|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ | | | | | |
| **ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ** | | | | | |
| **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** | | | | | |
| **«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»** | | | | | |
|  | | | | | |
| Институт информационных технологий и управления в технических системах | | | | | |
| (полное название института) | | | | | |
|  | | | | | |
| кафедра «Информационные системы» | | | | | |
| (полное название кафедры) | | | | | |
|  | | | | | |
|  | | | | | |
| **Лабораторные работы** | | | | | |
| по дисциплине “Объектно-ориентированное программирование” | | | | | |
| студента группы ИС/б-22-о  **Волобуева Юрия Сергеевича** | | | | | |
| № лр. | Выполнение | Работу принял | | | |
| Дата | Дата | Оценка | Ф.И.О. | Подпись |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| 20 | 18 | г. |

1 ЦЕЛЬ РАБОТЫ

Исследование основных средств определения класса, создания объектов класса, приобретение навыков разработки и отладки программ, использующих динамическую память. Исследование особенностей использования конструкторов копирования.

2 ПОСТАНОВКА ЗАДАЧИ

Необходимо для заданного по варианту динамического типа данных описать класс, содержащий указатель на динамический тип как поле данных. Для этого класса описать конструкторы (не менее трех, в том числе и конструктор копирования), деструктор, функцию печати данных. Создать экземпляр полученного класса и проиллюстрировать его корректную работу: распечатать данные, изменить данные и распечатать вновь. Создать второй экземпляр класса как копию первого и проиллюстрировать корректную работу конструктора копирования: распечатать и изменить данные объекта-копии, распечатать данные обоих объектов, сравнить результат. Предусмотреть обработку ошибок при манипуляции с данными.

Вариант 5

Динамическая структура — очередь. В списке хранится информация о чеках: наименование товара (строка), цена товара и дата продажи. Предусмотреть функции добавления элементов в список и удаления из него, а также поиска товаров, цена которых превышает значение, вводимое пользователем.

3 ТЕКСТ ПРОГРАММЫ

#include <fstream>

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

#include <time.h>

#include <iomanip>

#include <conio.h>

#include <iostream>

using namespace std;

struct elem

{

char name[30];

float price;

char date[10];

elem\* next;

} ;

class product

{

private:

elem \* head, \*tail;

public:

product() : head(NULL), tail(NULL) {};

product(product &cop)

{

elem \*temp, \*el;

if (!cop.head) return;

head = new elem;

strcpy(head->name, cop.head->name);

head->price = cop.head->price;

strcpy(head->date, cop.head->date);

head->next = NULL;

tail = head;

temp = cop.head->next;

while (temp)

{

el = new elem;

tail->next = el;

tail=el;

tail->next = 0;

strcpy(tail->name, temp->name);

tail->price = temp->price;

strcpy(tail->date, temp->date);

temp = temp->next;

}

}

~product()

{

elem \*temp = head;

while (head)

{

temp = head;

head = head->next;

delete temp;

}

}

void input(elem \*temp);

void addelem();

void output();

void change();

int create();

void del();

void searching();

};

void product :: input(elem \*temp)

{

cout << "Название товара:" << endl;

cin >> temp->name;

cout << "Стоимость товара:" << endl;

cin >> temp->price;

cout << "Дата продажи:" << endl;

cin >> temp->date;

}

void product :: change()

{

int i = 1, n;

elem \*tmp;

cout<<"Введите номер заменяемого поля:"<< endl;

cin >> n;

tmp = head;

while (i!= n)

{

i++;

tmp = tmp->next;

}

input(tmp);

}

int product :: create()

{

char key;

elem \*newE;

newE = new elem;

input(newE);

newE->next = head;

head = newE;

tail = newE;

do

{

cout << "Добавить еще?" << endl;

cout << "1 - Да" << endl;

cout << "2 - Нет" << endl;

cin >> key;

switch (key)

{

case '1': addelem(); break;

case '2': return 0;

}

}

while (1);

}

void product :: addelem()

{

elem \*newE = new elem;

input(newE);

newE->next=0;

tail->next=newE;

tail=newE;;

}

void product::output()

{

elem \*temp;

temp = head;

if (head)

{

while (temp)

{

cout << " " << temp->name<< " " << temp->price << endl;

temp = temp->next;

}

}

}

void product :: del()

{

elem \*temp = 0;

if (head)

{

temp = head;

head = head->next;

delete temp;

}

else cout << "Список пуст" << endl;

}

void product :: searching()

{

elem\* temp=head;

int t, i;

cout << "Введите минимальную необходимую цену:" << endl;

cin >> t;

while(temp)

{

if(temp->price>=t)

{

cout << temp->name << " " << temp->price;

i++;

}

temp=temp->next;

}

if (i==0) cout << "Не найдено" << endl;

}

int main()

{

setlocale(0, "rus");

char key1, key2;

product p1;

cout << "Для создания списка введите данные." << endl;

p1.create();

product p2(p1);

system("cls");

while (1)

{

cout << "1 - Показать список" << endl;

cout << "2 - Удалить товар" << endl;

cout << "3 - Добавить товар" << endl;

cout << "4 - Сменить товар" << endl;

cout << "5 - Поиск товара" << endl;

cout << "6 - Выход" << endl;

cin >> key2;

switch (key2)

{

case '1':

{

cout << "-------------------------------" << endl;

cout << "| Имя "<<" Номер |" << endl;

cout << "|-----------------------------|" << endl;

cout << "| Оригинал: |" << endl;

cout << "|-----------------------------|" << endl;

p1.output();

cout << "|-----------------------------|" << endl;

cout << "| Копия: |" << endl;

cout << "|-----------------------------|" << endl;

p2.output();

cout << "------------------------------" << endl;

system("pause");

system("cls");

}

break;

case '2':

{

cout << "Где удалить элемент?" << endl;

cout << "1 - Оригинал" << endl;

cout << "2 - Копия" << endl;

cin >> key1;

switch (key1)

{

case'1':p1.del(); break;

case'2':p2.del(); break;

}

system("pause");

system("cls");

}

break;

case '3':

{

cout << "Куда добавить элемент?" << endl;

cout << "1 - Оригинал" << endl;

cout << "2 - Копия" << endl;

cin >> key1;

switch (key1)

{

case'1':p1.addelem(); break;

case'2':p2.addelem(); break;

}

system("pause");

system("cls");

}

break;

case'4':

{

cout << "Где изменить элемент?" << endl;

cout << "1 - Оригинал" << endl;

cout << "2 - Копия" << endl;

cin >> key1;

switch (key1)

{

case'1':p1.change(); break;

case'2':p2.change(); break;

}

system("pause");

system("cls");

}

break;

case '5':

{

p1.searching();

system("pause");

system("cls");

break;

}

case '6': return 0;

default: cout << "Введите число от 1 до 6:" << endl;

}

}

}

4 ТЕСТИРОВАНИЕ

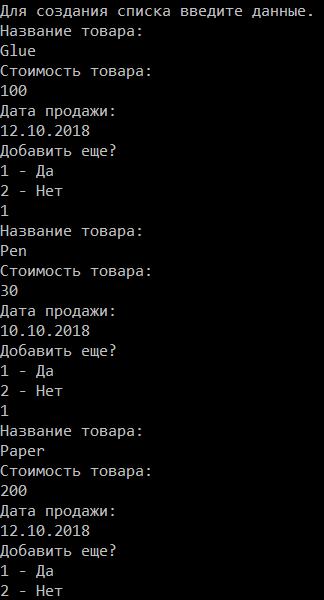


Рисунок 1 – Создание списка

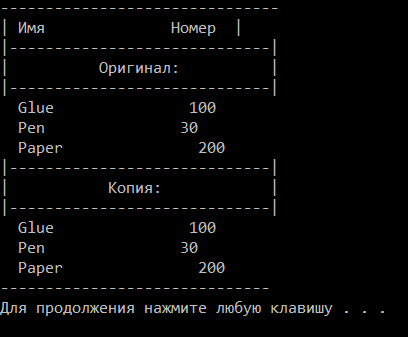


Рисунок 2 – Просмотр списка

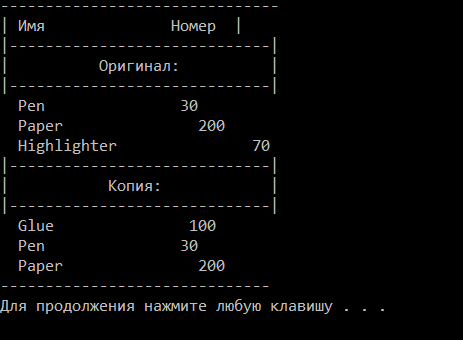


Рисунок 3 – Функция поиска по цене

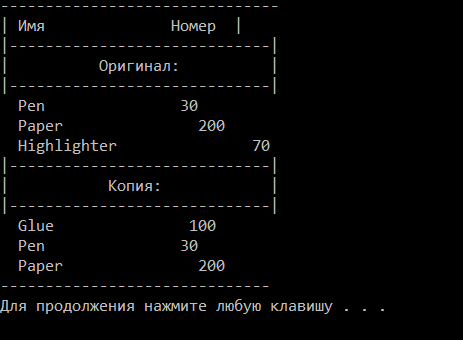


Рисунок 4 – Результат удаления элемента

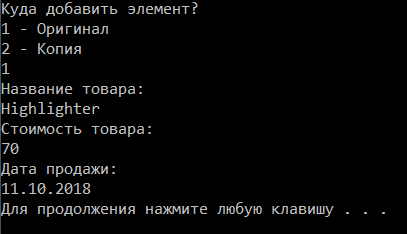


Рисунок 7 – Функция добавления элемента

ВЫВОД

В ходе выполнения лабораторной работы были получены навыки работы с классами. Изучены основные средства определения класса, создания объектов класса, особенности использования конструкторов копирования. Была написана программа, описывающая класс. Для этого класса были описаны конструкторы, деструкторы и функции добавления элементов, удаления и поиска товаров дороже введенной цены.