

Министерство образования Республики Беларусь  
Учреждение образования  
«Брестский государственный технический университет»  
Кафедра ИИТ

Лабораторная работа №3  
по дисциплине:  
Тема: Классы и объекты в C++

**Выполнил**  
студент 2 курса  
Корнасевич И. Д.

**Проверил**  
Миндер А. В.

## Задание:

1. Определить пользовательский класс в соответствии с вариантом задания.
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.

### main.cpp

---

```
1  #include <iostream>
2  #include "product.h"
3
4  int main() {
5      auto a = new Product("Apple", 1000, 2000);
6      Product b;
7      Product c("Balls", 2, 1);
8
9      // Указатель на функцию.
10     auto pgetName = &Product::getName;
11     auto name = (*a.*pgetName)();
12
13     a->setName("Milk")->setCount(10000)->setPrice(450);
14
15     delete a;
16 }
```

---

### product.h

---

```
1  //
2  // Created by ivan on 12/3/21.
3  //
4
5  #ifndef LAB3_PRODUCT_H
6  #define LAB3_PRODUCT_H
7
8
9  class Product {
```

```

10 public:
11     Product(const char *name, int count, long price);
12
13     Product(const Product &other);
14
15     Product();
16
17     ~Product();
18
19     const char *getName() const;
20
21     Product *setName(const char *newName);
22
23     Product *setName(char *newName);
24
25     int getCount() const;
26
27     Product *setCount(int newCount);
28
29     long getPrice() const;
30
31     Product *setPrice(long newPrice);
32
33 private:
34     char *name;
35     int count;
36     long price;
37 };
38
39
40 #endif //LAB3_PRODUCT_H

```

---

### product.cpp

---

```

1 #include <iostream>
2 #include <cstring>
3 #include "product.h"
4
5 Product::Product(const char *name, int count, long price) {
6     auto len = strlen(name) + 1;
7     this->name = new char[len];
8     this->name = strcpy(this->name, name);
9     this->count = count;
10    this->price = price;
11    std::cout << name << " created." << std::endl;
12 }
13
14 Product::Product(const Product &other) : Product::Product(other
    .name, other.count, other.price) {
15 }
16
17 Product::Product() : Product::Product("< blank >", -1, -1) {

```

```

18 }
19
20 Product::~~Product() {
21     std::cout << name << " deleted." << std::endl;
22     delete name;
23 }
24
25 const char *Product::getName() const {
26     return name;
27 }
28
29 Product *Product::setName(const char *newName) {
30     delete[] name;
31     auto len = strlen(newName) + 1;
32     name = new char[len];
33     strcpy(name, newName);
34     return this;
35 }
36
37 Product *Product::setName(char *newName) {
38     delete[] name;
39     name = newName;
40     return this;
41 }
42
43 int Product::getCount() const {
44     return count;
45 }
46
47 Product *Product::setCount(int newCount) {
48     count = newCount;
49     return this;
50 }
51
52 long Product::getPrice() const {
53     return price;
54 }
55
56 Product *Product::setPrice(long newPrice) {
57     price = newPrice;
58     return this;
59 }

```

---

**Вывод:** Я написал класс `Product`, который имеет 3 поля; конструктор по умолчанию, конструктор копирования, конструктор с параметрами; деструктор; геттер и сеттер к каждому полю. Также создал указатель на метод класса и указатель на объект класса.