

Matrica statičkih dimenzija

Treba konstruirati klasu `Matrix` koja je parametrizirana s brojem redaka i brojem stupaca. `Matrix` ne alocira dinamički memoriju i ne dozvoljava promjenu dimenzija. Klasu implementirati na sljedeći način:

```
// Matrica sa statički zadanim dimenzijama. Nije moguća realokacija.
// rows = broj redaka, cols = broj stupaca
template <int rows, int cols>
class Matrix{
public:
    // operatori indeksiranja: i=redak, j=stupac
    double & operator()(unsigned int i, unsigned int j);
    double const & operator()(unsigned int i, unsigned int j) const;

    // postavi sve elemente na nulu
    void zero();
    // postavi na jediničnu matricu
    void eye();
private:
    // Pamtimo elemente zapisane po recima
    std::array<double, rows*cols> mData;
};
```

Elementi se pamte u jednom jednodimenzionalnom polju i pretpostavlja se da je matrica zapisana u polje po recima (kao dvodimenzionalno polje u C-u). Operatori indeksiranja su implementirani kao funkcijski pozivi radi prirodnije sintakse.

Za klasu `Matrix` potrebno je implementirati sljedeće operatore:

1. Operatore dohvata kao funkcijske pozive;
2. Operatore zbrajanja i oduzimanja;
3. Složene operatore pridruživanja `+=` i `-=`;
4. Množenja skalarom s lijeva i desna;
5. Operator množenja koji implementira množenje matrica;
6. Eksplicitni operator konverzije u `bool`;
7. Operator ispisa.

Pored toga treba implementirati

1. Konstruktor koji uzima inicijalizacijsku listu.
2. Operator pridruživanja pomicanjem radi poboljšanja efikasnosti koda.

Sljedeća `main` funkcija mora ispravno raditi:

```
using std::cout;
using std::endl;

int main()
{
    Matrix<2,3> A{{1,2,3},{2,3,1}}, B({{-1,1,4},{0,0,3}});
    Matrix<3,1> x(3.0), y({{0},{1},{0}});
    Matrix<2,1> z, res{{48},{45}};
    z.zero();

    Matrix<2,3> M;
    M.eye();
    cout << M << endl;
```

```
z = (2*A-M+B*3)*(x-3*y); // koliko privremenih objekta je ovdje konstruirano?  
  
z -= res;  
if(z) cout << "o.k.\n";  
else cout << "pogresno!\n";  
return 0;  
}
```

Napomene.

- Sav kod mora biti u `.h` datoteci jer se radi o parametriziranoj klasi.
- Operatori definirani izvan klase neka koriste sučelje klase (ne koristiti `friend` deklaraciju).
- Definiciju funkcija članica pisati izvan klase ukoliko funkcije nisu sasvim male.
- Kada se funkcija članica parametrizirane klase piše izvan klase ona se mora kvalificirati imenom klase sa template parametrima. Na primjer:

```
template <int rows, int cols>  
Matrix<rows, cols>& Matrix<rows, cols>::operator=(Matrix && m)  
{  
    // ...  
    return *this;  
}
```

Last Modified: *February 26, 2015*

