МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

|  |
| --- |
| КАФЕДРА компьютерных технологий и программной инженерии |

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «Классы, конструкторы, деструкторы» |
| по дисциплине: Объектно-ориентированное программирование |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | Z9431 |  |  |  | Андреев Д.И. |
|  | номер группы |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | 2019/3781 | |  |  |  |

Санкт-Петербург 2021

Оглавление

[1. Условие 3](#_Toc70882716)

[2. Листинг программы 3](#_Toc70882717)

[3. Результаты работы программы. 4](#_Toc70882718)

[4. Вывод 5](#_Toc70882719)

1. Условие

Вариант 1. Создать класс, который хранил бы информацию о книге: название, автор, дата издания. Создать несколько объектов. Если в конструктор передан символьный параметр, например, «а», то контролировать, начинается ли имя автора с данной буквы. Если нет, то выводить сообщение об ошибке. Если параметр не задан, то контролировать не надо.

1. Листинг программы

Файл Book.h

#pragma once

#include <string>

class Book

{

public:

Book(char control = '\0');

Book(std::string name, std::string athor, std::string publishDate, char control = '\0');

Book(const Book& other);

~Book();

const std::string &name() const;

const std::string &author() const;

const std::string &publishDate() const;

void setName(const std::string &name);

void setAuthor(const std::string &author);

void setPublishDate(const std::string &date);

private:

std::string m\_name;

std::string m\_author;

std::string m\_publishDate;

char m\_control;

};

Файл Book.cpp

#include "Book.h"

#include <iostream>

Book::Book(char control)

: m\_name()

, m\_author()

, m\_publishDate()

, m\_control(control)

{

}

Book::Book(std::string name, std::string athor, std::string publishDate, char control)

: m\_name(name)

, m\_author(athor)

, m\_publishDate(publishDate)

, m\_control(control)

{

}

Book::Book(const Book& other)

: m\_name(other.m\_name)

, m\_author(other.m\_author)

, m\_publishDate(other.m\_publishDate)

, m\_control(other.m\_control)

{

}

Book::~Book()

{

}

const std::string& Book::name() const

{

return m\_name;

}

const std::string& Book::author() const

{

return m\_author;

}

const std::string& Book::publishDate() const

{

return m\_publishDate;

}

void Book::setName(const std::string& name)

{

if (m\_name == name)

return;

m\_name = name;

}

void Book::setAuthor(const std::string& author)

{

if (m\_author == author)

return;

if (m\_control != '\0' && \*author.begin() != m\_control)

std::cerr << "author name must start with " << m\_control << '\n';

m\_author = author;

}

void Book::setPublishDate(const std::string& date)

{

if (m\_publishDate == date)

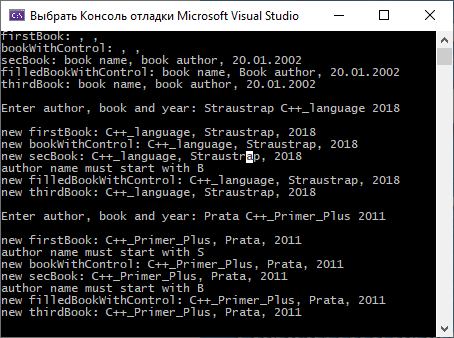
return;

m\_publishDate = date;

}

1. Результаты работы программы.

Результат работы программы:



Код для тестирования работы класса:

#include "Book.h"

#include <iostream>

#include <string>

std::ostream& operator<<(std::ostream& os, const Book& book)

{

return os << book.name() << ", " << book.author() << ", " << book.publishDate() << '\n';

}

int main()

{

Book firstBook;

std::cout << "firstBook: " << firstBook;

Book bookWithControl('S');

std::cout << "bookWithControl: " << bookWithControl;

Book secBook("book name", "book author", "20.01.2002");

std::cout << "secBook: " << secBook;

Book filledBookWithControl("book name", "Book author", "20.01.2002", 'B');

std::cout << "filledBookWithControl: " << filledBookWithControl;

Book thirdBook(secBook);

std::cout << "thirdBook: " << thirdBook << '\n';

std::string book, author, year;

std::cout << "Enter author, book and year: ";

std::cin >> author >> book >> year;

std::cout << '\n';

firstBook.setAuthor(author);

firstBook.setName(book);

firstBook.setPublishDate(year);

std::cout << "new firstBook: " << firstBook;

bookWithControl.setAuthor(author);

bookWithControl.setName(book);

bookWithControl.setPublishDate(year);

std::cout << "new bookWithControl: " << bookWithControl;

secBook.setAuthor(author);

secBook.setName(book);

secBook.setPublishDate(year);

std::cout << "new secBook: " << secBook;

filledBookWithControl.setAuthor(author);

filledBookWithControl.setName(book);

filledBookWithControl.setPublishDate(year);

std::cout << "new filledBookWithControl: " << filledBookWithControl;

thirdBook.setAuthor(author);

thirdBook.setName(book);

thirdBook.setPublishDate(year);

std::cout << "new thirdBook: " << thirdBook;

std::cout << '\n'; std::cout << "Enter author, book and year: ";

std::cin >> author >> book >> year;

std::cout << '\n';

firstBook.setAuthor(author);

firstBook.setName(book);

firstBook.setPublishDate(year);

std::cout << "new firstBook: " << firstBook;

bookWithControl.setAuthor(author);

bookWithControl.setName(book);

bookWithControl.setPublishDate(year);

std::cout << "new bookWithControl: " << bookWithControl;

secBook.setAuthor(author);

secBook.setName(book);

secBook.setPublishDate(year);

std::cout << "new secBook: " << secBook;

filledBookWithControl.setAuthor(author);

filledBookWithControl.setName(book);

filledBookWithControl.setPublishDate(year);

std::cout << "new filledBookWithControl: " << filledBookWithControl;

thirdBook.setAuthor(author);

thirdBook.setName(book);

thirdBook.setPublishDate(year);

std::cout << "new thirdBook: " << thirdBook;

std::cout << '\n';

return 0;

}

Для тестирования работы класса, как можно увидеть в коде выше, создаются 5 экземпляров класса. При этом вызываются разные конструкторы: для первых двух – конструктор по умолчанию, для вторых двух – конструктор инициализации, для последнего конструктор копирования. Второму и четвертому экземпляру в конструктор передается параметр для контролирования имени автора. Как можно на скриншоте работы программы, в случае, когда имя автора не начинается с символа, переданного в конструктор (если он задан) на экран выводится сообщение об ошибке.

1. Вывод

В ходе лабораторной работы был разработан класс, инкапсулирующий информацию о книге. Класс имеет конструктор по умолчанию для создания пустого экземпляра, конструктор копирования, для выполнения глубокого копирования, конструктор инициализации, для создания уже готового экземпляра. Все поля класса являются закрытыми. Доступ к ним осуществляется через соответствующие методы.