

Тема 5: Абстрактные классы, интерфейсы, сериализация

Цель

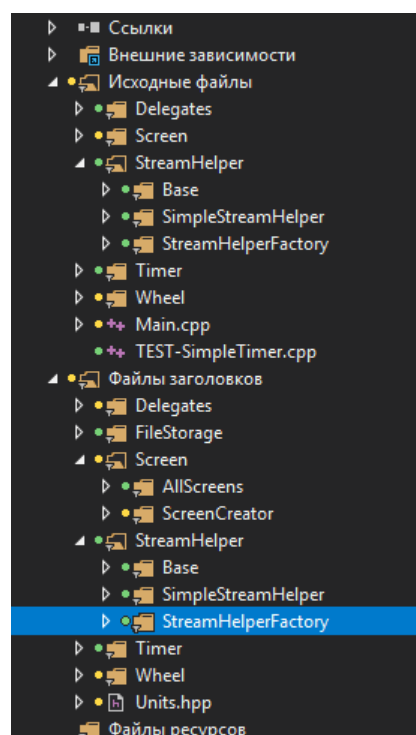
Научиться использовать интерфейсы для работы классов на примере сериализации

Общее задание

- Реализовать для каждого из классов данных своей иерархии возможность сохранения и загрузки данных с помощью класса CFileStorage, который выдается к лабораторной работе в виде библиотеки
- Показать бинарный дамп сохраненного файла
- В выводах отметить недостатки класса CFileStorage

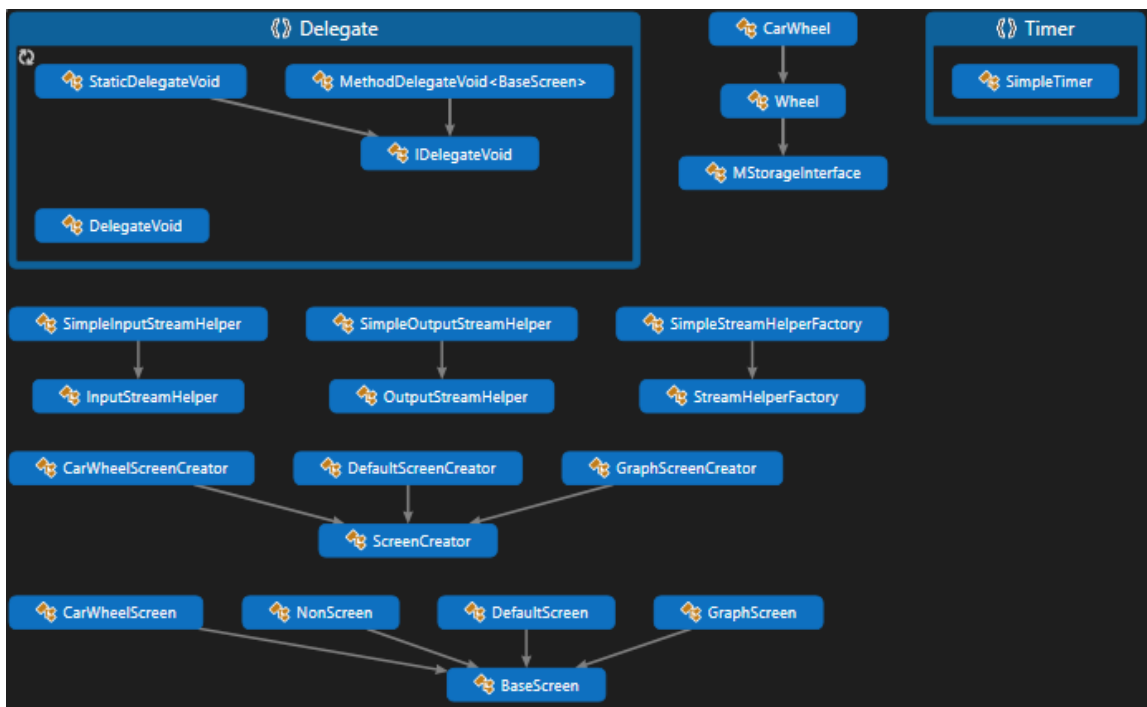
Структура проекта

Описание разработанных типов данных



▼ N Delegate	
C DelegateVoid	Класс для удобной работы с делегатами
C IDelegateVoid	Задаёт интерфейс
C MethodDelegateVoid	Класс реализующий интерфейс IDelegateVoid, для работы с функциями в классах
C StaticDelegateVoid	Класс реализующий интерфейс IDelegateVoid, для работы со статическими функциями
▼ N Timer	Интерфейс для работы с SimpleTimer
C SimpleTimer	Класс SimpleTimer
C abstract	Класс описывающий базовое отображение объектов
C CarWheel	Класс описывающий колесо машины
C CarWheelScreen	Класс описывающий отображение объектов класса CarWheelScreen
C CarWheelScreenCreator	Класс реализующий фабричный метод для создания объектов CarWheelScreen
C CFileStorage	
C DefaultScreen	Класс описывающий отображение объектов класса DefaultScreen
C DefaultScreenCreator	Класс реализующий фабричный метод для создания объектов DefaultScreen
C GraphScreen	Класс описывающий расширенное отображение объектов класса GraphScreen
C GraphScreenCreator	Класс реализующий фабричный метод для создания объектов GraphScreen
C MStorageInterface	
C NonScreen	Класс который используется в качестве заглушки
C SimpleInputStreamHelper	Вспомогательный класс для чтения параметров из потока
C SimpleOutputStreamHelper	Вспомогательный класс для записи параметров в поток
C SimpleStreamHelperFactory	Класс реализующий интерфейс абстрактной фабрики, для создания вспомогательных объектов ввода-вывода параметров из потока
C StreamHelperArg	Класс который используется для переадресации параметров между объектами класса StreamHelper
C Wheel	Класс описывающий колесо

Диаграмма классов



Разработанные методы и функции

Для работы с вводом-выводом данных была разработана абстрактная фабрика, которая в будущем позволит без труда вводить новые классы по работе с ввода-вывода

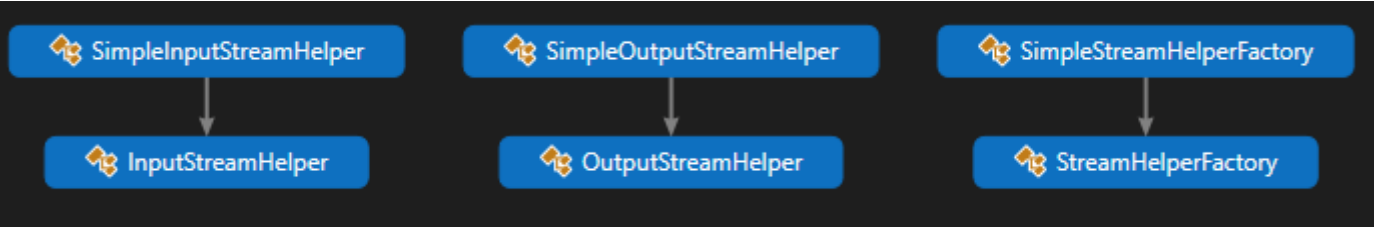


Рисунок 2.1 – Абстрактная фабрика

Ссылки на файлы проекта

Полный список файлов.

BaseScreen.cpp	Содержит реализацию класса BaseScreen
BaseScreen.h	Содержит объявление базового класса BaseScreen
CarWheel.cpp	Содержит реализацию класса CarWheel
CarWheel.h	Содержит объявление базового класса CarWheel
CarWheelScreen.cpp	Содержит реализацию класса CarWheelScreen
CarWheelScreen.h	Содержит объявление класса CarWheelScreen
CarWheelScreenCreator.hpp	Содержит объявление класса CarWheelScreenCreator
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
FileStorage.h	
GraphScreen.cpp	Содержит реализацию класса GraphScreen
GraphScreen.h	Содержит объявление класса GraphScreen
GraphScreenCreator.hpp	Содержит объявление класса GraphScreenCreator
IDelegateVoid.h	Объявление абстрактного класса IDelegateVoid
InputStreamHelper.h	Содержит объявление абстрактного класса InputStreamHelper
Main.cpp	
MethodDelegateVoid.hpp	Объявление и реализация класса MethodDelegateVoid
NonScreen.h	Содержит объявление класса NonScreen
OutputStreamHelper.h	Содержит объявление абстрактного класса OutputStreamHelper
ScreenCreator.h	
SimpleInputStreamHelper.cpp	Содержит реализацию класса SimpleInputStreamHelper
SimpleInputStreamHelper.h	Содержит объявление класса SimpleInputStreamHelper
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 реализацию класса SimpleTimer
Source.cpp	
StaticDelegateVoid.hpp	Объявление и реализация класса StaticDelegateVoid
StorageInterface.h	
StreamHelperArg.cpp	Содержит реализацию базового класса StreamHelperArg
StreamHelperArg.h	Содержит объявление базового класса StreamHelperArg
StreamHelperFactory.h	Содержит объявление абстрактного класса StreamHelperFactory
TEST-SimpleTimer.cpp	
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

```
int main() {
    Wheel *iWheel1 = new Wheel(6, 10, EUNITS_CENTIMETERS);
    CarWheel *iWheel2 = new CarWheel(0, 0, EUnits::EUNITS_CENTIMETERS, "", "");
    CFileStorage *file = CFileStorage::Create(*iWheel2, "Test.txt");
    file->Load();
    auto screenCreator = new ScreenCreator*[]{
        new CarWheelScreenCreator(iWheel2),
        new DefaultScreenCreator(iWheel1),
        new GraphScreenCreator(iWheel1) };
    RunCreatorScreen(screenCreator, 3);
    ОЧИСТКА ПАМЯТИ
    _CrtDumpMemoryLeaks();
    return 0;
}

void RunCreatorScreen(ScreenCreator **arrayScreenCreator, int count) {
    BaseScreen ** arrayScreen = new BaseScreen*[count];
    for (int i = 0; i < count; i++)
        arrayScreen[i] = arrayScreenCreator[i]->CreateScreen();
    RunPrintTimer(arrayScreen, count);
    ОЧИСТКА ПАМЯТИ
}

void RunPrintTimer(BaseScreen ** arrayScreen, int count) {
    Timer::SimpleTimer timer(4, 1000);
    for (int i = 0; i < count; i++)
        timer.OnTimerTact += Delegate::CreateDelegate<BaseScreen>(arrayScreen[i], &BaseScreen::ShowContent);
    timer.Start();
    timer.Wait(INFINITE);
}
```

Результаты работы

D:\Git\Hub\kit25a\ashazhko-artem\src\x64\Debug\shazhko05.exe

```
Diameter: 10
Width: 6
Units: MILLIMETRES
Volume: 471.238
Tire Manufacturer: X10
Tire Name: ALFA
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
-----Graphics-----
****
* * | *****
* * | *****
* * | *****
* * | *****
* * | *****
**** | *****
-----
Diameter: 10
Width: 6
Units: MILLIMETRES
Volume: 471.238
Tire Manufacturer: X10
Tire Name: ALFA
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
-----Graphics-----
****
* * | *****
* * | *****
* * | *****
* * | *****
* * | *****
**** | *****
-----
Diameter: 10
Width: 6
Units: MILLIMETRES
Volume: 471.238
Tire Manufacturer: X10
Tire Name: ALFA
```

```
Показать выходные данные из: Отладка
shazhko05.exe (Win32). Загружено "C:\Windows\System32\ucrtdll.dll". Невозможно найти или открыть PDB-файл.
"shazhko05.exe" (Win32). Загружено "C:\Windows\System32\ucrtdll.dll". Невозможно найти или открыть PDB-файл.
"shazhko05.exe" (Win32). Выгружено "C:\Windows\System32\ucrtdll.dll"
"shazhko05.exe" (Win32). Загружено "C:\Windows\System32\ucrtdll.dll". Невозможно найти или открыть PDB-файл.
"shazhko05.exe" (Win32). Выгружено "C:\Windows\System32\ucrtdll.dll"
Поток 0x380 завершился с кодом 0 (0x0).
Поток 0x1384 завершился с кодом -1073741510 (0xc000013a).
Поток 0x18e8 завершился с кодом -1073741510 (0xc000013a).
Поток 0xef8 завершился с кодом -1073741510 (0xc000013a).
Поток 0x2350 завершился с кодом -1073741510 (0xc000013a).
Программа "[1652] shazhko05.exe" завершилась с кодом -1073741510 (0xc000013a).
```

```
diameter : 10
units : 1
width : 6
tireManufacturer : X10
tireName : ALFA
```

```
[=====] Running 4 tests from 4 test cases.
[-----] Global test environment set-up.
[-----] 1 test from CarWheel_Store
[ RUN      ] CarWheel_Store.saveCarWheel
[ OK       ] CarWheel_Store.saveCarWheel (8 ms)
[-----] 1 test from CarWheel_Store (9 ms total)

[-----] 1 test from CarWheel_Load
[ RUN      ] CarWheel_Load.loadCarWheel
[ OK       ] CarWheel_Load.loadCarWheel (4 ms)
[-----] 1 test from CarWheel_Load (4 ms total)

[-----] 1 test from Wheel_Store
[ RUN      ] Wheel_Store.saveWheel
[ OK       ] Wheel_Store.saveWheel (5 ms)
[-----] 1 test from Wheel_Store (6 ms total)

[-----] 1 test from Wheel_Load
[ RUN      ] Wheel_Load.loadWheel
[ OK       ] Wheel_Load.loadWheel (2 ms)
[-----] 1 test from Wheel_Load (4 ms total)

[-----] Global test environment tear-down
[=====] 4 tests from 4 test cases ran. (27 ms total)
[ PASSED  ] 4 tests.
```

Выводы

В ходе лабораторной работы были получены практические навыки работы с библиотекой CFileStorage