Оглавление

[САОД. Задание 5 1](#_Toc54179658)

[Темы 1](#_Toc54179659)

[Предварительные соображения 1](#_Toc54179660)

[Задания 1](#_Toc54179661)

[Упражнения для самостоятельного изучения 3](#_Toc54179662)

[Отчет 4](#_Toc54179663)

# САОД. Задание 5

## Темы

* Порядок создания и разрушения объектов;
* Статические поля класса;
* Передача объектов как параметров;
* Порядок создания и разрушения составных объектов и их предков или полей;
* Работа с динамическими данными;
* Объявление и использование шаблонов в C++.

***Это задание является существенным шагом в подготовке к промежуточному тестированию***.

## Предварительные соображения

Если для класса определен конструктор и деструктор, то они вызываются при создании и уничтожении объекта. Таким образом, если в конструкторы и деструктор поместить оператор вывода на консоль,

cout << "…" << endl;

то при выполнении программы можно следить за созданием, копированием и уничтожением объектов, а, следовательно, и наблюдать за временем их жизни.

Используем это обстоятельство в ряде следующих упражнений, направленных на закрепление и расширение представлений о создании и времени жизни объектов в C++. Такие знания особенно ценны, если учесть, что они идентичны как для объектов разработанных классов, так и для переменных встроенных типов.

По мере движения по заданию консоль будет загромождаться и поведение объектов будет становиться не очевидным. Закомментируйте код, который создает паразитные сообщения.

В данном задании часто не указан явный текст C++, а только задача. Постарайтесь создавать соответствующий код самостоятельно. И только если это не получается, обращайтесь для примеров к выполненным ранее заданиям (2-4).

## Задания

1. Подготовительные действия:
   1. Создайте новый проект LifeTime и добавьте в него операции ввода-вывода на консоль. ***Не забывайте делать это на рабочем столе.***
   2. Добавьте при помощи мастера класс Test. На этот раз нам понадобится и файл Test.cpp. Доработайте деструктор и конструктор класса так, чтобы они сообщали о выполнении на консоль. Далее такие методы будут назваться “*говорящими”*.

Для этого, если вы не используете прекомпилированные заголовки, добавить “#include <iostream>” и “using namespace std;” в “Test.h”.

* 1. Объявите в main объект типа Test. Не забудьте включить заголовочный файл “Test.h”. Соберите и выполните проект. Если необходимо, устраните ошибки. Убедитесь, что объект создается и уничтожается автоматически.

1. Добавьте в main необходимые строки вывода, так что бы одна из них появилась до создания объекта, а другая после.
   1. Попробуйте сделать то же самое с уничтожением объекта. Если необходимо, слегка модифицируйте main.
2. Добавьте в класс статическое поле static int nCount. В нем мы будем считать количество созданных объектов. Вспомните правило объявления и создания статических полей (Лекция 2).
   1. Увеличивайте nCount каждый раз при вызове конструктора и уменьшайте на 1 при вызове деструктора.
   2. Добавьте вывод nCount в конструкторе и деструкторе.
3. Объявите глобальный объект Test (перед main). При помощи аналогичных экспериментов попытайтесь установить его время жизни.
4. Создайте динамический объект типа Test \*p = new Test; Выполните проект. Попытайтесь обнаружить следы вызова деструктора. Как согласуются сообщения от конструктора и деструктора и ваши представления о создании и уничтожении объектов.
5. Добейтесь, что бы вызывался деструктор созданного динамического объекта.
6. Создайте динамический массив объектов Test. Добейтесь, что бы деструктор вызывался для каждого из созданных объектов.
7. Создайте “говорящий” конструктор копирования для класса Test и напишите функцию void foo(Test t), которая получает в качестве параметра объект класса Test. Не забудьте увеличить счетчик объектов и при копировании. Пусть функция выводит на консоль сообщение о своем выполнении – “foo is running” или что-то подобное.
8. Вызовите foo с одним из объектов Test из main и понаблюдайте за созданием и уничтожением объектов. Объясните ваши наблюдения
9. Создайте еще один класс Child: public Test, который является потомком от Test. Наделите его конструктор и деструктор способностью сообщать о своем выполнении.

Создайте объект Child в main и понаблюдайте за порядком создания предка и объекта дочернего класса.

1. Создайте еще один класс Aggregate, который включает объект типа Test как Test m\_objTest. Наделите его конструктор и деструктор способностью сообщать о своем выполнении.

Создайте объект Aggregate в main и понаблюдайте за порядком создания объекта агрегата и его полей, а также с порядком их разрушения.

1. Создайте класс-шаблон, например, AggregateT – от “Aggregate Template”. *Не забывайте, что весь шаблон (и реализация) должен располагаться в заголовочном файле!*

Пусть он параметризуется типом поля данных (то что в Aggregate m\_objTest). Наделите его конструктор и деструктор способностью сообщать о своем выполнении.

Создайте в main объект AggregateT парметризованный классом Test и понаблюдайте за порядком создания объекта агрегата-шаблона и его полей, а также с порядком их разрушения.

Создайте в main объект AggregateT парметризованный классом Child и понаблюдайте за порядком создания объекта агрегата-шаблона и его полей, а также с порядком их разрушения.

**Теперь мы изучили на практике**:

* Порядок создания и разрушения объектов;
* Судьбу копий объектов при передаче их параметрами по значению;
* Порядок создания и разрушения составных объектов и их предков или полей;
* Как разместить статические поля класса;
* Правила работы с динамическими данными;
* Как создавать и использовать шаблоны.

# Упражнения для самостоятельного изучения

Убедитесь в полноте понимания аспектов создания и разрушений объектов в C++ и C#, попробуйте ответить на приведенные ниже вопросы. Если ответ вам кажется неочевидным, то поставьте соответствующие эксперименты, используя методику говорящих конструкторов и деструкторов. Тест дополнительных упражнений и проектов оставьте в решении для проверки.

**Вопросы**:

1. Какие случаи неявного копирования объектов существуют в C++? (их 2)
2. В каком порядке создаются составные объекты (потомки или агрегаты в C++)?
3. Для чего предназначен интерфейс IDisposable в C#?
4. Будет ли вызван Dispose по завершению блока using, если на объект, указанный в using имеется еще одна ссылка?

*Подтвердите это экспериментом (создайте простенький проект на C# с классом Test, который реализует интерфейс IDispoasable).*

1. Как вызвать конструктор предка с параметром в C++?

*Вы уже убедились, что конструктор предка работает до конструктора потомка. При этом, сработает конструктор по умолчанию, т.е. без параметров. Тогда в конструкторе потомка придется изменять поля предка, что неэффективно и иногда сложно. Как можно добиться того, что вместо конструктора предка по умолчанию будет вызван конструктор с параметрами. Проверьте это на эксперименте (добавьте в Test конструктор с параметром пусть при создании Child и при создании пусть будет именно он).*

1. Как вызвать конструктор предка с параметром в C#?
2. Как вызвать конструктор члена класса с параметром?

*Вы уже убедились, что конструктор члена класса-агрегата работает до конструктора самого агрегата. При этом, сработает конструктор по умолчанию, т.е. без параметров. Тогда в конструкторе агрегата придется изменять поля члена класса, что неэффективно и иногда сложно. Как можно добиться того, что вместо конструктора члена класса по умолчанию будет вызван конструктор с параметрами. Проверьте это на эксперименте.*

1. Опишите различие в понятии *ссылка* в смысле C++ и в смысле .Net.
2. Как объявить и использовать шаблон класса?

# Отчет

Подготовьте отчет в Word по проделанной работе и отправьте как ответ на задание. Отчет должен включать краткое описание (не более 2-х стр):

* времени жизни объектов в C++ и C# и
* ответы на поставленные в конце задания вопросы.

Ответ на задание состоит из 2-х единиц – отчета и архива с решением, содержащим тексты ваших упражнений.

**Важно!** Не забудьте очистить решение VS и все проекты от временных и двоичных файлов. Объем архива не должен превышать 300 Kb. Архивируйте сразу весь каталог решения: Правой кнопкой мыши -> Отправить -> Сжатая папка. Формат архива должен быть zip.

Поместите отчет и архив в качестве ответа на задание.