Задание по C++ #11, осень 2018

В стандартной библиотеке есть семейство функций to_string для преобразования чисел в строки. Однако обратное преобразование не такое удобное — для каждого числового типа есть своя функция (например, strtoi для int).

В данном задании вам предлагается написать шаблонную функцию from_string, которая умеет преобразовывать строку в разные типы. Для реализации from_string предлагается воспользоваться классом std::istringstream который представляет собой поток ввода из строки, т.е. для преобразования строки в тип T предлагается прочитать значение типа T из потока при помощи оператора >>. В случае неудачного преобразования функция должна бросать исключение bad_from_string, класс которого вам нужно реализовать самостоятельно.

Пример использования from_string:

```
string s1("123");
int    a1 = from_string<int>   (s1); // a1 = 123
double b1 = from_string<double>(s1); // b1 = 123.0
string c1 = from_string<string>(s1); // c1 = "123"

string s2("12.3");
int    a2 = from_string<int>   (s2); // исключение
double b2 = from_string<double>(s2); // b2 = 12.3
string c2 = from_string<string>(s2); // c2 = "12.3"

string s3("abc");
int    a3 = from_string<int>   (s3); // исключение
double b3 = from_string<double>(s3); // исключение
string c3 = from_string<string>(s3); // c3 = "abc"
```

Указания:

1. Для того, чтобы учитывать пробельные символы, используйте std::noskipws (например, если строка с числом начинается с пробела или заканчивается пробелом, то это должно быть ошибкой).
2. Флаг eof() у потоков устанавливается только, если не удалось прочитать символ: если при чтении из потока с 5-ю символами прочли 5 символов, но при этом 6-ой (отсутствующий) символ прочесть не пытались, то eof() будет выдавать false.
3. Не забудьте определить конструктор bad_from_string от char const * или от std::string.