

# Тема 8: Обработка ошибок и исключительных ситуаций

## Цель

---

Получить навыки обработки ошибок и исключительных ситуаций

## Общее задание

---

- Выполнить ASSERT проверку, если выполняется попытка удаления несуществующего элемента коллекции
- Доработать оператор индексирования разработанного класса коллекции таким образом, чтобы он порождал исключительную ситуацию наиболее подходящего типа

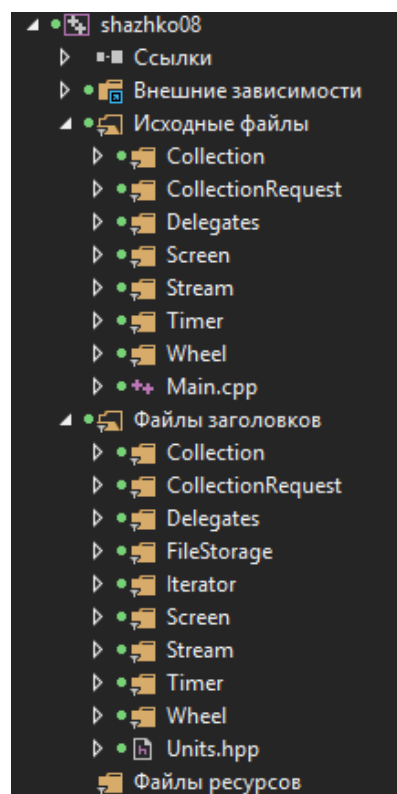
## Индивидуальное задание

---

- Использовать макросы, присущие компилятору Microsoft

## Структура проекта

---



# Описание разработанных типов данных

Разработка типов данных не производилась. Добавлялся новый функционал в классы SimpleList лабораторной работы #7.

## Диаграмма классов

См. предыдущий пункт.

## Разработанные методы и функции

См. предыдущий пункт.

## Ссылки на файлы проекта

CarWheelFilterByManufacturer.h	Содержит объявление класса CarWheelFilterByManufacturer
CarWheelFilterDecorator.cpp	Реализация класса CarWheelFilterDecorator
CarWheelFilterDecorator.h	Содержит объявление класса CarWheelFilterDecorator
CarWheelScreen.cpp	Содержит реализацию класса CarWheelScreen
CarWheelScreen.h	Содержит объявление класса CarWheelScreen
CarWheelScreenCreator.hpp	Содержит объявление класса CarWheelScreenCreator
CollectionRequest.hpp	Содержит объявление и реализацию класса CollectionRequest
DefaultScreen.cpp	Содержит реализацию класса DefaultScreen
DefaultScreen.h	Содержит объявление класса DefaultScreen
DefaultScreenCreator.hpp	Содержит объявление класса DefaultScreenCreator
Delegate.hpp	Содержит вспомогательные методы позволяющие создавать делегаты
DelegateVoid.cpp	Реализация класса DelegateVoid
DelegateVoid.h	Объявление класса DelegateVoid
GraphScreen.cpp	Содержит реализацию класса GraphScreen
GraphScreen.h	Содержит объявление класса GraphScreen
GraphScreenCreator.hpp	Содержит объявление класса GraphScreenCreator
ICollection.h	Содержит объявление класса ICollection
IDelegateVoid.h	Объявление абстрактного класса IDelegateVoid
Iterator.h	Содержит объявление класса Iterator
InputStreamHelper.h	Содержит объявление абстрактного класса InputStreamHelper
IOCollection.cpp	Содержит реализацию абстрактного класса IOCollection
IOCollection.h	Содержит объявление абстрактного класса IOCollection
Main.cpp	
MethodDelegateVoid.hpp	Объявление и реализация класса MethodDelegateVoid
OutputStreamHelper.h	Содержит объявление абстрактного класса OutputStreamHelper
ScreenCreator.h	Содержит объявление класса ScreenCreator
SimpleInputStreamHelper.cpp	Содержит реализацию класса SimpleInputStreamHelper
SimpleInputStreamHelper.h	Содержит объявление класса SimpleInputStreamHelper
SimpleList.cpp	Реализация класса SimpleList
SimpleList.h	Содержит объявление класса SimpleList
SimpleListIterator.cpp	
SimpleOutputStreamHelper.cpp	Содержит реализацию класса SimpleOutputStreamHelper
SimpleOutputStreamHelper.h	Содержит объявление класса SimpleOutputStreamHelper
SimpleStreamHelperFactory.cpp	Содержит реализацию класса SimpleStreamHelperFactory
SimpleStreamHelperFactory.h	Содержит объявление класса SimpleStreamHelperFactory
SimpleTimer.cpp	
SimpleTimer.h	Содержит объявление класса SimpleTimer
StaticDelegateVoid.hpp	Объявление и реализация класса StaticDelegateVoid
StreamHelperArg.cpp	Содержит реализацию базового класса StreamHelperArg
StreamHelperArg.h	Содержит объявление базового класса StreamHelperArg
StreamHelperFactory.h	Содержит объявление абстрактного класса StreamHelperFactory
Units.hpp	Содержит объявление перечислений EUnits и функции по работе с ними
Wheel.cpp	Содержит реализацию класса Wheel
Wheel.h	Содержит объявление класса Wheel

## Текст программы

```

18 void WheelListShow(Collection::ICollection<Wheel::CarWheel*>::iterator it);
19 void CallBackFunc(Collection::ICollection<Wheel::CarWheel*>* collection){... }
20 Wheel::CarWheel* StringToObject1(std::string type){... }
21 int main() {
22 {
23     Collection::ICollection<Wheel::CarWheel*> *wheelList = NULL;
24     auto sh = new Stream::StreamHelper::SimpleStreamHelperFactory();
25     wheelList = Stream::IOCollection::Load<Wheel::CarWheel*>("wheelList", "CarWheelListSave.txt", sh, StringToObject1);
26
27     auto test = (*wheelList)[0];
28     try {
29         auto test = (*wheelList)[10]; // error
30     }
31     catch (std::out_of_range error) {
32         std::cout << error.what();
33     }
34     try {
35         wheelList->RemoveAt(10); // error
36     } catch(...){}
37     CarWheelFilter *filter = new CarWheelFilterByDiameter(508.0, new CarWheelFilterByManufacturer("Nokian", NULL));
38
39     CollectionRequest::Find<CarWheelFilter, Wheel::CarWheel*>(wheelList, filter, &CallBackFunc);
40     ОЧИСТКА ПАМЯТИ
41     }
42     _CrtDumpMemoryLeaks();
43     return 0;
44 }

```

## Результаты работы

[illegible]

Рисунок 1 - Результат выполнения с ошибками

```
Показать выходные данные из: Отладка
"shazhko08.exe" (Win32). Загружено "C:\Windows\System32\msvcrt.dll". Невозможно найти или открыть PDB-файл.
"shazhko08.exe" (Win32). Загружено "C:\Windows\System32\rpcrt4.dll". Невозможно найти или открыть PDB-файл.
Debug Assertion Failed!
|
Program: D:\GitHub\kit25a\ashazhko-artem\src\x64\Debug\shazhko08.exe
File: d:\github\kit25a\ashazhko-artem\src\shazhko08\simplelist.cpp
Line: 137

Expression: The index is outside the list boundary

For information on how your program can cause an assertion
failure, see the Visual C++ documentation on asserts.

(Press Retry to debug the application)
shazhko08.exe вызвал срабатывание точки останова.

Вызвано исключение по адресу 0x00007FFB201E9E08 в shazhko08.exe: исключение Microsoft C++: std::out_of_range по адресу памяти 0x000000F11DAFF418.
Поток 0x17e0 завершился с кодом 0 (0x0).
Поток 0x2414 завершился с кодом 0 (0x0).
Поток 0x524 завершился с кодом 0 (0x0).
Программа "[10376] shazhko08.exe" завершилась с кодом 0 (0x0).
```

Рисунок 2 - Результат очистки памяти

```
[=====] Running 2 tests from 2 test cases.
[-----] Global test environment set-up.
[-----] 1 test from SimpleList_getItemByIndex
[ RUN ] SimpleList_getItemByIndex.exception
[ OK ] SimpleList_getItemByIndex.exception (2 ms)
[-----] 1 test from SimpleList_getItemByIndex (4 ms total)

[-----] 1 test from SimpleList_pop
[ RUN ] SimpleList_pop.exception
[ OK ] SimpleList_pop.exception (2 ms)
[-----] 1 test from SimpleList_pop (4 ms total)

[-----] Global test environment tear-down
[=====] 2 tests from 2 test cases ran. (10 ms total)
[ PASSED ] 2 tests.
```

Рисунок 2 - Результат выполнения тестов

## Выводы

В ходе лабораторной работы были получены практические навыки обработки ошибок и исключительных ситуаций