Smijete koristiti pisane materijale ili vlastite bilješke spremljene u HOME direktoriju, webstranicu kolegija i linkove koji vode s nje.

Strogo je zabranjeno korištenje ChatGPT-a te će se isto provjeravati za svaki kod. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail, chat, messenger i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultirat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretipkavanje teksta zadataka.

Iznimno, studenti koji kolokvij pišu u čitaonici, smiju upaliti hotspot na mobitelu, nakon čega mobitel moraju okrenuti licem prema stolu na prikladnoj udaljenosti. Mobitel se nakon toga više ne smije dirati do kraja izvođenja kolokvija.

Prvi zadatak (15 bodova)

Implementirajte sljedeću hijerarhiju klasa:

- apstraktna klasa `Zaposlenik` predstavlja zaposlenika neke firme. Ova klasa treba imati člansku varijablu `std::string ime`.
- Klasa `Menadzer` nasljeđuje klasu `Zaposlenik`. Ima dodatne članske varijable `double Placa` i `double Bonus`.
- Klasa `Inzenjer` nasljeđuje klasu `Zaposlenik`. Ima dodatne članske varijable `double cijenaSat` i `int brojOdradenihSati`.
- klasa `Sales` nasljeđuje klasu `Menadzer`. Ima dodatnu člansku varijablu `double provizija`.

Implementirajte konstruktore bez parametara i konstruktore s parametrima za sve četiri klase. Konstruktori (sa i bez parametara) ispisiju `Stvoren <klasa>.\n`. Konstruktori s parametrima primaju sve članske varijable.

Implementirajte destruktore za sve četiri klase. Destruktori ispisuju `Unisten <klasa>.\n`. Osigurajte da se pozovu svi destruktori nadređenih klasa prilikom uništavanja izvedenih klasa.

Implementirajte člansku funkciju `double izracunajPlacu()` prema sljedećim pravilima:

- klasa `Zaposlenik` - čista virtualna funkcija;
- klasa `Menadzer` - vraća zbroj plaće i bonusa;
- klasa `Inzenjer` - vraća umnožak cijene po satu i broja odrađenih sati;
- klasa `Sales` - vraća zbroj menadžerske plaće i menadžerske plaće pomnožene s provizijom.

Implementirajte člansku funkciju `void info()` koja ispisuje na ekran stanje svih članskih varijabli za objekt klase `Zaposlenik`, `Menadzer`, `Inzenjer` odnosno `Sales`.

Implementirajte globalnu funkciju `void IspisPlaca(std::vector<Zaposlenik*> V)` koja ispisuje plaće svh objekata u vektoru V.

Napišite i glavni program koji poziva funkciju `IspisPlaca`. Vektor mora sadržavati barem 3 objekta (barem jedan od svake klase).

Sve varijable u klasama moraju biti `private` odnosno `protected`. Izbjegavajte dupliciranje dijelova koda u protivnom možete izgubiti bodove! Zabranjeno je koristiti globalne varijable i friend funkcije! Potrebno je pobrinuti se da se `const` koristi svugdje gdje je moguće. Smijete implementirati i dodatne članske funkcije i varijable, no i dalje sve članske varijable trebaju biti `private`. Osigurajte da oslobodite svu dinamički alociranu memoriju.

Svoje rješenje spremite u jednu datoteku `zaposlenici.cpp`.

IME I PREZIME: MATIČNI BROJ STUDENTA: BODOVI:

Drugi zadatak (20 bodova)

Implementirajte klasu `Slika` koja sadrži **privatne** varijable:

- `vector<vector<tuple<int,int,int>>> pikseli` - kvadratna matrica uređenih trojki cijelih brojeva iz segmenta $[0, 255]$. Svaki element matrice predstavlja jedan piksel slike, a uređena trojka je RGB vrijednost tog piksela (Red, Green, Blue - zadaje boju piksela).
- `int dim` - dimenzija matrice `pikseli`

Implementirajte konstruktor `Slika(vector<vector<tuple<int,int,int>>> V, int d)` koji stvara sliku dimenzije `d` s pikselima iz vektora `V`. Možete pretpostaviti da će dimenzija matrice `V` biti jednaka `d` te da će svi elementi uređenih trojki biti iz segmenta $[0, 255]$.

Implementirajte funktor `CrnoBijelo`. `Operator()` prima referencu na `Slika` te je pretvara u crno-bijelu sliku na način se da svaki piksel postavi na vrijednost (S, S, S) gdje je $S = \lfloor \frac{R+G+B}{3} \rfloor$, a (R, G, B) je trenutna vrijednost piksela.

Implementirajte klasu `Video` koja sadrži **privatne** varijable:

- `vector<Slika> slike` - niz slika istih dimenzija
- `int dim` - dimenzija slika u vektoru `slike`

Implementirajte konstruktor `Video(vector<Slika> V, int d)` koji stvara video sa slikama dimenzije `dim` iz vektora `V`. Možete pretpostaviti da će sve slike iz `V` biti dimenzije `dim`. Implementirajte i defaultni konstruktor koji stvara "prazan" video (vektor `slike` je prazan, `dim` se postavlja na 0).

Omogućite rad sljedećih operatora:

- `operator+` - od dva videa `v1` i `v2` istih dimenzija stvara se novi `Video` sa slikama iz `v1` i `v2` (prvo dolaze sve slike iz `v1` u poretku kao i u `v1`, a zatim dolaze sve slike iz `v2` u poretku kao u `v2`). Ako `v1` i `v2` nisu istih dimenzija, stvara se prazan video.
- `operator+=` - dodaje sliku na kraj videa na kojem je pozvan operator (ako je `v` video, a `s` je slika, tada operator `+=` koristimo kao `v+=s`). Ako slika nije odgovarajuće dimenzije, ne radi ništa. Omogućite ulančavanje.
- `operator-` (prefiks i postfiks) - izbacuje zadnju sliku iz videa na kojem je pozvan operator.
- `operator[]` - prima integer `i` te vraća referencu na sliku na `i`-tom indeksu u videu na kojem pozvan operator. Ako je `i` manji od 0, vraća se referenca na sliku na indeksu 0. Ako je `i` veći ili jednak broju slika, vraća se referenca na zadnju sliku.

Implementirajte generičku funkciju `primijeni` koja prima iteratore `s` i `e` na redom početak i kraj spremnika te unarni predikat `f`. Funkcija na svaki element unutar intervala $[\texttt{s}, \texttt{e}\rangle$ primijenjuje `f`.

Koristeći funkciju `primijeni` i funktor `CrnoBijelo`, klasi `Video` dodajte funkciju `void crnoBijeliSvijet()` koja video na kojem je pozvana funkcija pretvara u crno-bijeli video.

IME I PREZIME: MATIČNI BROJ STUDENTA: BODOVI:

Implementirajte globalnu funkciju `Video stvoriVideo(int dim)` koja od svih slika dimenzije `dim` u glavnom programu stvara video. Ako nema takvih slika, stvara se prazan video.

Napomene:

- Rješenje spremite u datoteku `video.cpp`.
- Zabranjeno je korištenje globalnih varijabli i funkcija iz zaglavlja `algorithm`.
- Smijete implementirate dodatne članske varijable, no i dalje sve članske varijable moraju biti privatne. Smijete implementirati dodatne funkcije (članske i globalne).
- Osigurajte da ne dolazi do kopiranja memorije prilikom prijenosa parametara funkcijama i operatorima (osim ako to nije nužno).
- Onemogućite mijenjanje svih objekata proslijeđenih funkcijama i operatorima ukoliko njihovo mijenjanje nije nužno za implementaciju navedenih funkcionalnosti.
- Nije potrebno pisati glavni program.