PROGRAMSKI JEZICI 1 (30.1.2017.)

1. Dat je sljedeći segment koda.

```
#include <iostream>
class A
{
        int x;
        A() { std::cout << "A" << (x = ++y); }
        A(const A&) { std::cout << "A" << (x = ++y); }
        ~A() { std::cout << "dA" << x; }
        static int y;
};
template <class T = A>
class B : public A
{
        T a;
        B() { std::cout << "B"; }
        B(B&&) { std::cout << "mB"; }
        B& operator=(const B&) = delete;
};
using C = B<>;
```

```
int A::y = 1;

A* pa;
C* pc;
A a;

int main()
{
        A a;
        C c;
        pa = pc = new C();
        { A a = A(*pc);} //X
        std::cout << std::endl;
}
```

   1.1. Koje dvije izmjene je potrebno uraditi da bi se dati kod uspješno kompajlirao?
   1.2. Kada se date izmjene naprave, šta je izlaz ovakvog programa?
   1.3. Označiti liniju koda u kojoj je napravljeno curenje memorije, i napisati način da se ono riješi.
   1.4. Napisati segment koda koji će aktivirati konstruktor sa pomijeranjem klase **A**.
   1.5. Da je klasa **B** imala dinamičku alokaciju u konstruktoru, šta bi trebalo dopisati u dati kod da nakon njegovog izvršenja ne bude curenja memorije?
   1.6. Da li je linija koda označena sa *X* mogla da se napiše korištenjem operatora dodjele? Da se u istoj liniji koda instancirala klasa **C**, da li je tada mogao da bude korišten operator dodjele i zašto?

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** četvorobajtna cjelobrojna vrijednost). Klasu izvesti iz klase **`std::unordered_set<T>`** tako da se realizuje operacija kompozicije. Pored ovoga, potrebno je realizovati i sljedeće elemente.
   - Realizovati štampanje na izlazni tok (svaki element u posebnu liniju) metodom **`print(std::ostream&)`** i operatorom za ispis na izlazni tok, tako da se omogući štampanje potomaka na standardni izlaz, bez potrebe za definisanjem novih operatora za ispis.
   - Definisati alias za funkciju za filtriranje elemenata tipa **T**, te definisati 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 **`operator +=`** koji prihvata referencu na element tipa **T** i omogućava njegovo dodavanje u listu, ukoliko on zadovoljava uslov definisan funkcijom za filtriranje. Ukoliko element ne zadovoljava uslov, podići standardni izuzetak sa porukom „*Invalid element*“.

3. Napisati klasu koja enkapsulira jedan cjelobrojan osmobajtni podatak, alociran u dinamičkoj memoriji. Ispravno izvesti enkapsulaciju podatka i ispoštovati osnovne principe objektno-orijentisanog programirana. Implementirati sve potrebne metode, konstruktore i destruktore, tako da ne postoji curenje memorije i da je omogućeno dalje nasljeđivanje klase.

4. Uzimajući u obzir prethodne zadatke, odgovoriti na sljedeća pitanja.
   4.1. Na koji način bi se realizovalo umetanje u objekat **x** tipa **`FilteredSet<int, 10>`** u glavnom programu, tako da se garantuje uspješno izvršavanje programa, bez neočekivanog zatvaranja?
   4.2. Kada se neka lokalna promjenljiva **n** može koristiti za instanciranje klase **`FilteredSet<int, n>`**?
   4.3. Napisati elementaran primjer tipa koji ne može da se koristi kao argument šablona za klasu **`FilteredSet<T>`**.

NAPOMENE:

- Minimizirati broj linija koda i izbjeći dupliranje koda.
- Ispravno označiti metode koje sigurno bacaju izuzetke i koje sigurno ne bacaju izuzetke, kao i konstante metode.
- Poštovati konvencije, razdvojiti interfejs od implementacije.