Što je i čemu služi apstraktna klasa?

nekad želimo napraviti klasu koja je toliko „općenita”, da uopće nemamo namjeru stvarati instancu te klase ona nam samo služi kao zajednička podloga za stvaranje izvedenih klasa takvu klasu nazivamo apstraktna (abstract) apstraktnu klasu ne možete instancirati, ali je od nje moguće nasljeđivati osim toga, nema razlike između apstraktnih i „običnih” klasa mogu sadržavati podatkovna polja i metode uglavnom sadrži apstraktne metode (ali nije nužno). apstraktne metode ne sadrže nikakve naredbe (tijelo) prilikom deklaracije apstraktne metode navodi se ključna riječ abstract, tip, ime i parametri nije potrebno niti pisati zagrade { i } za tijelo metode:

Objasnite razliku između overload/override

Overload = Metode istog imena različitog potpisa.

Override= Ponekad nam neke metode ili svojstva (eng. property) izvorne klase nisu prikladne za upotrebu u izvedenoj klasi. Na primjer, u izvornoj klasi je definirana metoda u kojoj se vrši ispis mjesta prebivališta (npr. Split), a u izvedenoj klasi se također želi ispisati mjesto prebivališta, s tim da se pored mjesta prebivališta ispisuje još ulica i broj. Tada se koristi premošćivanje (eng. override) kako bi se izmijenila metoda iz izvorne klase i prilagodila potrebama korisnika.

Koji se modifikator pristupa koristi za metodu unutar sučelja?

sučelje je skup apstraktnih metoda koje možemo uključiti u našu klasu, te napraviti implementaciju tih metoda sučelje je slično apstraktnoj klasi, sa razlikom da apstraktna klasa može imati metode koje nisu apstraktne dok su u sučelju sve metode apstraktne deklarira se slično kao klasa

Može li klasa nasljediti konstruktor svoje bazne klase?

Podklasa nasljeđuje sve članove (članske varijable, metode i ugniježđene klase) svoje nadklase. Ne može pristupiti članovima nadklase koji su označeni s private. Može (naravno) javnim i onima koji su označeni s protected. - podsjetnik: članovi bez modifikatora dostupni su ako su klase u istom paketu. **Konstruktori nisu članovi**, pa se ne nasljeđuju**, ali se iz izvedene klase može pozvati konstruktor bazne klase.**

Koje su prednosti korištenja gettera i settera umjesto da koristimo public za ta svojstva (varijable)?