



Kolokvijum 2

Koncert Evergreen

Napomene:

1. Obavezno pročitati **SVE** napomene.
2. Polja ne smeju biti public vidljivosti.
3. Napisati test program i u njemu testirati svaki implementirani konstruktor, metod i funkciju. **Ono što nije testirano (u main() funkciji) se ne boduje.**
4. Program se mora uspešno kompajlirati, u suprotnom možete dobiti najviše 1 bod.
5. Sve metode koje imaju `bool` povratnu vrednost vraćaju informaciju o uspešnosti metode u zavisnosti od uslova u zadatku.
6. Obezbediti da geteri ne mogu da promene vrednost polja.
7. Asistent za vreme testa daje informacije vezane isključivo za razumevanje teksta zadatka.
8. Naredba za aktiviranje latinične verzije srpskog rasporeda tastature: `setxkbmap rs latin`.
9. Za generisanje slučajne vrednosti iskoristiti `rand()` funkciju koja se nalazi u C++ zaglavlju ``.
10. Za olakšano testiranje programa, koristite iste nazive metoda, klasa i polja i koristite main.cpp iz rešenja.

U gimnaziji Jovan Jovanović Zmaj svake godine održava se čuveni koncert pod nazivom Evergreen. Ovo nije samo običan koncert – to je trenutak kada se učenici, maturanti, bivši maturanti, profesori, prijatelji i gosti okupe kako bi zajedno uživali u melodijama koje slave ljubav, mladost i sećanja. Da biste pomogli da ova tradicija traje, potrebno je osmisliti softver za organizaciju Evergreen koncerta. Profesor Travica, poznat po svojoj posvećenosti i strasti prema muzici, svake godine marljivo prikuplja informacije o učenicima, njihovim veštinama, i pesmama koje žele da izvedu. Koristeći svoj čuveni, tajanstveni "Evergreen algoritam", profesor sastavlja savršen program koncerta – odabirom izvođača i pesama koje će osvojiti srca publike. Ovaj sistem treba da čuva duh ljubavi i zajedništva, koji Evergreen čini nezaboravnim događajem svake godine.

Napisati klasu **Učenik** koja ima sledeća polja:

- ime (tipa DinString)
- prezime (tipa DinString)
- pol (tipa boolean) gde False (0) označava muški a True (1) označava ženski pol
- godina (tipa int, maksimalna vrednost 4, minimalna 1)

Za klasu **Učenik** potrebno je implementirati sledeće metode:

- Konstruktor bez parametara
- Konstruktor sa parametrima
 - **Učenik(const DinString&, const DinString&, bool, int)**
- Konstruktor sa parametrima
 - **Učenik(const char*, const char*, bool, int)**
- **GET** metode za sva polja i **SET** metodu za polje godina
- Preklopiti operator ispisa <<, tako da ispisuje sve informacije o objektu. Polja nabrojivog tipa ispisati rečima.

Iz klase **Učenik** izvesti klasu **UčenikHorista** koja ima dodatna polja:

- mentor (tipa DinString) - označava puno ime i prezime mentora, podrazumevana vrednost je "Jovan Travica"
- tipHoriste (tipa TipHoriste) : TipHoriste može imati sledeće vrednosti (podrazumevana je NEPOZNATO)
 - NEPOZNATO, BASS, TENOR, SOPRAN, ORKESTAR
- iskustvo (tipa int) - označava broj godina koliko je neko dugo u horu
- counter (tipa static int) - označava ukupan broj učenika horista

Za klasu **UčenikHorista** realizovati sledeće metode:

- Konstruktor bez parametara
- Konstruktor kopije
- Konstruktor sa parametrima
 - **UčenikHorista(const DinString&, const DinString&, bool, int, const DinString&, TipHosriste, int);** gde je prvi parametar ime, drugi prezime, treći pol, četvrti godina, peti ime mentora, šesti tip hosite, i sedmi iskustvo
- Konstruktor sa parametrima
 - **UčenikHorista(const char*, const char*, bool, int, const char*, TipHosriste, int);** gde je prvi parametar ime, drugi prezime, treći pol, četvrti godina, peti ime mentora, šesti tip hosite, i sedmi iskustvo
- Destruktor
- Preklopiti operator << za ispis svih polja koja nisu statička
- **GET** metode za sva polja i **SET** metode za sva ne statička polja

Napisati klasu **Pesma** koja sadrži sledeća polja:

- ime (tipa DinString)
- izvodjac (tipa UcenikHorista)
- autor (tipa DinString) - podrazumeva puno ime i prezime autora ili puno ime grupe
- zanr (tipa ZanrPesme) : ZanrPesme može imati sledeće vrednosti (podrazumevana vrednost je NEPOZNATO)
 - NEPOZNATO, POP, ROK, REP, JAZZ, BLUES, REGGAE, TURBOFOLK, HIPHOP
- duzina (tipa int) - dužina trajanja pesme u sekundama
- brojPesama (tipa static int) - ukupan broj pesama

Za klasu **Pesma** realizovati sledeće metode:

- Konstruktor bez parametara
- Konstruktor kopije
- Konstruktor sa parametrima
 - **Pesma(const DinString&, const UcenikHorista&, const DinString&, ZanrPesme, int);** gde je prvi parametar ime, drugi autor, treći žanr pesme i četvrti dužina pesme
- Konstruktor sa parametrima
 - **Pesma(const char*, const UcenikHorista&, const char*, ZanrPesme, int);** gde je prvi parametar ime, drugi horista koji izvodi pesmu, treći autor, četvrti žanr pesme i peti dužina pesme
- Destruktor
- Virtuelni metod **prikaziDetalje()** koji ispisuje detalje o pesmi, ali pri tome formatira duzinu trajanja pesme u formatu **XX:XX (MINUTE:SEKUNDE)**. Kao return vrednost vraća string sa svim podacima o pesmi sa novim redovima tako da kada se string ispiše ispis bude čitak i lep.
- Preklopiti operator << za ispis svih polja koja nisu statička
- **GET** metode za sva polja

Napisati klasu **Violina** koja sadrži sledeća polja:

- tipGudala (tipa TipGudala) : TipGudala može imati sledeće vrednosti (podrazumevana vrednost je MODERNO)
 - BAROKNO, KLASIČNO, MODERNO
- brojViolina (tipa static int) - ukupan broj violina
- stanje (tipa StanjeViole) : StanjeViole može imati sledeće vrednosti (podrazumevana vrednost je SPREMNA)
 - RAŠTIMOVANA, SPREMNA
- godinaProizvodnje (tipa int) - podrazumevana vrednost je 0

Za klasu **Violina** realizovati sledeće metode:

- Konstruktor bez parametara
- Konstruktor kopije
- Konstruktor sa parametrima
 - **Violina(TipGudala, int, StanjeViole);** gde je prvi parametar tip gudala, drugi godina proizvodnje a treći stanje violine
- Destruktor
- Preklopiti operator << za ispis svih polja koja nisu statička

- **GET** metode za sva polja
- Metod **bool nastimajViolinu()**; koji ukoliko je violina u stanju **RAŠTIMOVANA** menja njeno stanje u **SPREMNA** i kao povratnu vrednost vraća da li je izmena uspešna
- Metod **bool rastimajViolinu()**; koji ukoliko je violina u stanju **SPREMNA** menja njeno stanje u **RAŠTIMOVANA** i kao povratnu vrednost vraća da li je izmena uspešna

Iz klase **Instrument** izvesti klasu **Klavir** koja sadrži sledeća polja:

- **tipKlavira** (tipa **TipKlavira**) : **TipKlavira** može imati sledeće vrednosti (podrazumevana vrednost je **AKUSTIČNI**)
 - **AKUSTIČNI**, **DIGITALNI**, **KONCERTNI**
- **brojKlavira** (tipa **static int**) - ukupan broj klavira
- **stanje** (tipa **StanjeKlavira**) : **StanjeKlavira** može imati sledeće vrednosti (podrazumevana vrednost je **SPREMAN**)
 - **RAŠTIMOVAN**, **SPREMAN**
- **godinaProizvodnje** (tipa **int**) - podrazumevana vrednost je 0

Za klasu **Klavir** realizovati sledeće metode:

- Konstruktor bez parametara
- Konstruktor kopije
- Konstruktor sa parametrima
 - **Klavir(tipKlavira, int, StanjeKlavira)**; gde je prvi parametar tip klavira, drugi godina proizvodnje a treći stanje klavira
- Destruktor
- **GET** metode za sva polja
- Preklopiti operator **<<** za ispis svih polja koja nisu statička
- Metod **bool nastimajKlavir()**; koji ukoliko je klavir u stanju **RAŠTIMOVAN** menja njegovo stanje u **SPREMAN** i kao povratnu vrednost vraća da li je izmena uspešna
- Metod **bool rastimajKlavir()**; koji ukoliko je instrument u stanju **SPREMAN** menja njegovo stanje u **RAŠTIMOVAN** i kao povratnu vrednost vraća da li je izmena uspešna

Iz klase **UčenikHorista** izvesti generičku klasu **HoristaOrkestar<INSTRUMENT>** koja je određena instrumentom koji član hora i orkestra svira. Klasa treba da sadrži sledeća polja:

- **instrument** (tipa **INSTRUMENT**) - objekat instrument kojim član orkestra upravlja
- **stanje** (tipa **StanjeClanaOrkestra**) : **StanjeClanaOrkestra** može imati sledeće vrednosti (podrazumevana vrednost je **SPREMAN**)
 - **SPREMAN**, **BOLESTAN**, **MATURIRAO**
- **brojClanovaOrkestra** (tipa **static int**) - ukupan broj članova orkestra

Za generičku klasu **HoristaOrkestar<INSTRUMENT>** implementirati sledeće metode:

- Konstruktor bez parametara
- Konstruktor sa parametrima
 - **HoristaOrkestar(const DinString&, const DinString&, bool, int, const DinString&, TipHoriste, int, INSTRUMENT*, StanjeClanaOrkestra, INSTRUMENT*, StanjeClanaOrkestra)**; gde je prvi parametar ime, drugi prezime, treći pol, četvrti godina, peti ime mentora, šesti tip hosite, i sedmi iskustvo, osmi instrument od člana orkestra i deveti parametar je stanje člana orkestra (podrazumevana vrednost je **SPREMAN**)
- Konstruktor sa parametrima
 - **HoristaOrkestar(const char*, const char*, bool, int, const char*, TipHoriste, int, INSTRUMENT*, StanjeClanaOrkestra, INSTRUMENT*, StanjeClanaOrkestra)**; gde je prvi parametar ime, drugi prezime, treći pol, četvrti godina, peti ime mentora, šesti tip hosite, i sedmi iskustvo, osmi instrument od člana orkestra i deveti parametar je stanje člana orkestra (podrazumevana vrednost je **SPREMAN**)
- Konstruktor kopije
- Destruktor
- **GET** metode za sva polja
- Metod **bool dodeliInstrument(INSTRUMENT*)**; koji prima kao prvi parametar objekat klase **INSTRUMENT** i vezuje ga za člana orkestra

- Metod **bool ozdravi();** koji ukoliko je član orkestra u stanju **BOLESTAN** menja njegovo stanje u **SPREMAN** i kao povratnu vrednost vraća da li
- Metod **bool razboliSe();** koji ukoliko je član orkestra u stanju **SPREMAN** menja njegovo stanje u **BOLESTAN** i kao povratnu vrednost vraća da li je izmena uspešna
- Preklopiti operator **<<** za ispis svih polja koja nisu statička

Napisati klasu **Evergreen** koja ima sledeća polja:

- **clanoviOrkestraViolina** (tipa **List<HoristaOrkestar>**) - lista svih članova orkestra hora gimnazije koji sviraju violinu
- **clanoviOrkestraKlavis** (tipa **List<HoristaOrkestar>**) - lista svih članova orkestra hora gimnazije koji sviraju klavis
- **clanoviHora** (tipa **List<UcenikHorista>**) - lista svih članova hora gimnazije
- **pesme** (tipa **List<Pesma>**) - lista svih pesama
- **dirigent** (tipa **DinString**) - puno ime i prezime dirigenta (podrazumevana vrednost "Jovan Travica")
- **brojPosetioca** (tipa **int**)

Za klasu **Evergreen** implementirati sledeće metode:

- konstruktor bez parametara
- konstruktor sa parametrima
 - **Evergreen(const DinString&, int);** gde je prvi argument puno ime dirigenta a drugi argument očekivan broj posetilaca
- konstruktor sa parametrima
 - **Evergreen(const char*, int);** gde je prvi argument puno ime dirigenta a drugi argument očekivan broj posetilaca
- Konstruktor kopije
- **GET** metode za sva polja i **SET** metodu za dirigenta i broj posetilaca
- Metodu **bool dodajPesmu(Pesma&);** dodaje novu pesmu u listu pesama na kraj liste, ako ne postoji pesma sa istim imenom, vraća status da li je promena uspešna
- Metodu **bool dodajClanaHora(UcenikHorista&);** dodaje novog učenika horistu u listu članova hora na kraj liste, vraća status da li je promena uspešna
- Metodu **bool dodajClanaOrkestra(HoristaOrkestar&);** dodaje novog učesnika člana orkestra na početak liste, vraća status da li je promena uspešna
- Metodu **bool izbaciPesmu(DinString&);** izbacuje pesmu sa željenim imenom iz liste pesama, vraća status da li je promena uspešna
- Metodu **bool izbaciClanaHora(UcenikHorista&);** izbacuje člana horistu iz liste članova hora, vraća status da li je promena uspešna
- Metodu **bool izbaciClanaOrkestra(HoristaOrkestar&);** izbacuje člana orkestra iz liste članova orkestra, vraća status da li je promena uspešna
- Metodu **bool izbaciClanaHoraSaNajmanjiskustva();** izbacuje člana hora koji ima najmanji broj godina iskustva, prvog kojeg pronađe, vraća status da li je promena uspešna
- Metodu **HoristaOrkestar pretraziOrkestarPolmenu(DinString&);** kao argument prima ime člana orkestra, a kao povratnu vrednost vraća njegov objekat iz liste članova orkestra (obraditi obe liste, prvo violinu)
- Metodu **void evergreenAlgoritam();**
 - Ispisuje spisak svih pesama kao i njihovih izvođača i dužinu pesme (formatiranu) (npr. **Aleksandra Golić - Kiki Lesendrić - Te Noći Smo Se Potukli Zbog Nje - 4:12**)
 - Ispisuje ukupno procenjeno vreme trajanja koncerta, ova vrednost se dobija kao zbir trajanja svih pesama i između svake 2 pesme se računa pauza od 30 sekundi. Ukupno vreme ispisati u formatu **HH:MM:SS**
 - Ispisuje sva imena i prezimena svih članova hora i orkestra gimnazije (obraditi obe liste, prvo violinu), ispisati puno ime i prezime, tip ili ime instrumenta i ime mentora. Primer ispisa je dat u nastavku:

————— KONCERT EVERGREEN 2023 —————

- Procenjeno vreme trajanja koncerta je 01:15:49

- Ukupan broj posetilaca: 102

Repertoar:

1. Ana Meseldžija - ABBA - Gimme! Gimme! Gimme! - 4:52
2. ...

Članovi Hora:

1. Jovana Kalafatić - Sopran - Jovan Travica
2. ...

Članovi Orkestra:

1. Momir Carevic - Klavis - Jovan Travica
2. ...