Course $C^{++}$

Exercise List 4

Deadline: 27.03.2014

Topic of this exercise is the use of defined operators:

1. In class `rational` of Task 2, make the fields `num` and `denum` private.
   Add getters for `num` and `denum`, and rewrite the rest of the program so that it compiles again.

2. Extend the `stack` class of Task 3 with
   
   `stack operator + ( const stack& s1, const stack& s2 )`. You may also implement `stack operator + ( const stack& s )` as member function.

   (This operator is a bit unnatural for stacks, but we need to practice use of operators in some way.)

3. Extend the `stack` class with operators

   ```cpp
   double operator [] ( unsigned int i ) const;
   double& operator [] ( unsigned int i );
   ```

   The top of stack has index 0 and the element on the bottom of the stack has index `size() - 1`.

   If you want the operator to check bounds, use `ASSERT`.

4. Extend the `stack` class with a constructor for initializer lists.

   ```cpp
   stack( std::initializer_list< double > s );
   ```

   Show that you know how to use the constructor in `main`.

5. Add operators

   ```cpp
   void operator += ( double ); // Same as push( );
   void operator += ( const stack& s );
   // Be careful with self assignment!
   ```

6. Check for the absence of memory leaks, using a loop that contains the addition operator, the new assignments, and the new constructor.