

# Тема 1: Стековые объекты. Конструктор, деструктор, отображение, передача

## Цель

Научиться создавать объекты. Получить навыки создания объекта на стеке а также передачу объекта по значению

## Общее задание

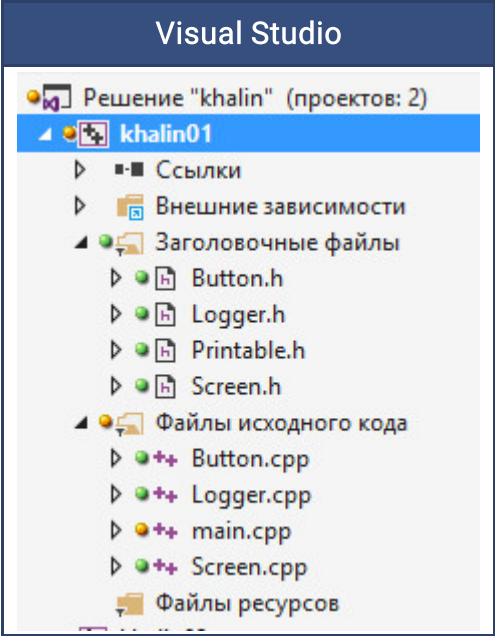
- Создать класс данных и класс отображения данных
- Объект отображения конструируется на стеке функции main() объектом данных, который заранее создан на стеке
- Передавать класс данных как значение
- Класс данных имеет все публичные поля и методы
- Класс отображения только выполняет отображения данных
- В соответствии с индивидуальным заданием определить класс по варианту, разработать программу, которая демонстрирует использование классов

## Прикладная область

В соответствии с вариантом #15

Прикладная область	Имя класса данных	Поля класса данных	Отображение
Клавиатура	Кнопка	state (true / false)	text format

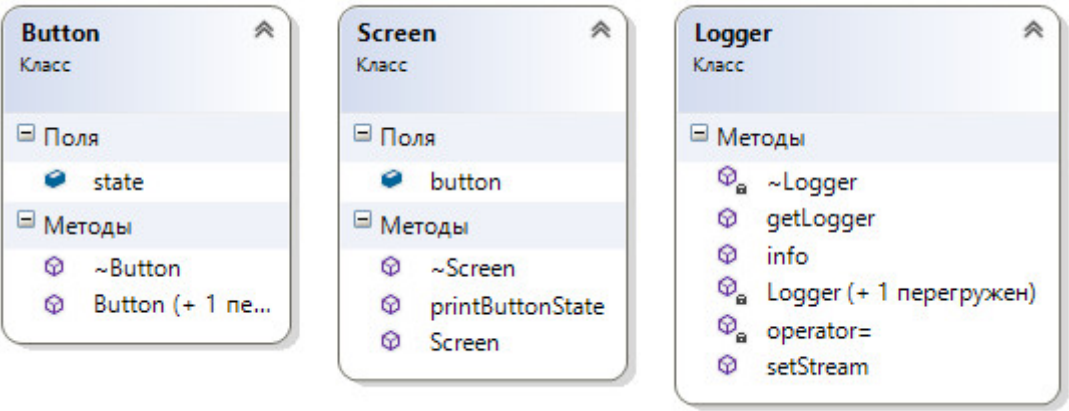
## Структура проекта



## Описание разработанных типов данных

Classes	
<b>Button</b>	Represents an entity with two states
<b>Logger</b>	Provides methods for logging information
<b>Printable</b>	The objects whose class is extended by this will be able to be printed to the output stream using 'print' method
<b>Screen</b>	Displays state of the passed button

## Диаграмма классов



Описание разработанных методов и функций

Class	Methods
Button	<div><div>Public Member Functions</div><div><div>bool <b>getState</b> ()</div><div>void <b>setState</b> (bool state)</div><div>void <b>print</b> (std::ostream &amp;out) Prints a state of the current object to the output stream. <a href="#">More...</a></div></div></div>
Screen	<div><div>Public Member Functions</div><div><div><b>Screen</b> (<b>Button</b> btn)</div><div>void <b>printButtonState</b> (std::ostream &amp;out) prints button state into an output stream <a href="#">More...</a></div></div></div>
Logger	<div><div>Public Member Functions</div><div><div>void <b>info</b> (char *msg) prints msg</div><div>void <b>setStream</b> (std::ios_base *os)</div></div><div><div>Static Public Member Functions</div><div><div>static <b>Logger</b> &amp; <b>getLogger</b> () returns a logger instance</div></div></div></div>
Printable	<div><div>Public Member Functions</div><div><div>virtual void <b>print</b> (std::ostream &amp;out)=0 Prints a state of the current object to the output stream. <a href="#">More...</a></div></div></div>

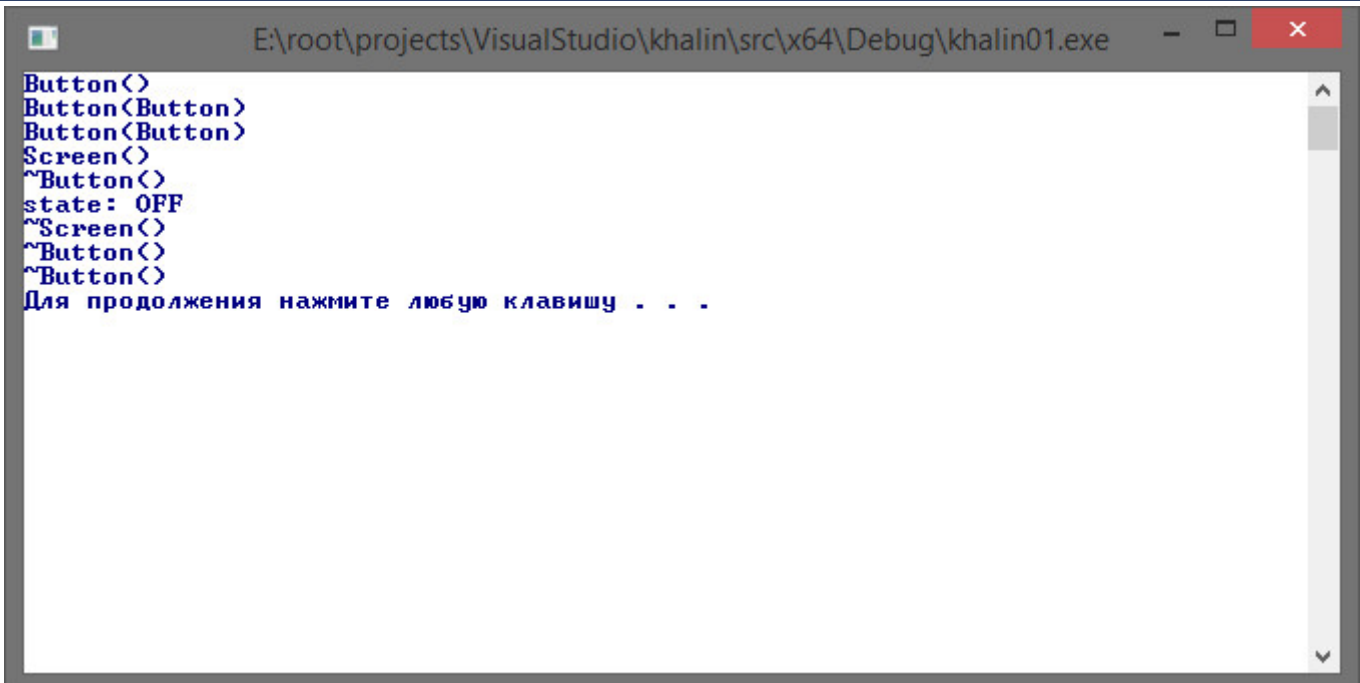
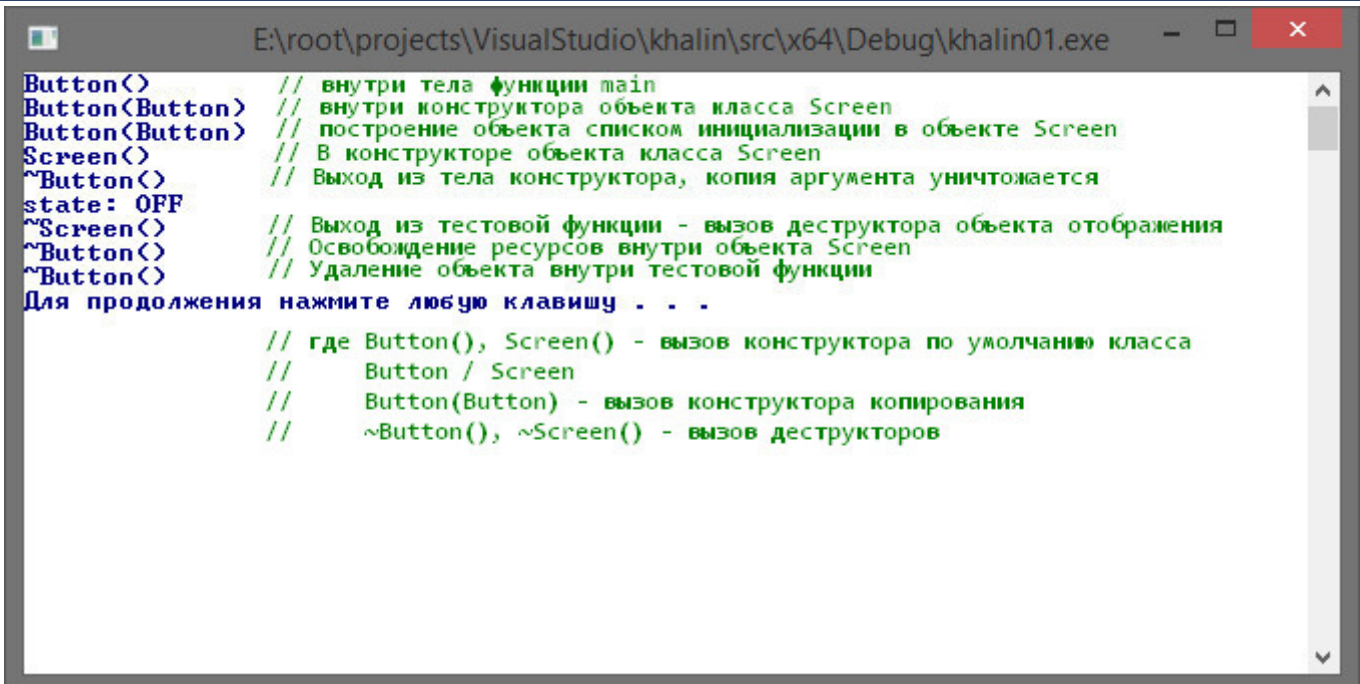
Ссылки на файлы проекта

Имя файла	Описание
Button.h	Contains <a href="#">Button</a> class declaration
Screen.h	Contains <a href="#">Screen</a> class declaration
Logger.h	Contains <a href="#">Logger</a> class declaration
Button.cpp	Contains <a href="#">Button</a> class implementation
Screen.cpp	Contains <a href="#">Screen</a> class declaration
Logger.cpp	Contains <a href="#">Logger</a> class declaration

Текст программы

Имя файла	Описание
Button	<a href="#">Button</a> class implementation
Screen	<a href="#">Screen</a> class implementation
Logger	<a href="#">Logger</a> class implementation

Результаты работы

Описание	Изображение
Выполнение программы	
Описание вывода	

Выводы

В ходе выполнения лабораторной работы было проведено исследование жизненного цикла объектов, созданных в стеке:

- При передаче объекта в функцию в качестве аргумента по значению, создается его локальная копия с вызовом конструктора копирования
- При выходе из зоны видимости места инициализации осуществляется вызов деструктора созданного объекта, вызывающего деструкторы размещенных в нем объектов с помощью композиции