נושאים מתקדמים בתכנות מונחה עצמים

תרגיל/מעבדה 8

פרופ' עפר שיר

ofersh@telhai.ac.il

החוג למדעי המחשב

תל-חי

המכללה האקדמית

הצעת פתרון: Matrix

- נושא: מימוש מטריצה בדיזיין אלטרנטיבי והשגת יעילות באמצעות *move semantics*
- שימו לב לשימוש באלגוריתמי העתקה שונים:
  - `std::uninitialized_copy` שימוש ב-*"placement new"*
  - `std::copy`
- העברות מסתמכות על "נטילת ייצוג" או פשוט על `std::swap`

```
template<class T>
class Matrix {
  std::array<int,2> dim;
  T* elem; // pointer to size() elements of type T
public:
  Matrix(int d1, int d2) :dim{d1,d2}, elem{new T[d1*d2]} { }
  int size() const { return dim[0]*dim[1]; }
  /* copy constructor */
  Matrix(const Matrix& m) : dim{m.dim}, elem{new T[m.size()]}
  {
    std::uninitialized_copy(m.elem,m.elem+m.size(),elem);
  // place (using "placement new") + copy elements
  }
  /* move constructor */
  Matrix(Matrix&& a) : dim{a.dim}, elem{a.elem} {
    a.dim = {0,0};
    a.elem = nullptr;
  }
```

```
/* copy assignment */
Matrix& operator=(const Matrix& m) {
  if (dim[0]!=m.dim[0] || dim[1]!=m.dim[1])
    throw runtime_error("bad size in Matrix =");
  std::copy(m.elem,m.elem+m.size(),elem);
  return *this;
}
/* move assignment */
Matrix& operator=(Matrix&& a) {
  std::swap(dim,a.dim);
  std::swap(elem,a.elem);
  return *this;
}

~Matrix() { delete[] elem; }

friend Matrix operator+(const Matrix& a, const Matrix& b);
};
```

operator+

```
Matrix operator+(const Matrix& a, const Matrix& b)
{
  if (a.dim[0]!=b.dim[0] || a.dim[1]!=b.dim[1])
    throw std::runtime_error("unequal Matrix
sizes");
  Matrix res{a.dim[0],a.dim[1]};
  constexpr auto n = a.size();
  for (int i = 0; i!=n; ++i)
    res.elem[i] = a.elem[i]+b.elem[i];
  return res;
// move semantics takes care of efficiency
}
```

תרגיל נוכחי: C++0x

1. חמשת הגדולים של **`SqMatrix`**
2. מימוש **`printf`** באמצעות תבניות וריאדיות