Министерство образования Республики Беларусь

Учреждение образования

«Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №3**

За 4 семестр

По дисциплине: “ООТПиСП”

Тема: “классы и объекты в С++”

**Вариант №9**

**Выполнил**:

студент 2 курса

группы ПО-7 (перв.)

Крупенков М.Д.

**Проверил:**

Хацкевич М.В.

Брест 2022

**Цель.** Получить практические навыки реализации классов на С++.

**Основное содержание работы.** Написать программу, в которой создаются и разрушаются объекты, определенного пользователем класса. Выполнить исследование вызовов конструкторов и деструкторов.

**Порядок выполнения работы:**

1. Определить пользовательский класс в соответствии с вариантом задания (смотри приложение)
2. Определить в классе следующие конструкторы: без параметров, с параметрами, копирования
3. Определить в классе деструктор
4. Определить в классе компоненты-функции для просмотра и установки полей данных
5. Определить указатель на компоненту-функцию
6. Определить указатель на экземпляр класса
7. Написать демонстрационную программу, в которой создаются и разрушаются объекты пользовательского класса и каждый вызов конструктора и деструктора сопровождается выдачей соответствующего сообщения (какой объект какой конструктор или деструктор вызвал)
8. Показать в программе использование указателя на объект и указателя на компоненту-функцию.

**Вариант задания:**

|  |  |  |  |
| --- | --- | --- | --- |
| Вариант | Название | Поле | Тип |
| 8 | Товар | имя | char\* |
| количество | int |
| стоимость | float |

Код программы

**main.cpp**

#include **<cstdio>**#include **<cstring>**#include **"Product.h"  
  
int** main() {  
 **auto** empty = **new** Product;  
 **delete** empty;  
  
 **auto** beerPtr = **new** Product(**"Ideal"**, 35, 1.29 */\* from groshik \*/* );  
 Product beerCopy(\*beerPtr);  
 **delete** beerPtr;  
 beerCopy.setName(strcat(beerCopy.getName(), **" NEW!!!"**));  
 beerCopy.setPrice(beerCopy.getPrice() \* 1.45f);  
  
 Product chips(**"Lay's: Chili & lime"**, 10, 2.39);  
 chips.setCount(15);  
  
 **float** starterPack = chips.getPrice() + beerCopy.getPrice();  
 printf(**"The minimum price = %.2f\n"**, starterPack);  
  
 **void** (Product::\*printPtr)() **const**;  
 printPtr = &Product::print;  
 (beerCopy.\*printPtr)();  
}

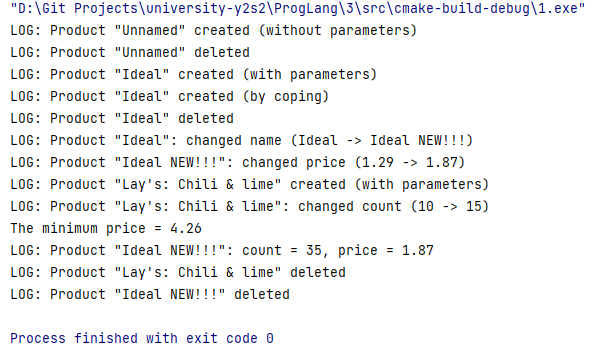
**Product.h**

#ifndef **PRODUCT\_H**#define **PRODUCT\_H  
  
  
class** Product {  
**public**:  
 Product();  
 Product(**const char** \*name, **int** count, **float** price);  
 Product(**const** Product &product);  
 ~Product();  
  
 **void** print() **const**;  
  
 **char**\* getName() **const**;  
 **int** getCount() **const**;  
 **float** getPrice() **const**;  
  
 **void** setName(**const char**\* name);  
 **void** setCount(**int** count);  
 **void** setPrice(**float** price);  
**protected**:  
 **char** \*name\_;  
 **int** count\_;  
 **float** price\_;  
};  
  
  
#endif *//PRODUCT\_H*

**Product.cpp**

#include **<cstring>**#include **<cstdio>**#include **"Product.h"**Product::Product() : count\_(0), price\_(0) {  
 name\_ = **new char**[8];  
 strcpy(name\_, **"Unnamed"**);  
 printf(**"LOG: Product \"%s\" created (without parameters)\n"**, name\_);  
}  
  
Product::Product(**const char** \*name, **int** count, **float** price) : count\_(count), price\_(price) {  
 name\_ = **new char**[strlen(name)];  
 strcpy(name\_, name);  
 printf(**"LOG: Product \"%s\" created (with parameters)\n"**, name\_);  
}  
  
Product::Product(**const** Product &product) : count\_(product.count\_), price\_(product.price\_) {  
 name\_ = **new char**[strlen(product.name\_)];  
 strcpy(name\_, product.name\_);  
 printf(**"LOG: Product \"%s\" created (by coping)\n"**, name\_);  
}  
  
Product::~Product() {  
 printf(**"LOG: Product \"%s\" deleted\n"**, name\_);  
 **delete**[] name\_;  
}  
  
**void** Product::print() **const** {  
 printf(**"LOG: Product \"%s\": count = %d, price = %.2f\n"**, name\_, count\_, price\_);  
}  
  
**char** \*Product::getName() **const** {  
 **auto** name = **new char**[strlen(name\_)];  
 strcpy(name, name\_);  
 **return** name;  
}  
  
**int** Product::getCount() **const** {  
 **return** count\_;  
}  
  
**float** Product::getPrice() **const** {  
 **return** price\_;  
}  
  
**void** Product::setName(**const char** \*name) {  
 printf(**"LOG: Product \"%s\": changed name (%s -> %s)\n"**, name\_, name\_, name);  
 **delete**[] name\_;  
 name\_ = **new char**[strlen(name)];  
 strcpy(name\_, name);  
}  
  
**void** Product::setCount(**int** count) {  
 printf(**"LOG: Product \"%s\": changed count (%d -> %d)\n"**, name\_, count\_, count);  
 count\_ = count;  
}  
  
**void** Product::setPrice(**float** price) {  
 printf(**"LOG: Product \"%s\": changed price (%.2f -> %.2f)\n"**, name\_, price\_, price);  
 price\_ = price;  
}

Результаты программы



**Вывод:** Было получено преисполнение при работе с классами в С++.