

# Тема 7: Шаблоны, алгоритмы, функторы

## Цель

---

Получить навыки использования аппарата шаблонов функций и классов на базе собственного кода

## Общее задание

---

- К своему индивидуальному заданию написать алгоритм, используя шаблоны и функтор
- Для реализации взять контейнер из предыдущей работы

## Прикладная область

---

- Колеса

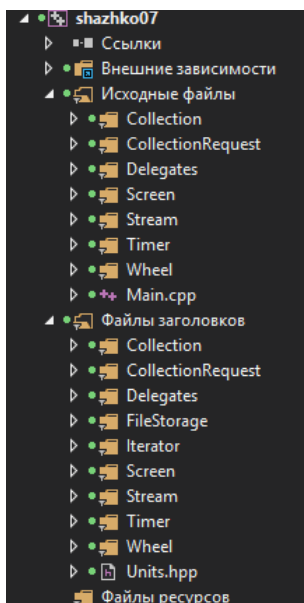
## Индивидуальное задание

---







- Алгоритм поиска колес заданного диаметра и конкретного производителя

## Структура проекта

---



# Описание разработанных типов данных

 <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


## Диаграмма классов



## Разработанные методы и функции

См. раздел **members** на заголовочных страницах соответствующих типов данных

## Ссылки на файлы проекта

 <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>

# Текст программы

```
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;
}
```

# Результаты работы

```
***** CallbackFunc *****
-----
Diameter: 508
Width: 500
Units: MILLIMETRES
Volume: 1.01341e+08
Tire Manufacturer: Nokian
Tire Name: MP-16
-----
Diameter: 508
Width: 275
Units: MILLIMETRES
Volume: 5.57378e+07
Tire Manufacturer: Nokian
Tire Name: Hakkapelitta 9
-----
=====
```

Рисунок 1 - Результат поиска

```
Показать выходные данные из: Отладка
"shazhko07.exe" (Win32). Загружено "D:\GitHub\kit25a\ashazhko-artem\src\x64\Debug\shazhko07.exe". Символы загружены.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\ntdll.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\kernel32.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\KernelBase.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\vcruntime140d.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\ucrtbased.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\msvcrt140d.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\msvcrt.dll". Невозможно найти или открыть PDB-файл.
"shazhko07.exe" (Win32). Загружено "C:\Windows\System32\rpcrt4.dll". Невозможно найти или открыть PDB-файл.
Поток 0x21b0 завершился с кодом 0 (0x0).
Поток 0x10a0 завершился с кодом 0 (0x0).
Поток 0x3054 завершился с кодом 0 (0x0).
Программа "[4056] shazhko07.exe" завершилась с кодом 0 (0x0).
```

Рисунок 2 - Результат очистки памяти

## Выводы

---

В ходе лабораторной работы были получены практические навыки использования шаблонов функций и классов на базе собственного кода.