Лабораторная работа №01. Подсказки.

Ниже представлены интерфейсы класса IniParser разной степени сложности (внимание, есть разрывы страницы – листайте документ до конца).

1. Простой.

```
class IniParser {
public:
    IniParser();
    ~IniParser();

    // Opens Ini file.
    void Initialize(const char* filename_cstr);

    // Checks if a section exists.
    bool IsHaveSection(const char* section_name);

    // Checks if a pair param-section exists.
    bool IsHaveParam(const char* section_name, const char* param_name);

    // Returns integer value for a pair param-section.
    int GetValueInt(const char* section_name, const char* param_name);

    // Returns double value for a pair param-section.
    double GetValueDouble(const char* section_name, const char* param_name);

    // Return string value for a pair param-section.
    std::string GetValueString(const char* section_name, const char* param_name);

private:
    // ...
};
```

2. Добавлены модификаторы const для методов, не изменяющих поля класса.
   - https://msdn.microsoft.com/en-us/library/07x6b05d.aspx (раздел const member functions)

```
class IniParser {
public:
    IniParser();
    ~IniParser();

    void Initialize(const char* filename_cstr) const;

    bool IsHaveSection(const char* section_name) const;

    bool IsHaveParam(const char* section_name, const char* param_name) const;

    int GetValueInt(const char* section_name, const char* param_name) const;

    double GetValueDouble(const char* section_name, const char* param_name) const;

    std::string GetValueString(const char* section_name, const char* param_name) const;

private:
    // ...
};
```

См. далее следующую страницу.

3. Добавлены спецификации исключений (пользовательские).
   - http://www.cplusplus.com/doc/tutorial/exceptions/ (раздел Exception specification)

```
#include "parser_exceptions.h"

class IniParser {
public:
    IniParser();
    ~IniParser();

    void Initialize(const char* filename_cstr) const
        throw (exc_io);

    bool IsHaveSection(const char* section_name) const
        throw (exc_cfg_not_initied);

    bool IsHaveParam(const char* section_name, const char* param_name) const
        throw (exc_cfg_not_found, exc_cfg_not_initied);

    int GetValueInt(const char* section_name, const char* param_name) const
        throw (exc_cfg_not_found, exc_cfg_not_initied);

    double GetValueDouble(const char* section_name, const char* param_name) const
        throw (exc_cfg_not_found, exc_cfg_not_initied);

    std::string GetValueString(const char* section_name, const char* param_name) const
        throw (exc_cfg_not_found, exc_cfg_not_initied);

private:
    // ...
};
```

4. Шаблонная функция получения значений.
   - Templeate specialization (специализация шаблона) http://ru.cppreference.com/w/cpp/language/template_specialization

```
class IniParser {
public:
    IniParser();
    ~IniParser();

    void Initialize(const char* filename_cstr) const
        throw (exc_io);

    bool IsHaveSection(const char* section_name) const
        throw (exc_cfg_not_initied);

    bool IsHaveParam(const char* section_name, const char* param_name) const
        throw (exc_cfg_not_found, exc_cfg_not_initied);

    template<typename T>
    T GetValue(const char* section_name, const char* param_name) const
        throw (exc_cfg_not_found, exc_cfg_not_initied);

private:
    // ...
};
```