ELEKTROTEHNIČKI FAKULTET
BANJA LUKA

PROGRAMSKI JEZICI 1 – K2

❶ (**20 bodova**) Data je sljedeća klasa.

```
#pragma once
#include <iostream>
#include <string>
namespace Kolokvijum
{
  class Printable
  {
    friend std::ostream& operator<<
        (std::ostream& str, Printable& pr)
    {
      pr.print(str);
      return str;
    }
  public:
    Printable(const char* name) : name(name)
    { id = ++count; }
    virtual ~Printable() {}
  protected:
    std::string name; int id;
    void print(std::ostream& str)
    { run1(); printName(str); run2(); }
    virtual void printName(std::ostream& str)
    { str << name; }
    virtual void run1() { std::cout << "s" << id++; }
    virtual void run2() { std::cout << "e" << id; }
  private:
    static int count;
  };
}
```

Može se smatrati da je statička promjenljiva **count** inicijalizovana na 0. Ako je program implementiran na sljedeći način, odrediti ispis na standardnom izlazu nakon izvršavanja programa.

```
#include "Printable.h"
#include <algorithm>
#include <vector>
using ::std::cout;

class Test : public Kolokvijum::Printable
{
public:
  Test(char* test) : Printable(test) {}
  ~Test() { cout << "D"; }
protected:
  virtual void run1() override
  { cout << "r"; Printable::run2(); }
  virtual void run2() override
  {
    std::for_each(name.begin() + 1, name.end() - 2,
      [this](auto x) {cout << x << ++id;});
  }
};

int main()
{
  Test test("Terra");
  std::vector<Kolokvijum::Printable*> vec;
  vec.push_back(&test);
  vec.push_back(&test);
  for (auto& x : vec)
    cout << *x << std::endl;
  return 0;
}
```

❷ (**40 bodova**) U prostoru imena **Kolokvijum** definisati klasu **Person** javno naslijeđenu iz klase **Printable** definisane u prethodnom zadatku. Klasa **Person** predstavlja osobu i treba da kroz konstruktor omogući inicijalizaciju imena i jedinstvenog matičnog broja (pri čemu treba koristiti naslijeđene podatke članove), te da otkriva njihove getere. S obzirom da je klasa **Printable** loše projektovana, potrebno je u naslijeđenoj klasi identifikovati i ispraviti greške u dizajnu originalne klase. Potrebno je obezbijediti mogućnost daljeg nasljeđivanja iste klase. Realizovati:

- Operator poređenja dvaju osoba na osnovu jedinstvenog matičnog broja.
- Getere za ime i matični broj
- Metodu **print(std::ostream&)** koja treba da bude vidljiva samo izvedenim klasama, pri čemu ispis na standardni izlaz pokazivačem na klasu **Printable** mora i dalje da bude moguć. Ova metoda treba da štampa matični broj osobe i njeno ime, pri čemu matični broj uvijek treba da zauzima isti broj karaktera pri ispisu, s lijevim poravnanjem (koristiti **<iomanip>**)
- Onemogućiti izvršavanje nepotrebnih metoda iz natklase, onemogućiti njihovo dalje preklapanje i sakriti ih od potomaka
- Izvršiti specijalizaciju šablona za **std::hash**, pri čemu se vrijednost heša računa iz jedinstvenog matičnog broja. Zaglavlje za **std::hash** dato je ispod.

```
struct std::hash<Klasa>
{
  size_t operator()(const Klasa& x) const
  {
    return izracunHesa(x);
    /*npr.:
      return std::hash<int>()(x.Vrijednost);*/
  }
};
```

❸ (**40 bodova**) U prostoru imena **Kolokvijum** implementirati klasu **HashTable<T, hash>** koja predstavlja heš[1] tabelu elemenata tipa **T** sa podrazumijevanim heš generatorom **std::hash<T>**. Konstruktor ove klase treba da prihvata parametar koji određuje kapacitet tabele. Potrebno je implementirati konstruktore kopije i pomjeranja, operatore dodjele (sa pomjeranjem i kopiranjem) i destruktor. Omogućiti optimizaciju koda garantovanjem da konstruktori neće bacati izuzetke. Potom realizovati:

- Funkciju **insert(T&)** za umetanje elemenata. Elemente umetati na indeks određen heš vrijednošću preslikanom na opseg niza operatorom **%** (koristiti niz pokazivača na umetnute elemente), a ako element već postoji na tom indeksu u neku privatnu kolekciju (na primjer **std::vector**). Ako je u tabeli već maksimalan broj elemenata, baciti izuzetak tipa **char*** sa porukom „Hash table full."
- Funkciju **exists(const T&)** koja određuje da li dati element postoji u tabeli. Ova funkcija ne smije da baci izuzetak.
- U glavnom programu ilustrovati rad heš tabele osoba, sa hvatanjem potencijalnih izuzetaka i upisom osoba u datoteku pod nazivom „out.txt".

Napomene:

Razdvojiti interfejs i implementaciju.

Izbjeći dupliranje koda.

U zaglavljima klasa koristiti punu specifikaciju imena objekata i metoda iz standardne biblioteke.

Koristiti standardnu biblioteku.

[1]Heš je generisana cjelobrojna vrijednost kojom se predstavlja objekat neke klase. Heš ne mora da bude jedinstven za svaki objekat, ali se obično konstruiše tako da različiti objekti iste klase imaju različite heševe, kako bi se pojednostavilo poređenje. Ako dva objekta imaju različit heš, oni su različiti. Ako dva objekta imaju isti heš, onda ih je neophodno porediti odgovarajućim operatorom poređenja.