro upisan". Ukoliko OIB nije valjan, u glavnom dijelu programa ispisati "Greška pri unosu OIB-a: " te detalje o grešci koji sadržavaju poruku iznimke koje se pri tome pojavila. Izraditi primjere za sva tri slučaja: i) OIB je valjan, ii) broj znakova OIB-a je različit od 11, iii) OIB nije sačinjen isključivo od brojki.
 - 1.6.2. Izraditi programski kôd kojim se stvaraju još dva objekta razreda **osoba** te provjeriti da li osobe međusobno imaju isto ime i prezime, izraditi programski kôd koji prikazuje oba slučaja.

[Najviše 11 bodova]

2. Koristeći programski jezik C# i razvojnu okolinu Visual Studio nadogradite projekt iz prethodnog zadatka na način da se doda razred **Student** kojem je razred **Osoba** bazni razred:

- 2.1. Podesiti podatkovne članove razreda **Osoba** tako da su vidljivi u naslijeđenom razredu, no nisu vidljivi javno.
- 2.2. Korištenjem jednodimenzionalnih polja omogućiti za svaki primjerak razreda **Student** pohranjivanje do 5 parova podataka naziva predmeta i ocjene predmeta. Koristiti jednodimenzionalno polje naziva „nazivPredmeta“ te jednodimenzionalno polje „ocjena“. Odabrati odgovarajuće tipove podataka.
 - 2.2.1. Tip podatka za polje „ocjena“ mora biti odabran tako da je numerički uz uvjet potrošnje najmanje količine radne memorije.
 - 2.2.2. Par podataka (NazivPredmeta, Ocjena) za studenta se u jednodimenzionalna polje pohranjuje pomoću metode naziva **dodajOcjenuZaPredmet** kojoj se predaje naziv predmeta i ocjena.
 - 2.2.3. Metoda **dodajOcjenuZaPredmet** provjerava da li je broj upisanih ocjena prešao dozvoljeni kapacitet od 5 ocjena. Ukoliko je uvjet zadovoljen, metoda grešku dojavljuje iznimkom `IndexOutOfRangeException` uz poruku "Nije moguće dodati nove ocjene."
 - 2.2.4. Metoda **dodajOcjenuZaPredmet** provjerava da li je numerička ocjena koja joj se predaje u dozvoljenom opsegu (1-5). Ukoliko uvjet nije zadovoljen, metoda grešku dojavljuje iznimkom `ArgumentOutOfRangeException` uz poruku "Podatak za ocjenu nije valjan".

- 2.3. Izraditi svojstvo naziva **ProsjeKOcjena** koje daje prosjek ocjena pojedinog studenta.
- 2.4. Izraditi metodu **ispisiPodatke** te za metodu o razredima **Osoba** i **Student** omogućiti mehanizam dinamičkog povezivanja. Metoda **ispisiPodatke** razreda student daje znakovni niz koji počinje s „Student – OIB: „ te se nastavlja s podacima prezime, ime i prosjek ocjena. Metoda sama ne ispisuje na konzolu, NE KORISTI Console.WriteLine već vraća podatak tipa string.
- 2.5. Implementirati sučelje **IComparable** te omogućiti uvođenje poretka među studente prema prosjeku, koristiti svojstvo **ProsjeKOcjena**.
- 2.6. U glavnom dijelu programa:
 - 2.6.1. Izraditi programski kôd kojim se stvaraju najmanje tri objekta razreda student, svakom pridijeliti različita imena i prezimena te dodati najmanje tri predmeta s ocjenama.
 - 2.6.2. Objekte razreda student dodati u listu, ispisati listu, sortirati listu metodom **List<T>.Sort** te je ponovno ispisati sortiranu.

[Najviše 10 bodova]

3. U glavnom dijelu programa u funkciji **Main**:
 - 3.1. Programskim kôdom prikazati mehanizam dinamičkog povezivanja na primjeru razreda **Osoba** i **Student** korištenjem metode **ispisiPodatke**. Izraditi primjer pozivanja metode iz jednog i drugog razreda te prikazati primjer stvaranja objekta kojem je tip baznog razreda a objekt naslijeđenog razreda te prikazati poziv metode **ispisiPodatke**. Objasnite u komentaru programskog kôda koja metoda (iz koleg razreda) se poziva te koje je ključne riječi i gdje potrebno primijeniti da bi se omogućio mehanizam dinamičkog/kasnog povezivanja.

[Najviše 20 bodova]

4. Projektu stvorenom u 1., 2. i 3. zadatku dodati programski kôd za testiranje funkcionalnosti, tzv. Unit Test. Stvoriti ove testove kojima se ispituje (paziti da razredi **Osoba** i **Student** imaju modifikatore pristupa postavljene tako da su dostupni u tzv. Unit Testu.

[Najviše 5 bodova] za svaku pod-točku, ukupno najviše 20]

- 4.1. Za razred **Osoba** ispitati:
 - 4.1.1. Ispravnost usporedbe imena i prezimena metodom Equals kada su podaci jednaki, test metodu nazvati „UsporedbaOsobaPolmenuPrezimenuPozitivna“, test ima pozitivan ishod ukoliko metoda Equals daje vrijednost „true“ za dva objekta razreda Osoba koji imaju isto ime i prezime.
 - 4.1.2. Ispravnost usporedbe imena i prezimena metodom Equals kada podaci nisu jednaki. Test metodu nazvati „UsporedbaOsobaPolmenuPrezimenuNegativna“, test ima pozitivan ishod ukoliko metoda Equals daje vrijednost „false“ za dva objekta razreda Osoba koji nemaju isto ime i prezime.
- 4.2. Za razred **Student** ispisati:
 - 4.2.1. Provjeriti stvaranje/bacanje iznimke u slučaju kada se studentu dodjeljuje ocjena koja nije valjana, test metodu nazvati „StudentIznimkaZaOcjenuKojaNijeValjanja“. Test ima pozitivan ishod ukoliko se iznimka pokrene. Ukoliko se iznimka ne pokrene, pozvati metodu Assert.Fail().
 - 4.2.2. Provjeriti stvaranje/bacanje iznimke u slučaju kada je premašen kapacitet za dodavanje predmeta studentu, metodu test nazvati „StudentIznimkaZaKapacitetDodavanjaPredemta“. Test ima pozitivan ishod ukoliko se iznimka pokrene. Ukoliko se iznimka ne pokrene, pozvati metodu Assert.Fail().

Minimum potreban za usmeni ispit je 25 bodova.