**Министерство образования республики Молдова Технический университет Молдовы**

**Кафедра «Автоматика и информационные технологии»**

**Отчёт**

**Лабораторная работа № 2**

**Тема: Структура – механизм абстракции**

**Вариант 5**

**Выполнил студент гр. TI-197** Макеев Никита

**Проверил:** Брынзан Л.

Оглавление

[ЛАБОРАТОРНАЯ РАБОТА №2 3](#_Toc52622691)

[Задание: 3](#_Toc52622692)

[Реализация задания: 3](#_Toc52622693)

[(Листинг номер 1). 3](#_Toc52622694)

[(Листинг программы 2). 4](#_Toc52622695)

[(Листинг программы 3,4). 5](#_Toc52622696)

[(Листинг номер 5). 6](#_Toc52622697)

[(Листинг номер 6). 7](#_Toc52622698)

[(Листинг номер 7). 8](#_Toc52622699)

[(Листинг номер 8). 9](#_Toc52622700)

[Код лабораторной работы доступен по ссылке https://github.com/Randwow/oop\_lab2 10](#_Toc52622701)

[Вывод на экран: 10](#_Toc52622702)

[Вывод: 10](#_Toc52622703)

# ЛАБОРАТОРНАЯ РАБОТА №2

**Тема:** Конструктор – функция инициализации объектов класса

**Цели работы:**

* изучение основ определения и использования конструкторов;
* изучение основ определения и использования деструкторов;
* изучение типов конструкторов;

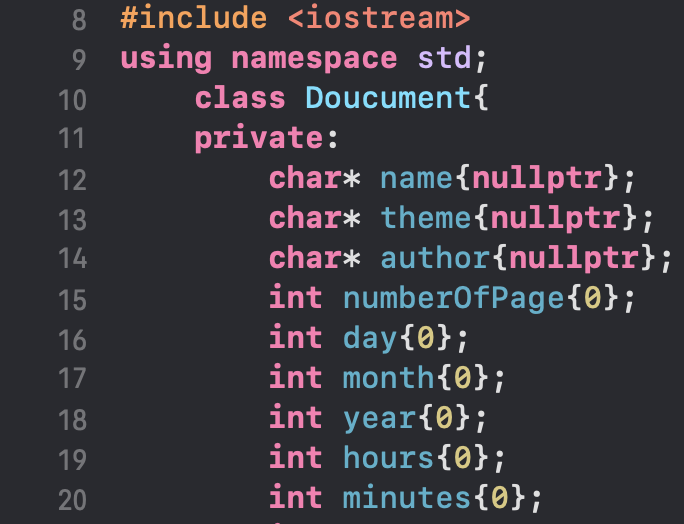
## **Задание:**

Создать класс *Document* – документ, содержащий информацию о названии, теме, авторе документа используя динамическую память, количество страниц, дату и время последней редакции. Определить все конструкторы: конструктор приведения типа - параметр обозначающий название документа. Определить функции переназначение темы, изменение даты последней редакции и др.

# Реализация задания:

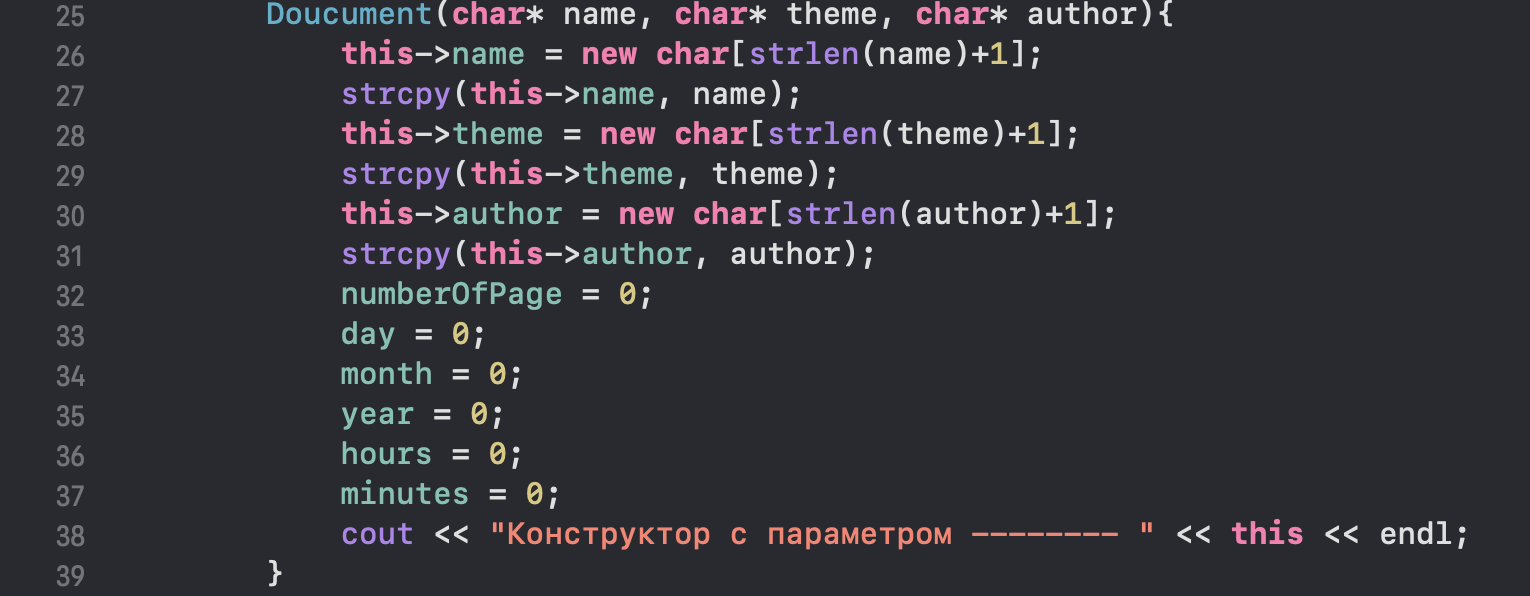
Создаем класс с называнием документ , у которой объявляем переменные с названием документа, темой, автором, и временем. Для времени используем 6 переменных, потому что в задании нужно реализовать изменение времени сохранения документа.

( см.листинг 1).Данные поля имеют модификаторы private для того чтобы у пользователя не было доступа к ним.



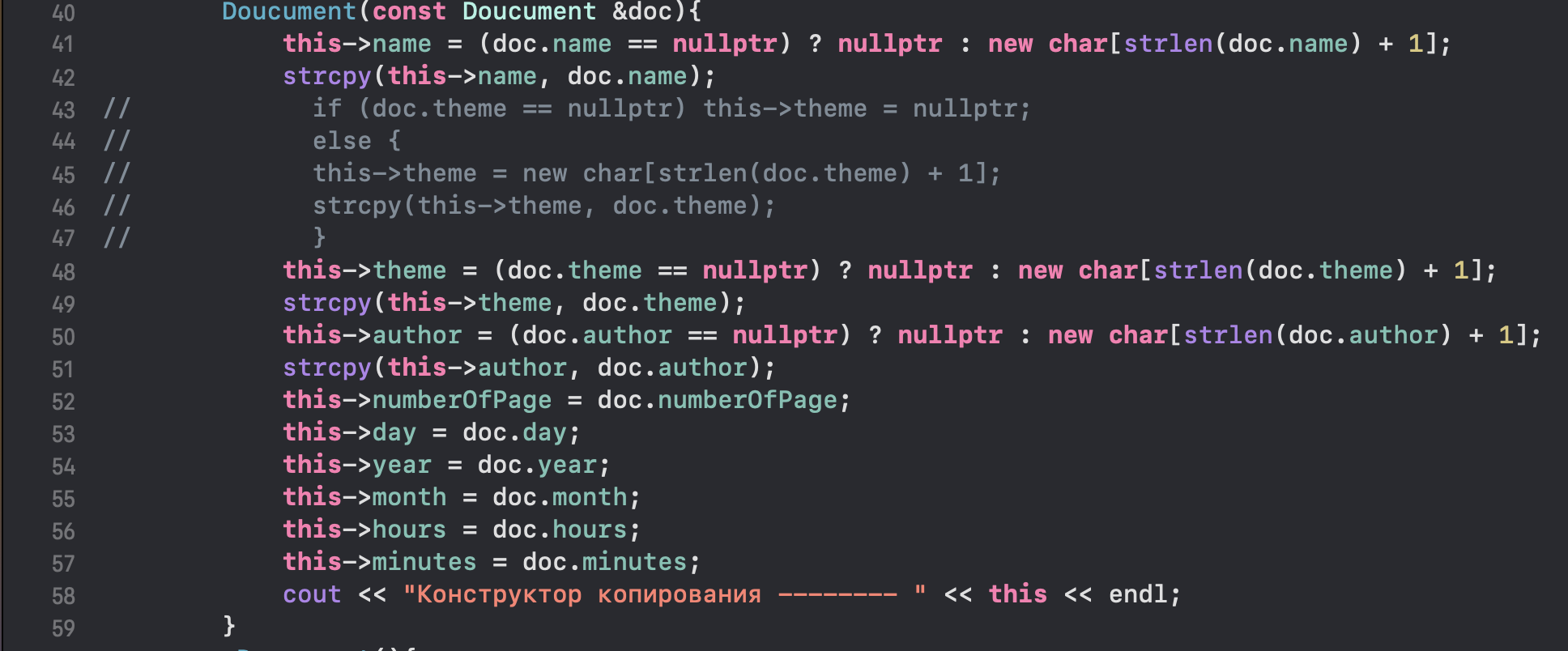
## **(Листинг номер 1).**

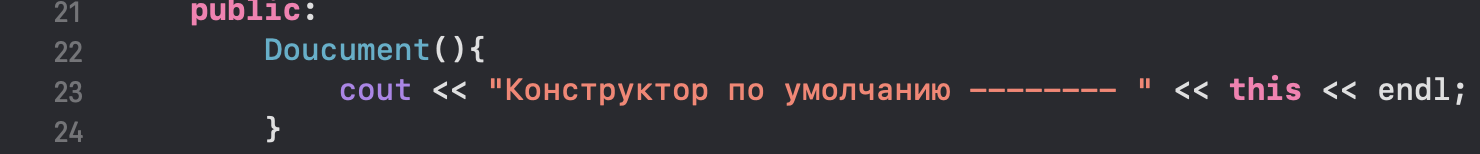
Следующие этапы – это создание конструкторов.А именно, конструктора параметров, копирования и по умолчанию, а также диструктора для удаления памяти (см листинг номер 2,3,4,5).



## **(Листинг программы 2).**

Конструктор с параметрами нужен для того чтобы экземпляры нашего класса имели определенные значения, которые мы предоставим им.





## **(Листинг программы 3,4).**

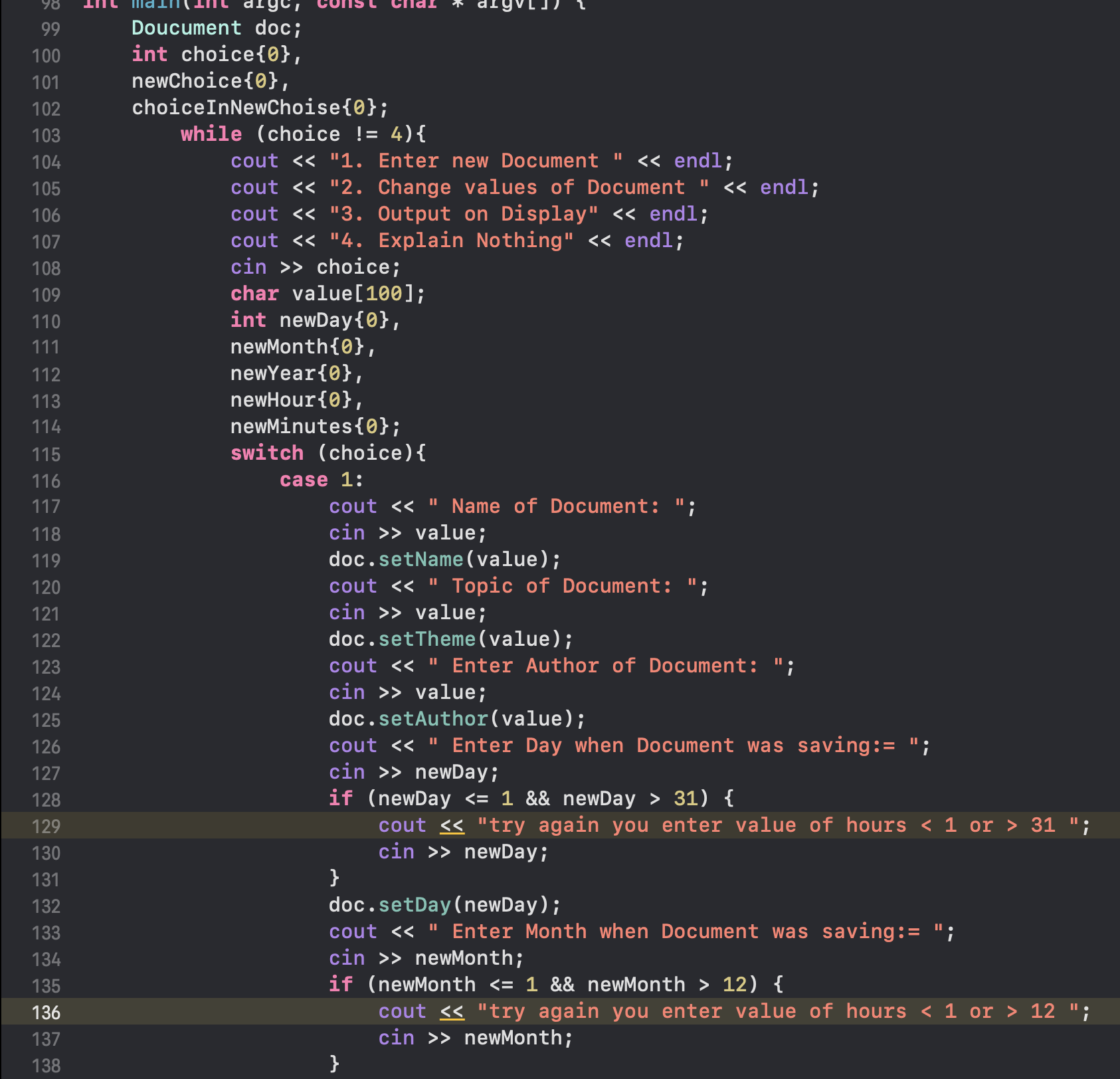
Листинг программы 3 – это конструктор копирования - это особый тип конструктора, который используется для создания нового объекта через копирование существующего объекта. И, как в случае с конструктором по умолчанию, если вы не предоставите конструктор копирования для своих классов самостоятельно, то язык C++ создаст [**public**](https://ravesli.com/urok-114-spetsifikatory-dostupa-public-i-private/)-конструктор копирования автоматически. В данном конструкторе был использован тернарный оператор для копирования уже существующего объекта в новый, это сделано для того чтобы сократить кол-во кода и облегчить его “Читаемость”.Листинг программы номер 4 – это конструктор по умолчанию, если его убрать компилятор сам создаст его.



## **(Листинг номер 5).**

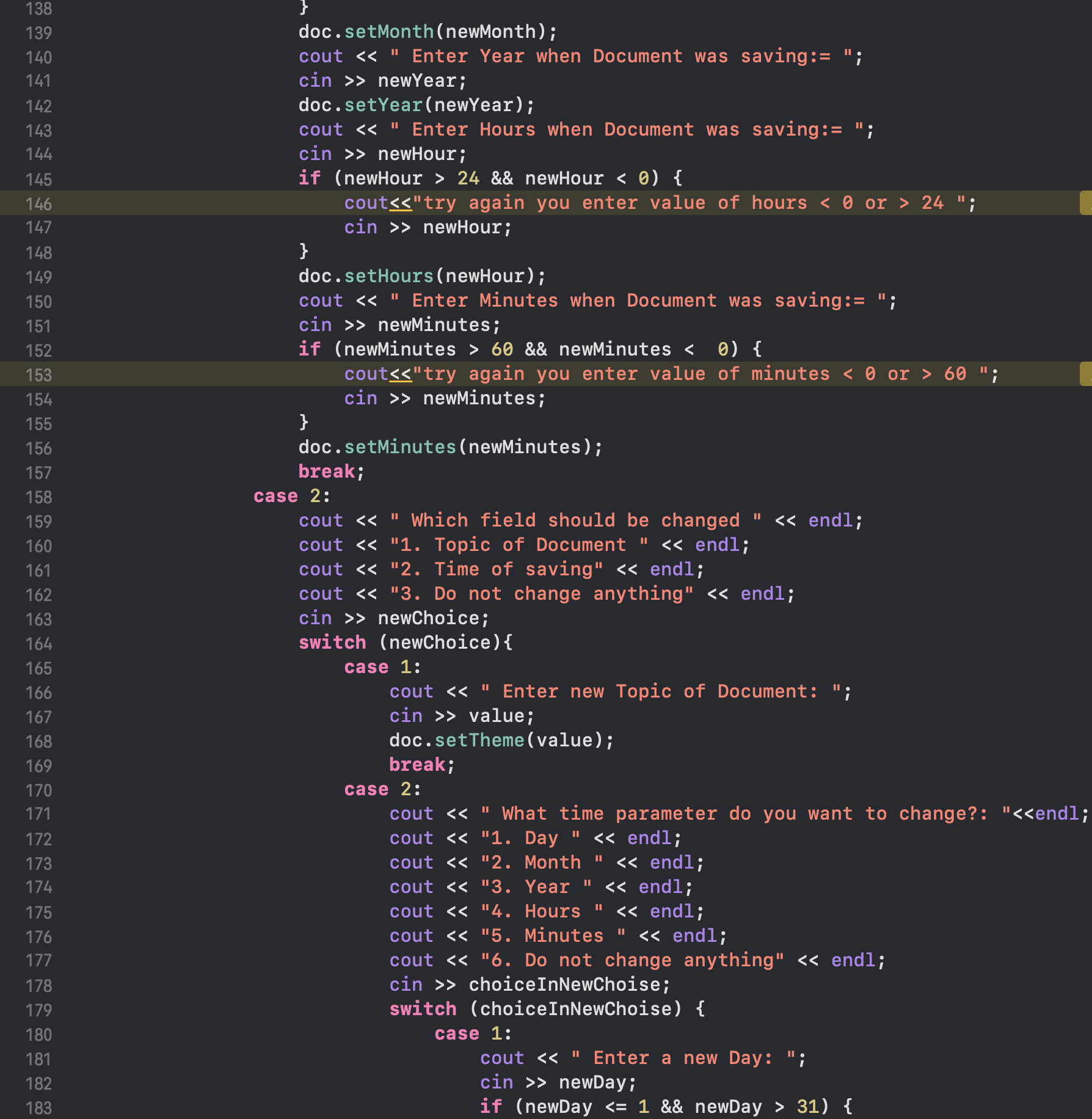
В данном листинге реализуется диструктор ~Document(){…}, он служит для того, чтобы при завершении программы очистить память, именно для того чтобы не было утечек памяти. Также как и с конструктором по умолчанию если его не будет, то компилятор сгенерирует его. Кроме диструктора в программе представлены еще sette’ы для получения доступа к нашим полям класса, а именно для их изменения.

Второй этап – это оформление программы в main (см листинг номер 6,7,8).



## **(Листинг номер 6).**

В главной функции main, я инициализировал несколько вспомогательных переменных, которые в дальнейшем помогут реализовать оформление программы



## **(Листинг номер 7).**

Само оформление сделано с помощью оператора множественного выбора **switch** (<#expression#>) {

**case** <#constant#>:

<#statements#>

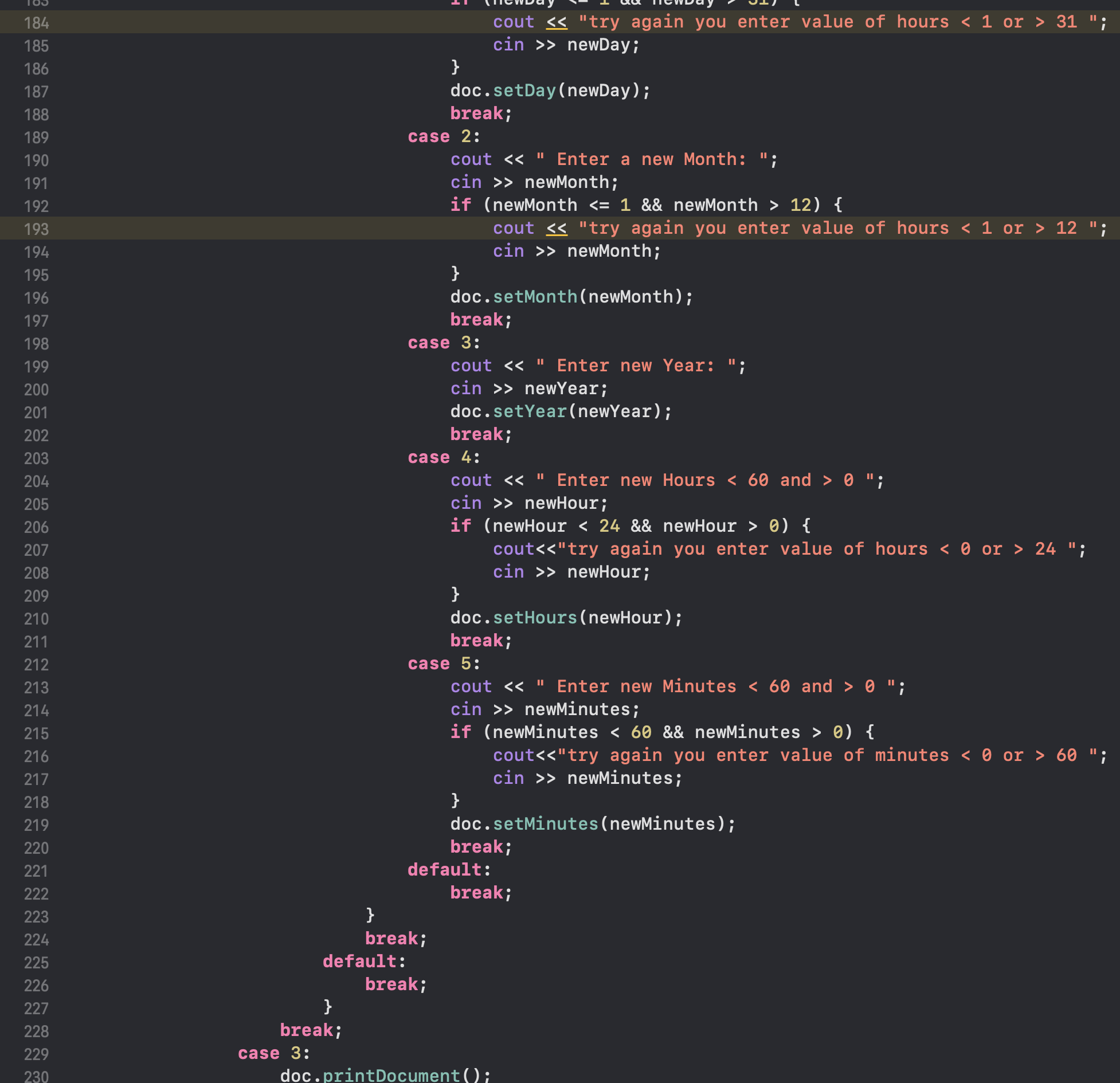
**break**;

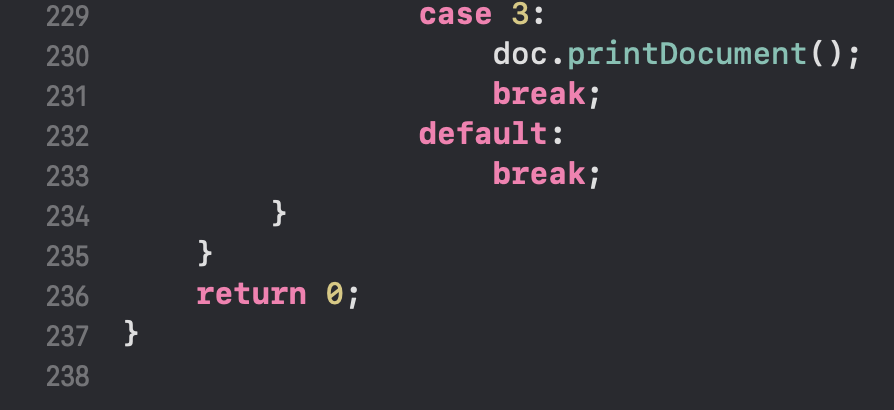
**default**:

**break**;

}

Что помогло сделать, удобный интерфейс для пользователя, а именно для того, чтобы он выбрал какое действие хочет соверишть





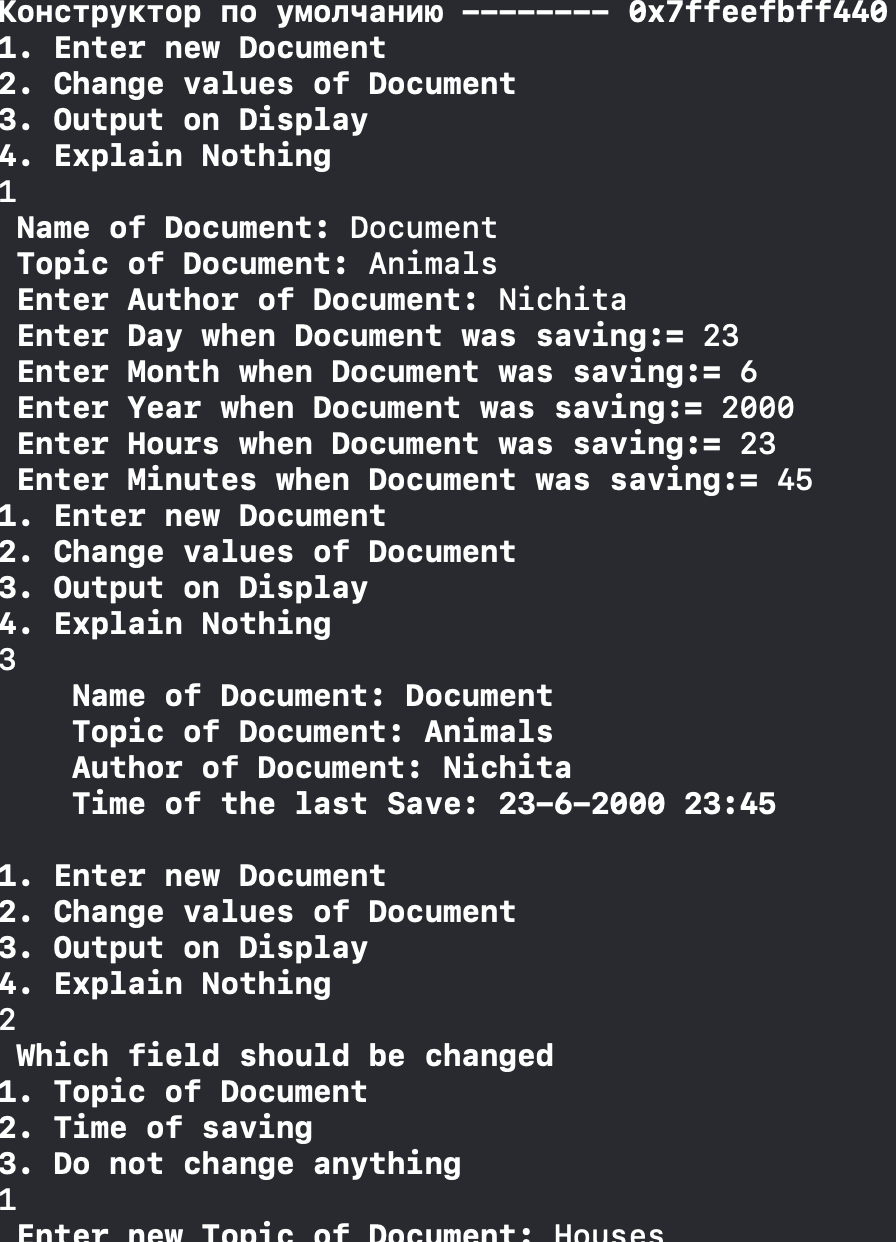
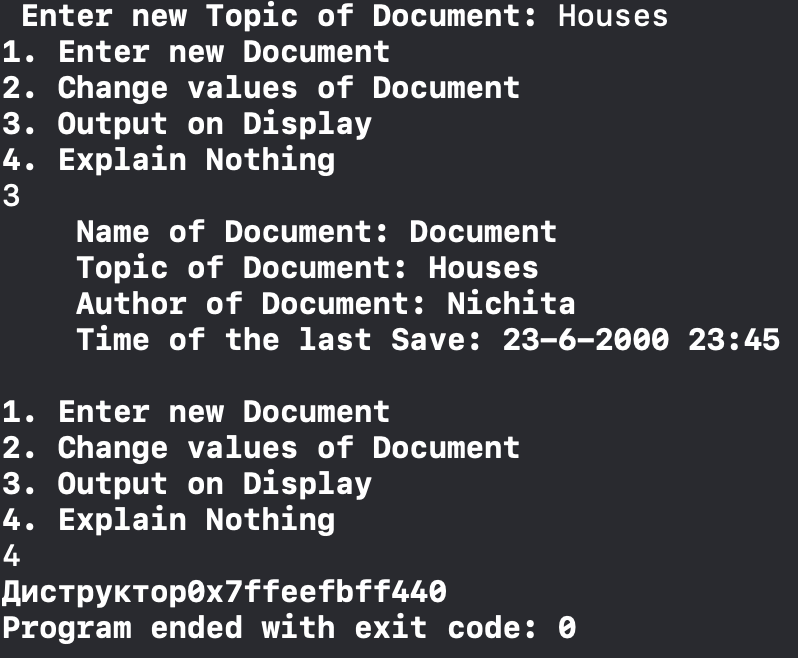
## **(Листинг номер 8).**

Окончание действия оператора Switch и завершение работы программы

# 

# Код лабораторной работы доступен по ссылке https://github.com/Randwow/oop\_lab2

## **Вывод на экран:**

**** ****

# Вывод:

В данной лабораторной работе былизучен абстрактный тип данных, изучены правила определения и использования структур данных, создание переменных типа структуры, обращение к полям и изучение принципов программирования, основанное на работе со структурами. Также были реализованы и изучены методы get и set, функции очистки памяти, сортировки массива структур, инициализация, изменения данных и сравнение элементов массива структур.