
PROGRAMSKI JEZICI 1 – K2 (17.1.2017.)

1. Napisati klasu **IPrintable** koja treba da predstavlja interfejs koji omogućava ispis objekta na izlazni tok. Ova klasa treba da implementira sljedeće elemente.
 - Čistu virtuelnu metodu **print(std::ostream&)** koja štampa objekat na izlazni tok.
 - Operator za ispis na izlazni tok koji koristi metodu **print(std::ostream&)** za ispis, te omogućava štampanje potomaka korištenjem reference na ovu klasu.
 - 1.1. Kakva je klasa **IPrintable**? Može li se instancirati objekat te klase?
2. Napisati generičku klasu **FilteredSet<T, N>** koja predstavlja skup podataka tipa **T**, sa maksimalnim brojem elemenata **N** (pri čemu je, za podrazumijevanu vrijednost **N=0**, maksimalni broj elemenata beskonačan, te je **N** osmobajtna cjelobrojna vrijednost). Klasu izvesti iz klase **std::unordered_set<T>** tako da se realizuje operacija kompozicije. Klasu je potrebno izvesti iz klase **IPrintable** tako da se omogući polimorfizam, te da se potomci ove klase mogu ispisivati na izlazni tok. Pored ovoga, potrebno je realizovati i sljedeće elemente.
 - Realizovati štampanje na izlazni tok (svaki element u posebnu liniju) korištenjem odgovarajućih metoda iz natklase.
 - Definirati alias za funkciju za filtriranje elemenata tipa **T**, te definirati konstruktor koji prihvata referencu na objekat s tim aliasom. Klasa treba da čuva konstantnu referencu na funkciju za filtriranje proslijeđenu kroz konstruktor.
 - Onemogućiti korištenje operatora čija funkcionalnost nije moguća s obzirom na čuvanu referencu.
 - Statički onemogućiti instanciranje klase čija je vrijednost parametra **N** manja od 0.
 - Realizovati metodu **size()** koja vraća trenutni broj elemenata u skupu (elementi skupa ne smiju da se ponavljaju).
 - Realizovati virtuelnu metodu **add(...)** koja prihvata inicijalizatorsku listu elemenata tipa **T** koji se odjednom mogu dodati u skup. Onemogućiti kompajliranje klase u slučaju da tip **T** nema konstruktor kopije. Metoda **add(...)** treba da omogući dodavanje elemenata u klasu samo ako prolaze uslov definisan funkcijom za filtriranje, te ako u skupu ima dovoljno mjesta za njihovo ubacivanje. Svi elementi koji se mogu dodati u skup treba da budu dodati, a svi elementi čije dodavanje nije moguće, treba da budu sačuvani u izuzetak koji se podiže na kraju funkcije (pogledati 3. zadatak).
3. Napisati generičku klasu **FilterException<T>** koja predstavlja izuzetak koji sadrži sve elemente klase **FilteredSet<T, N>** koji nisu uspješno dodati u skup. Ova klasa treba da bude izvedena iz klase **FilteredSet<T>** i **std::exception**, pri čemu polimorfizam treba da bude omogućen. Za ovu klasu potrebno je implementirati sljedeće elemente.
 - Konstantan privatni podatak član koji predstavlja funkciju za filtriranje koja uvijek vraća true kao rezultat.
 - Podrazumijevani konstruktor koji inicijalizuje objekat natklase **FilteredSet<T>** sa navedenom funkcijom za filtriranje, te objekat natklase **std::exception** sa porukom „*Filter exception*“.
 - Konstruktor kopije delegiranjem konstruktora i korištenjem zaštićenog iteratora iz natklase i metode **add(...)** iz natklase.
 - Redefinisati metodu **print(...)** iz natklase tako da prije štampanja elemenata umetnutih u izuzetak ispiše poruku „*Elements not placed*“. Onemogućiti dalje redefinisanje ove metode.
4. U glavnom programu potrebno je uraditi sljedeće korake, redom.
 - Definirati objekat klase **FilteredSet<int, X>** sa funkcijom za filtriranje koja filtrira samo pozitivne elemente. Vrijednost maksimalnog broja elemenata **X** jednaka je bitskoj operaciji XOR nad binarnom vrijednošću 1111010100100001 i vrijednošću faktoriijela broja 20.
 - U istom bloku pokušati dodavanje elemenata 2, 4, -2, 5, 2 i -1, a zatim elemenata 1, 2, 3 i -1. Uхватiti potencijalne izuzetke, te upisati poruku izuzetka kao i elemente koji nisu uspješno umetnuti u skup u izlazni fajl pod nazivom „error_output.txt“.
 - Uspješno umetnute elemente ispisati na standardni izlaz.
 - 4.1. Šta je ispisano na standardni izlaz, a šta u izlaznu datoteku?

NAPOMENE:

- Minimizirati broj linija koda.
- Izbjeći dupliranje koda.
- Ispravno označiti metode koje sigurno bacaju izuzetke i koje sigurno ne bacaju izuzetke.
- Ispravno označiti konstantne metode.
- Poštovati konvencije.
- Nije neophodno modularizovati kod. Smatrati da se sav kod, osim koda glavnog programa, nalazi u istom heder fajlu.
- Izbjeći korištenje mehanizma prijateljstva.