

# 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

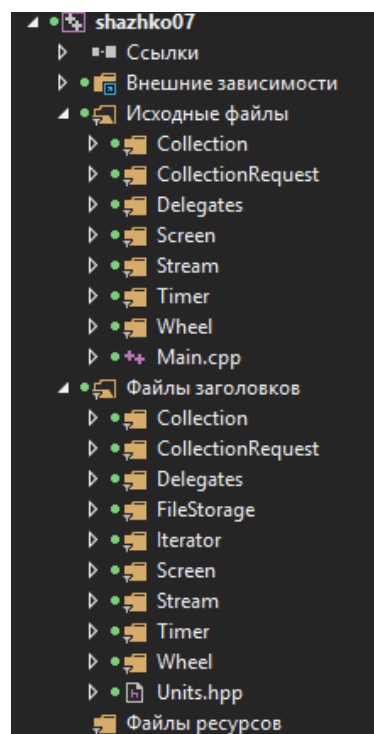
## Индивидуальное задание

---

- 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.

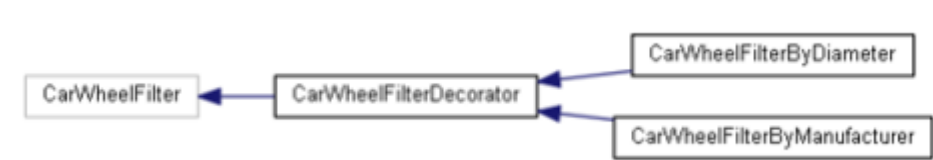
## Разработанные методы и функции

См. предыдущий пункт.

## Ссылки на файлы проекта

 <a href="#">abstract</a>	Абстрактный класс, который задает интерфейс фильтров
 <a href="#">CarWheelFilterByDiameter</a>	Класс, который фильтрует данные по диаметру
 <a href="#">CarWheelFilterByManufacturer</a>	Класс, который фильтрует данные по производителю
 <a href="#">CarWheelFilterDecorator</a>	Класс декоратор, от которого будут поражаться подклассы
 <a href="#">CFileStorage</a>	
 <a href="#">CollectionRequest</a>	Класс запросов для работы с коллекциями

## Текст программы



## Результаты работы








 <a href="#">CarWheelFilter.h</a>	Содержит объявление класса <a href="#">CarWheelFilter</a>
 <a href="#">CarWheelFilterByDiameter.cpp</a>	Реализация класса <a href="#">CarWheelFilterByDiameter</a>
 <a href="#">CarWheelFilterByDiameter.h</a>	Содержит объявление класса <a href="#">CarWheelFilterByDiameter</a>
 <a href="#">CarWheelFilterByManufacturer.cpp</a>	Реализация класса <a href="#">CarWheelFilterByManufacturer</a>
 <a href="#">CarWheelFilterByManufacturer.h</a>	Содержит объявление класса <a href="#">CarWheelFilterByManufacturer</a>
 <a href="#">CarWheelFilterDecorator.cpp</a>	Реализация класса <a href="#">CarWheelFilterDecorator</a>
 <a href="#">CarWheelFilterDecorator.h</a>	Содержит объявление класса <a href="#">CarWheelFilterDecorator</a>

Рисунок 1 - Результат выполнения с ошибками

```

void CarWheelListShow(Collection::ICollection<Wheel::CarWheel*>::iterator it);
void CallBackFunc(Collection::ICollection<Wheel::CarWheel*>* collection) {
    std::cout << "\n***** CallBackFunc *****\n";
    if (collection) {
        auto it = collection->CreateIterator();
        CarWheelListShow(it);
        delete it;
    }
    else
        std::cout << "Don't found\n";
    std::cout << "=====\\n";
}
Wheel::CarWheel* StringToObject1(std::string type){ ... }
int main() {
    {
        Collection::ICollection<Wheel::CarWheel*> *carWheelList =NULL;
        auto sh = new Stream::StreamHelper::SimpleStreamHelperFactory();
        carWheelList = Stream::IOCollection::Load<Wheel::CarWheel*>("wheelList", "CarWheelListSave.txt", sh, StringToObject1);

        CarWheelFilter *filter = new CarWheelFilterByDiameter(508.0, new CarWheelFilterByManufacturer("Nokian", NULL));

        CollectionRequest::Find<CarWheelFilter, Wheel::CarWheel*>(carWheelList, filter, &CallBackFunc);
    }
    _CrtDumpMemoryLeaks();
    return 0;
}

```

Рисунок 2 - Результат очистки памяти

## Выводы

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