

PROGRAMSKI JEZICI 1 – 22.4.2016.

- ❶ (30 bodova) Definirati klasu **ITyped** koja ima javnu čistu virtuelnu metodu **getType()** koja vraća **char*** string koji predstavlja oznaku tipa (pri implementaciji u izvedenim klasama, treba da vraća string sa nazivom izvedene klase). Poznata je činjenica da će ova klasa biti korištena pri višestrukom izvođenju.

Definirati klasu **Exception** izvedenu iz klase **ITyped**. Ova klasa treba da bude konkretna, da sadrži podatak član koji predstavlja poruku, te da ju je moguće štampati na izlazni tok u obliku „tip izuzetka: poruka“, operatorom za štampanje na izlazni tok. Ovaj operator treba da poziva metodu **print(std::ostream&)** koja treba da bude dostupna za redefiniciju potomcima. Ova metoda pri štampanju treba da poziva metodu **getType()** iz natklase.

Definirati klasu **IndexOutOfBoundsException** naslijeđenu iz klase **Exception**. Njena metoda **getType()** treba da vraća string „*IndexOutOfBoundsException*“.

- ❷ (70 bodova) Definirati generičku klasu **Matrix<T>** naslijeđenu iz klase **ITyped**. Klasa **Matrix<T>** treba da bude konkretna. Ova klasa predstavlja dvodimenzionu matricu dimenzija $n \times m$ elemenata tipa **T**. Klasa treba da ima konstruktor koji prihvata njene dimezije, te odgovarajuće konstruktore kopije, pomjeranja, operatore dodjele (sa i bez pomjeranja), te odgovarajući destruktor. Omogućiti dalje nasljeđivanje ove klase. Potom:

- Implementirati operatore sabiranja, oduzimanja, množenja i dijeljenja matrica (po elementima, a ne matrično množenje). Ako se dimezije matrica ne slažu, baciti izuzetak **Exception** sa porukom „*Matrix sizes do not match.*“. Izbjeći dupliciranje koda korištenjem pomoćne funkcije i lambda operatora.
- Implementirati **operator()** koji prihvata dva argumenta koji predstavljaju indekse elementa u matrici. Ovaj operator treba da omogući pristup elementima matrice i njihovo mijenjanje. Ako je neki od indeksa van opsega matrice, baciti izuzetak **IndexOutOfBoundsException** sa porukom „*Matrix index out of bounds.*“).
- Obezbijediti štampanje matrice na izlazni tok operatorom šiftovanja. Ovaj operator treba da poziva odgovarajuću metodu **print(std::ostream&)** koja treba da bude dostupna potomcima klase **Matrix<T>**.
- Implementirati poseban konstruktor za slučaj da je **T** tip **int**. U ovom slučaju, potrebno je matricu popuniti nulama pri konstrukciji.
- U glavnom programu instancirati dvije matrice cijelih brojeva dimenzija 2x2, u prvoj inicijalizovati element na poziciji (0,0) sa 1, a u drugoj element na (1,1) sa jedan. Potom ispisati zbir ovih dviju matrica u izlazni fajl pod nazivom „out.txt“ bez korištenja privremenog objekta.

Pitanja:

- Kod navedene implementacije glavnog programa, da li se negdje poziva konstruktor kopije ili konstruktor pomjeranja? Gdje?
- Kada bi se u glavnom programu koristio jedan **try** blok za obradu svih izuzetaka koje klasa **Matrix<T>** može da baci koliko bi bilo potrebno **catch** blokova? Zašto?

Napomene:

Razdvojiti interfejs i implementaciju.

Izbjeći dupliranje koda.

U zaglavlju klase koristiti punu specifikaciju imena objekata i metoda iz standardne biblioteke.

Koristiti standardnu biblioteku.

Sve metode treba da u zaglavlju sadrže odgovarajuću specifikaciju za signaliranje da li metoda baca izuzetak ili ne.

Deklaracije i definicije svih klasa treba da budu u prostoru imena **Ispit**.