Министерство образования Республики Беларусь

УО «Брестский государственный технический университет»

Кафедра ИИТ

**Лабораторная работа №2**

По дисциплине: “Языки программирования”

Тема: “Наследование и виртуальные функции”**Вариант №9**

**Выполнил**: студент 2 курса группы ПО-7 Крупенков Михаил Дмитриевич

**Проверила:** Дряпко А. В.

Брест 2021

**Цель:**

Получить практические навыки создания иерархии классов и использования статических компонентов класса.

**Постановка задачи:**

Написать программу, в которой создается иерархия классов. Включить полиморфные объекты в связанный список, используя статические компоненты класса. Показать использование виртуальных функций.

Порядок выполнения работы:

1. Определить иерархию классов (в соответствии с вариантом).
2. Определить в классе статическую компоненту - указатель на начало связанного списка объектов и статическую функцию для просмотра списка.
3. Реализовать классы.
4. Написать демонстрационную программу, в которой создаются
5. объекты различных классов и помещаются в список, после чего список просматривается.
6. Сделать соответствующие методы не виртуальными и посмотреть, что будет.
7. Реализовать вариант, когда объект добавляется в список при создании, т.е. в конструкторе (смотри пункт 6 следующего раздела).

**Вариант задания:**

Перечень классов:

9) игрушка, продукт, товар, молочный продукт

**Иерархия классов:**

**class** SaleItem  
 **const char** \*name;  
 **float** price;

**class** Product: SaleItem  
 **bool** sAlcoholic;

**class** Toy : SaleItem  
 **int** min\_age;

**class** MilkProduct : Product  
 **float** fatness;

**Код программы:**

**main.cpp**

#include **"classes.h"  
  
int** SaleItem::count = 0;  
SaleItem \*\*SaleItem::items = **nullptr**;  
  
**int** main() {  
 Toy pony(**"AppleJack"**, 7, 0);  
 pony.add();  
 *// Для остальных add используется по умолчанию в родительском классе* Product beer(**"beer"**, 1.59, **true**);  
 Product crisps(**"crisps"**, 3.47, **false**);  
 MilkProduct milk(**"milk"**, 1.59, **false**, 3.6);  
  
 SaleItem::showItems();  
}

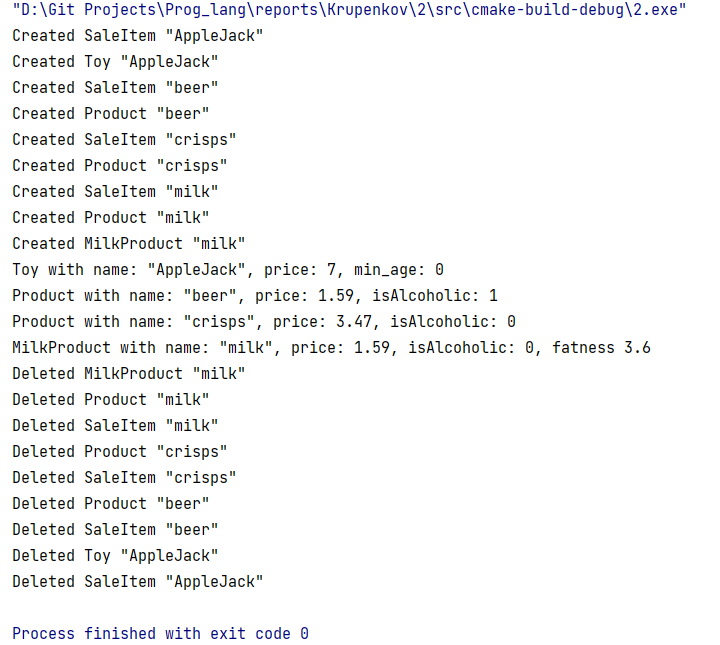
**classes.h**

#ifndef **INC\_2\_CLASSES\_H**#define **INC\_2\_CLASSES\_H  
  
  
class** SaleItem {  
**protected**:  
 **const char** \*name;  
 **float** price;  
  
**public**:  
 **static int** count;  
 **static** SaleItem \*\*items;  
  
 SaleItem(**const char** \*name, **float** price, **bool** isAdd=**true**);  
  
 ~SaleItem();  
  
 **virtual void** print() = 0;  
  
 **virtual void** add() **final**;  
  
 **static void** showItems();  
};  
  
  
**class** Toy : **public** SaleItem {  
**protected**:  
 **int** min\_age;  
  
**public**:  
 Toy(**const char** \*name, **float** price, **int** min\_age);  
  
 ~Toy();  
  
 **void** print() **override**;  
};  
  
  
**class** Product : **protected** SaleItem {  
**protected**:  
 **bool** isAlcoholic;  
  
**public**:  
 Product(**const char** \*name, **float** price, **int** isAlcoholic);  
  
 ~Product();  
  
 **void** print() **override**;  
};  
  
  
**class** MilkProduct : Product {  
**protected**:  
 **float** fatness;  
  
**public**:  
 MilkProduct(**const char** \*name, **float** price, **int** isAlcoholic, **float** fatness);  
  
 ~MilkProduct();  
  
 **void** print() **override**;  
};  
  
  
#endif *//INC\_2\_CLASSES\_H*

**classes.cpp**

#include **<iostream>**#include **"classes.h"  
using namespace** std;  
  
  
SaleItem::SaleItem(**const char** \*name, **float** price, **bool** isAdd) :  
name(name), price(price) {  
 **if** (isAdd) add();  
 cout << **"Created SaleItem \""** << name << **"\"\n"**;  
}  
  
SaleItem::~SaleItem() {  
 cout << **"Deleted SaleItem \""** << name << **"\"\n"**;  
}  
  
**void** SaleItem::add() {  
 SaleItem\*\* temp = items;  
 items = **new** SaleItem\* [count + 1];  
 **for**(**int** i = 0; i < count; i++)  
 items[i] = temp[i];  
 items[count] = **this**;  
 count++;  
}  
  
**void** SaleItem::showItems() {  
 **for**(**int** i=0; i<count; i++)  
 items[i]->print();  
}  
  
  
Toy::Toy(**const char** \*name, **float** price, **int** min\_age) :  
SaleItem(name, price, **false**), min\_age(min\_age) {  
 cout << **"Created Toy \""** << name << **"\"\n"**;  
}  
  
Toy::~Toy() {  
 cout << **"Deleted Toy \""** << name << **"\"\n"**;  
}  
  
**void** Toy::print() {  
 cout << **"Toy with name: \""** << name << **"\", price: "** << price << **", min\_age: "** << min\_age << **"\n"**;  
}  
  
  
Product::Product(**const char** \*name, **float** price, **int** isAlcoholic) :  
SaleItem(name, price, **true**), isAlcoholic(isAlcoholic) {  
 cout << **"Created Product \""** << name << **"\"\n"**;  
}  
  
Product::~Product() {  
 cout << **"Deleted Product \""** << name << **"\"\n"**;  
}  
  
**void** Product::print() {  
 cout << **"Product with name: \""** << name << **"\", price: "** << price << **", isAlcoholic: "** << isAlcoholic << **"\n"**;  
}  
  
  
MilkProduct::MilkProduct(**const char** \*name, **float** price, **int** isAlcoholic, **float** fatness) :  
Product(name, price, isAlcoholic), fatness(fatness) {  
 cout << **"Created MilkProduct \""** << name << **"\"\n"**;  
}  
  
MilkProduct::~MilkProduct() {  
 cout << **"Deleted MilkProduct \""** << name << **"\"\n"**;  
}  
  
**void** MilkProduct::print() {  
 cout << **"MilkProduct with name: \""** << name << **"\", price: "** << price  
 << **", isAlcoholic: "** << isAlcoholic << **", fatness "** << fatness << **"\n"**;  
}

**Результат программы:**

****

**Вывод:**

Я получил практические навыки создания иерархии классов и использования статических компонентов класса.