Poruke

Cilj ovog zadatka je implementirati kompletnu **kontrolu kopiranja**. U programu imamo dvije klase: `Poruka` i `Kategorija`. `Poruka` je klasa koja drži string (sadržaj) poruke. Kako svaka poruka može pripadati raznim kategorijama, ona sadrži i skup pokazivača na tip `Kategorija`. Ti pokazivači pokazuju na kategorije kojima poruka pripada.

`Kategorija` poruke je klasa koja sadrži opis kategorije (string) i skup pokazivača na poruke koje pripadaju toj kategoriji. Nudi samo metode za dodavanje i uklanjanje poruke, te metodu koja vraća opis kategorije (string).

Klasa `Poruka` sadrži metode `dodaj()` i `ukloni()` kojima poruku ubacujemo u kategoriju i izbacujemo je iz kategorije. Pored toga ona može ispisati sadržaj poruke i sve kategorije u kojima se poruka nalazi.

Preostali dio sučelja su konstruktori i kontrola kopiranja.

Napomene o kontroli kopiranja:

- Instance klasa `Poruka` i `Kategorija` moraju u svakom trenutku biti usklađene.To znači da ako neka poruka tvrdi da je u nekoj kategoriji, onda ta kategorija mora držati pokazivač na tu poruku, i obratno. Ako kategorija drži pokazivač na poruku, onda i poruka mora držati pokazivač na kategoriju. Metode `dodaj()` i `ukloni()` u klasi `Poruka` trebaju brinuti o sinhronizaciji. Ta se zavisnost mora sačuvati i kroz kontrolu kopiranja.
- Pri konstrukciji kopije poruke mi stvaramo novu poruku koja mora biti u istim kategorijama kao i poruka koju kopiramo. Stoga će nova kopija kopirati skup kategorija stare i dodati sebe u sve te kategorije.
- Operator pridruživanja poruke djeluje slično, ali kod njega je operand na lijevoj strani već konstruiran i on mora brinuti o samopridruživanju. Činjenica da je operand na lijevoj strani već u nekim kategorijama znači da se prije preuzimanja novih kategorija mora izbaciti iz starih.
- Konstruktor kopije poruke **preuzimanjem** treba koristiti `std::move` kako bi angažirao M-Ctor klasa `string` i `set`. On se mora pobrinuti da poruka na kojoj smo izvršili move-operaciju ne ostaje visiti u kategorijama te da je njena lista kategorija na kraju prazna.
- OP poruke **preuzimanjem** djeluje analogno kao i M-Ctor. K tome mora brinuti o samopridruživanju i izbacivanju lijeve strane iz *starih kategorija*.
- Operacije kopiranja klase Kategorija su proglašene **obrisanim**.

Sučelje obiju klasa su dana u *poruka.h* datoteci. Sučelje *ne mijenjati*. Sve metode za obje klase implementirati ustotteci **poruke.cpp**, s time da se prvo implementiraju metode klase `Kategorija` a zatim metode klase `Poruka`.

```
#ifndef PORUKA_H_INCLUDED
#define PORUKA_H_INCLUDED

#include <string>
#include <set>

class Poruka;

class Kategorija{
public:
    // pri konstrukciji je obavezan opis kategorije
    explicit Kategorija(std::string const & opis);
    // kontrola kopiranja
    Kategorija(Kategorija const&) = delete;
    Kategorija(Kategorija &&) = delete;
```

```
    Kategorija& operator=(Kategorija const &) = delete;
    Kategorija& operator=(Kategorija &&) = delete;
    ~Kategorija();

    // dodaj poruku u kategoriju
    void dodaj_poruku(Poruka *);
    // ukloni poruku iz kategorije
    void ukloni_poruku(Poruka *);
    // vrati opis kategorije
    std::string opis() const { return mopis; }
private:
    // pokazivači na poruke u kategoriji
    std::set<Poruka*> mporuke;
    // opis kategorije
    std::string mopis;
};

class Poruka{
    friend class Kategorija;
  public:
    // pri konstrukciji poruke obavezano je dati sadržaj poruke
      explicit Poruka(std::string const & sadrzaj = "");
      // Kontrola kopiranja
      Poruka(Poruka const &);
      Poruka(Poruka &&);
      Poruka& operator=(Poruka const&);
      Poruka& operator=(Poruka&&);
      ~Poruka();
      // dodaj poruku u kategoriju
      void dodaj(Kategorija&);
      // ukloni poruku iz kategorije
      void ukloni(Kategorija&);
      // vrati poruku
      std::string sadrzaj() const { return msadrzaj; }
      // vrati sve kategorije u kojima je poruka (odvojene bjelinom, kao jedan
string)
      std::string kategorije() const;
  private:
    std::string msadrzaj;
    std::set<Kategorija*> mkategorije;
};

#endif // PORUKA_H_INCLUDED
```

Sljedeći glavni program mora raditi ispravno (*ne mijenjati*):

```
#include <iostream>

#include "poruka.h"


// Pomoćna funkcija za ispis
void ispis(Poruka const & poruka){
    std::cout << "[ Poruka: " << poruka.sadrzaj()
              << ": Kategorije: " << poruka.kategorije()
              << " ]\n";
 }

 // generiranje poruke
Poruka prazna_poruka(Kategorija& kat){
     Poruka tmp("b.o.");
     tmp.dodaj(kat);
     return tmp;
}
```

```
int main()
{
    // Definiramo 3 kategorije, no moguć je proizvoljan broj kategorija
    Kategorija redovno("Redovno");
    Kategorija vazno("Vazno");
    Kategorija hitno("Hitno");


    // Definiramo nekoliko poruka
    Poruka p1("New USB device found");
    Poruka p2("Manufacturer: Broadcom Corp");
    Poruka p3("[sda] Attached SCSI disk");
    Poruka p4("Disk quotas exceeded");
    Poruka p5("Kernel panic");

    // Dodajmo poruke u neke kategorije
    p1.dodaj(redovno); p1.dodaj(vazno);
    p2.dodaj(redovno);
    p3.dodaj(redovno); p3.dodaj(vazno);
    p4.dodaj(redovno); p4.dodaj(vazno); p4.dodaj(hitno);
    p5.dodaj(vazno); p5.dodaj(hitno);

    // Ispis radi kontrole
    std::cout << "Poruke:\n";
    ispis(p1);
    ispis(p2);
    ispis(p3);
    ispis(p4);
    ispis(p5);

    // test uklanjanja
    p1.ukloni(vazno);
    p4.ukloni(redovno);

    // Ispis radi kontrole
    std::cout << "Modificirane poruke:\n";
    ispis(p1);
    ispis(p4);

    Poruka p6{p5};
    std::cout << "CCtor:\n";
    ispis(p6);

    p6 = p1;
    p5 = p5;
    std::cout << "OP\n";
    ispis(p6);
    ispis(p5);

    p6 = prazna_poruka(redovno);
    std::cout << "M-OP\n";
    ispis(p6);

    Poruka* pp7 = new Poruka("Watching system buttons");
    pp7->dodaj(redovno);
    Poruka p8(std::move(*pp7));
    std::cout << "M-Ctor\n";
    ispis(p8);
    delete pp7;

    return 0;
}
```

Datoteke `main.cpp` i `poruka.h` su vam u potpunosti dane na ovoj stranici. Trebate implementirati samo

datoteku `poruka.cpp`.

Last Modified: *February 26, 2015*

<>