

Тема 4: Статические методы, перегрузка операторов и методов

Цель

Научиться уместно использовать статические методы, использовать перегрузку операторов и методов

Общее задание

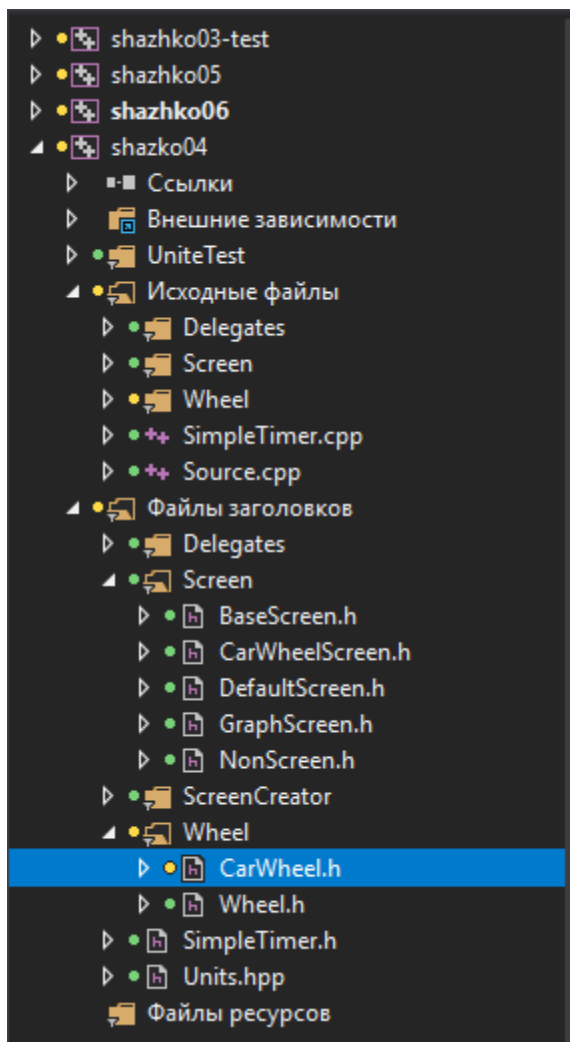
- Определить в классе статический метод OnTimerAction. Данный метод должен отображать на экране
- Выбрать для Win32-таймера собственный интервал повторных вызовов
- Установить реализованный метод OnTimerAction на вызов в таймере
- Таймер должен срабатывать всего 4 раза
- Метод должен выводить на экран данные про полученный объект
- Дополнительные 4 балла добавляются за отдельную реализацию класса по работе с таймером
- Реализовать перегруженные операторы и методы в соответствии с индивидуальным заданием

Прикладная область

В соответствии с вариантом #16

Прикладная область	Перегружаемые методы Data2	Перегружаемые операторы
Колесо	setData	== в Data2 для сравнение диаметра =Data2 устанавливает производителя и название

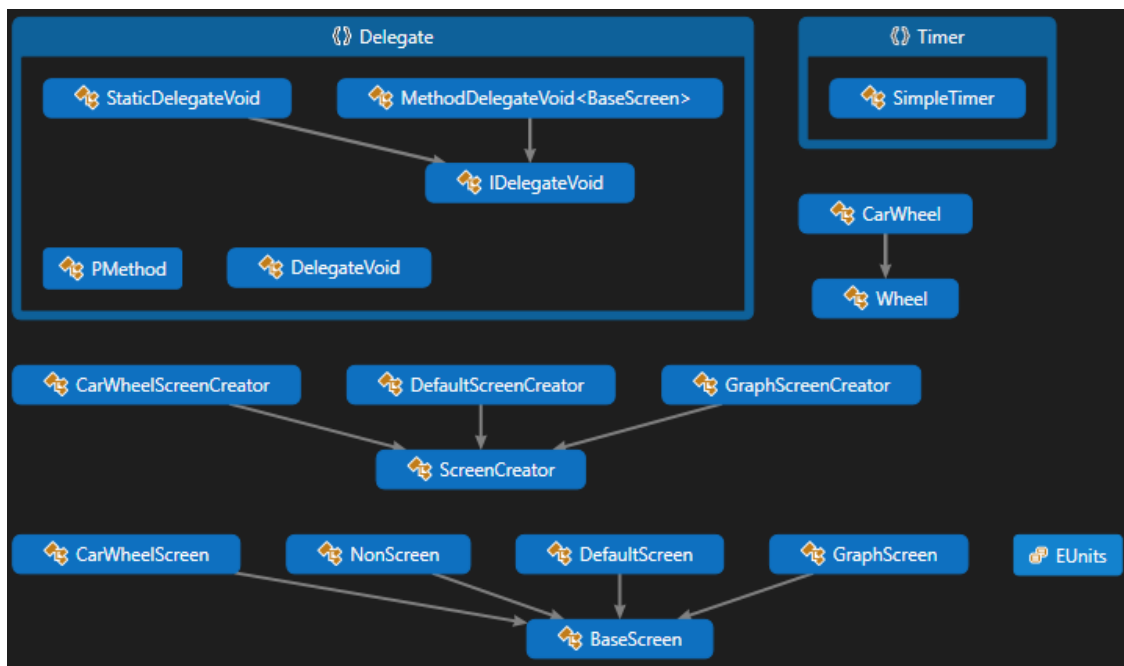
Структура проекта



Описание разработанных типов данных

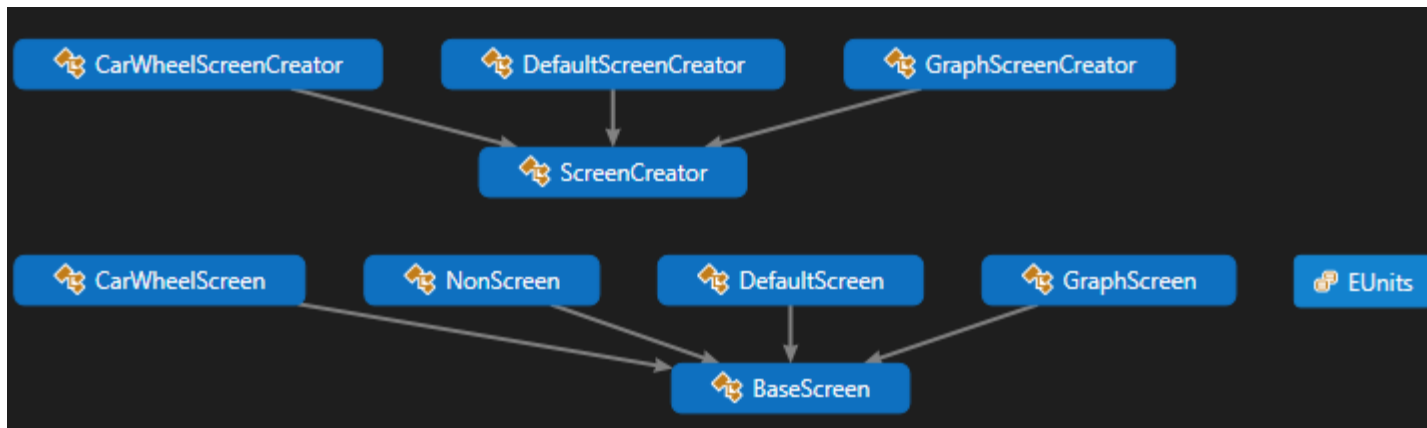
▼ N Delegate	
C DelegateVoid	Класс для удобной работы с делегатами
C IDelegateVoid	Задаёт интерфейс
C MethodDelegateVoid	
C StaticDelegateVoid	Класс реализующий интерфейс IDelegateVoid , для работы со статическими фун
▼ N Timer	Интерфейс для работы с SimpleTimer
C SimpleTimer	Класс SimpleTimer
C abstract	Класс описывающий базовое отображение объектов
C CarWheel	
C CarWheelScreen	Класс описывающий отображение объектов класса CarWheelScreen
C CarWheelScreenCreator	Класс реализующий фабричный метод для создания объектов CarWheelScreen
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 Wheel
C GraphScreenCreator	Класс реализующий фабричный метод для создания объектов GraphScreen
C NonScreen	Класс который используется в качестве заглушки
C Wheel	Класс описывающий колесо

Диаграмма классов



Разработанные методы и функции

Был разработан фабричный метод



Ссылки на файлы проекта

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
GraphScreen.cpp	Содержит реализацию класса GraphScreen
GraphScreen.h	Содержит объявление класса GraphScreen
GraphScreenCreator.hpp	Содержит объявление класса GraphScreenCreator
IDelegateVoid.h	Объявление абстрактного класса IDelegateVoid
MethodDelegateVoid.hpp	
NonScreen.h	Содержит объявление класса NonScreen
ScreenCreator.h	Содержит объявление класса ScreenCreator
SimpleTimer.cpp	Содержит реализацию класса SimpleTimer
SimpleTimer.h	Содержит объявление класса SimpleTimer
Source.cpp	
StaticDelegateVoid.hpp	Объявление и реализация класса StaticDelegateVoid
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);

    ScreenCreator *creatorDefaultScreen = (ScreenCreator*)new DefaultScreenCreator(iWheel1);
    ScreenCreator *creatorGraphScreen = (ScreenCreator*)new GraphScreenCreator(iWheel1);

    CarWheel *iWheel2=new CarWheel(6, 10, EUNITS_CENTIMETERS, "X10", "Canon");
    ScreenCreator *creatorCarWheelScreen = (ScreenCreator*)new CarWheelScreenCreator(iWheel2);

    BaseScreen *screenD = creatorDefaultScreen->CreateSreen();
    BaseScreen *screenG = creatorGraphScreen->CreateSreen();
    BaseScreen *screenC = creatorCarWheelScreen->CreateSreen();
    {
        Timer::SimpleTimer timer(4, 1000);

        timer.OnTimerTact += Delegate::CreateDelegate<BaseScreen>(screenD, &BaseScreen::ShowContent);
        timer.OnTimerTact += Delegate::CreateDelegate<BaseScreen>(screenG, &BaseScreen::ShowContent);
        timer.OnTimerTact += Delegate::CreateDelegate<BaseScreen>(screenC, &BaseScreen::ShowContent);

        timer.Start();
        timer.Wait(INFINITE);
    }
    ОЧИСТКА ПАМЯТИ
    _CrtDumpMemoryLeaks();

    return 0;
}
```

Результаты работы

```
D:\GitHub\kit25a\ashazhko-artem\src\x64\Debug\shazko04.exe
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
-----Graphics-----
****
*   *   *
*   *   *
*   *   *
*   *   *
****
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
Tire Manufacturer: Canon
Tire Name: X10
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
-----Graphics-----
****
*   *   *
*   *   *
*   *   *
*   *   *
****
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
Tire Manufacturer: Canon
Tire Name: X10
Diameter: 6
Width: 10
Units: CENTIMETERS
Volume: 282.743
-----Graphics-----
****
*   *   *
*   *   *
*   *   *
*   *   *
****
```

```
Вывод
Показать выходные данные из: Отладка
shazko04.exe (Win32). Загружено "C:\Windows\System32\user32.dll". Невозможно найти или открыть PDB-файл.
"shazko04.exe" (Win32). Выгружено "C:\Windows\System32\user32.dll". Невозможно найти или открыть PDB-файл.
"shazko04.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll". Невозможно найти или открыть PDB-файл.
"shazko04.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll". Невозможно найти или открыть PDB-файл.
"shazko04.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll". Невозможно найти или открыть PDB-файл.
Поток 0xa94 завершился с кодом 0 (0x0).
"shazko04.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll". Невозможно найти или открыть PDB-файл.
"shazko04.exe" (Win32). Загружено "C:\Windows\System32\kernel.appcore.dll". Невозможно найти или открыть PDB-файл.
Поток 0x2e70 завершился с кодом 0 (0x0).
Поток 0x3374 завершился с кодом 0 (0x0).
Поток 0x2088 завершился с кодом 0 (0x0).
Программа "[1728] shazko04.exe" завершилась с кодом 0 (0x0).
```

Выводы

В ходе лабораторной работы были получены практические навыки работы с Win32-таймером; реализована система событий для взаимодействия с таймером