

Analiza sistema

Skola

Atributi:

- Ucenici: List(Ucenik)
- Roditelji: List(Roditelj)
- Odjeljenja: List(Odjeljenje)
- Nastavnici: List(Nastavnik)
- Predmeti: List(Predmet)
- Administrator: Administrator

Metode:

- Konstruktori
- get/set metode
- dodajUcenika(ucenik: Ucenik):void
- obrisiUcenika(ucenik: Ucenik):void
- azurirajUcenika(ucenik: Ucenik):void
- dodajRoditelja(roditelj: Roditelj):void
- azurirajRoditelja(roditelj: Roditelj):void
- obrisiRoditelja(roditelj: Roditelj):void
- dodajNastavnika(nastavnik: Nastavnik):void
- obrisiNastavnika(nastavnik: Nastavnik):void
- dodajOdjeljene(odjeljenje: Odjeljenje):void
- obrisiOdjeljenje(odjeljenje: Odjeljenje):void
- azurirajOdjeljenje(odjeljenje: Odjeljenje):void
- dodajPredmet(String predmet):void
- obrisiPredmet(String predmet):void

Osoba je apstraktna klasa iz koje su izvedeni Ucenik, Roditelj, Nastavnik, i Administrator.

Atributi:

- id: int
- ime: String
- prezime: String
- datumRodjenja: Date
- mjestoRodjenja: String
- jmbg: String
- adresa: String
- kontakt: String
- korisnickoIme: String
- sifra: String

Metode:

- Konstruktori
- get/set metode
- (-)generisiSifru():void

Ucenik

Atributi:

- roditelj:Roditelj
- Ocjene:List(Ocjene)
- Izostanci:List(Izostanak)
- Vladanje:Vladanje
- ProsjekNaPolugodistu:double
- ProsjekNaKrajuGodine:double
- zakljucneOcjenePrvoPolugodiste:Mapa(Predmet, int)
- zakljucneOcjeneDrugooPolugodiste:Mapa(Predmet, int)

Metode:

- konstruktor
- get/set metode
- dajOcjeneZaPredmet(predmet:Predmet):List(Ocjena)
- dajOpravdaneIzostanke():void
- dajNeopravdaneIzostanke():void

Nastavnik

Atributi:

- predmeti:List(Predmet)

Metode:

- konstruktor
- get/set metode
- upisiOcjenu(ucenik:Ucenik, predmet:Predmet,vrijednost:int, tip:tipOcjene, datum:Date):void
- napisiNapomenu(datum:Date, napomena:String, odjeljenje:Odjeljenje):void
- zakaziProvjeruZnanja(tip:tipProvjereZnanja, datum:Date, predmet: Predmet, odjeljenje:Odjeljenje):void
- otkaziProvjeruZnanjat(provjeraZnanja, ProvjeraZnanja):void
- upisiCas(odjeljenje:Odjeljenje, tematskaJedinica:String, tip:tipCasa, redniBrojUGodini:int, redniBrojUDanu:int):void
- upisiIzostanak(ucenik: Ucenik, predmet:Predmet, datum: Date):void
- zakljuciOcjenu(ucenik: Ucenik, predmet:Predmet, ocjena:int)

Razrednik

Atributi:

Metode:

- Konstruktor
- get/set metode
- opravdajIzostanak(učenik:Učenik, izostanak: Izostanak):void
- napisiIzvjestajSaRoditeljskogSastanka(datum: Date, tema: String, zakljucak: String):void
- evidentirajPrisustvo (datum: Date, Roditelj roditelj):void
- izracunajOpciUspjeh(učenik: Učenik)
- promjeniVladanje(učenik:Učenik, vladanje)

Administrator

Atributi:

Metode:

- konstruktor

Roditelj

Atributi:

Metode:

- Konstruktori
- prisustvoRoditelja

Razred

Atributi:

- Predmeti:List(Predmet)

Metode:

- Konstruktor(predmeti:List(Predmet))
- get/set metode
- dodajPredmet(predmet:Predmet)
- obrisiPredmet(predmet:Predmet)

Odjeljenje

Atributi:

- razred:Razred
- razrednik:Razrednik
- ucenici:List(Ucenik)
- napomena:List(Napomena)
- održaniCasovi:List(Cas)
- raspored:Mapa(Dan, List(Predmet))
- roditeljskiSastanci:List(RoditeljskiSastanak)
- zakazaneProvjereZnanja:List(ProvjeraZnanja)

Metode:

- konstruktor
- get/set metode
- dodajCas(cas: Cas)
- zakaziProvjeru(provjera: ProvjeraZnanja)
- otkaziProvjeru(provjera: ProvjeraZnanja)
- dodajNapomenu(napomena: Napomena)
- dodajUcenik(ucenik:Ucenik)
- dodajRoditeljskiSastanak(sastanak:RoditeljskiSastanak)

Ocjena

Atributi:

- ucenik:Ucenik
- predmet:Predmet
- vrijednost:int
- datum:Date
- tip:tipOcjene

Metode:

- Konstruktor
- get/set metode

Izostanak

Atributi:

- ucenik: Ucenik
- predmet: Predmet
- datum: Date
- statusIzostanka:StatusIzostanka

- razlog: String

Metode:

- konstruktor
- get/set metode

Cas

Atributi:

- tematskaJedinica:String
- tip:tipCasa
- datum:Date
- brojCasaUGodini:int
- brojCasaUDanu:int

Metode:

- konstruktori
- get/set metode

ProvjeraZnanja

Atributi:

- tip:tipProvjereZnanja
- datum:Date
- predmet:Predmet

Metode:

- konstruktori
- get/set metode

RoditeljskiSastanak

Atributi:

- tema:String
- datum:Date
- zakljucak:String

Metode:

- konstruktor
- get/set metode

Napomena

Atributi:

- datum:Date
- opis:String

Metode:

- konstruktor
- get/set metode

Enum **tipCasa** { OBRADA, UTVRDIJIVANJE, PONAVLJANJE,PROVJERA }

Enum **tipOcjene**{ PZ, AKTIVNOST, USMENI,TEST,LEKTIRA }

Enum **statusIzostanka**{ OPRAVDANO, NEOPRAVDANO }

Enum **vladanje** { PRIMJERNO, VRLODOBRO, DOBRO,ZADOVOLJAVA, LOŠE }

Enum **tipProvjereZnanja** { TEST, PISMENA ZADAĆA }

Enum **dan**{ Ponedjeljak, Utorak, Srijeda, Četvrtak, Petak }

Veze:

NASTAVNIK-RAZREDNIK:generalizacija, jer je razrednik izveden iz klase nastavnik

OSOBA-UCENIK,ADMINISTRATOR,RODITELJ,NASTAVNIK:generalizacija, jer su razrednik, roditelj, ucenik i administratornizvedeni iz klase osoba

ODJELJENJE-RAZREDNIK:generalizacija, jer je klasa odjeljenje izvedena iz klase razred

RAZREDNIK-ODJELJENJE:kompozicija, jer odjeljenje ne može postojati bez razrednika

UCENIK-ODJELJENJE:kompozicija, jer odjeljenje ne može postojati bez učenika

PREDMET-RAZRED:kompozicija, jer razred ne može postojati bez predmeta

IZOSTANAK-UCENIK:kompozicija, jer izostanak ne može postojati bez učenika

IZOSTANAK-PREDMET:kompozicija, jer izostanak ne može postojati bez predmeta

OCJENA-UCENIKA:kompozicija, jer ocjena ne može postojati bez učenika

OCJENA-PREDMET:kompozicija, jer ocjena ne može postojati bez predmeta

RODITELJSKISASTANAK-ODJELJENJE:agregacija, jer brisanjem roditeljskog sastanka, odjeljenje idalje postoji

NAPOMENA-ODJELJENJE:agregacija, jer brisanjem napomene, odjeljenje idalje postoji

RODITELJ-UCENIK-agregacija, jer brisanjem roditelja učenik idalje postoji

NASTAVNIK-PREDMET:asocijacija, nastavnik predaje jedan ili više predmeta

Solid principi

1. Single Responsibility Principle:

Ovaj princip nije narušen s obzirom da svaka klasa ima zasebnu funkciju, odnosno više klasa nema iste funkcionalnosti.

2. Open-Closed Principle:

Apstraktna klasa Osoba daje mogućnost da se iz nje naslijede još neke klase koje će predstavljati nove vrste korisnika sistema bez potreba da se kod modificira. Prema tome, ovaj princip nije narušen.

3. Liskov Substitution Principle:

Svaki roditelj, učenik, administrator i nastavnik su ujedno i osobe, tako da se mogu koristiti u kontekstu klase iz koje su naslijeđene. Također, klasa Razrednik je naslijeđena iz klase Nastavnik, a svaki razrednik je ujedno i nastavnik. Dakle, ovaj princip nije narušen.

4. Interface Segregation Principle:

U našem modelu nemamo nijednog interfejsa, tako da ovaj princip nije narušen.

5. Dependency Inversion Principle:

U našem modelu imamo klasu Osoba iz koje su naslijeđene klase Učenik, Roditelj, Administrator i Nastavnik. Također, klasa Razrednik je naslijeđena iz klase Nastavnik. Naslijeđene klase zavise od nadređene, prema tome ovaj princip je zadovoljen.