|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 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  **Долженко Ильи Андреевича** | | | | | |
| № 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 — двусвязный список. Хранит информацию о товарах на складе: инвентарный номер (число 5 знаков), название товара (строка). Предусмотреть функции добавления элементов в список и удаления из него, а также функцию проверки, существует ли товар на складе, по инвентарному номеру.

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

{

int number;

string name;

elem \*next, \*prev;

};

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;

head->number = cop.head->number;

head->name = cop.head->name;

head->next = NULL;

tail = head;

temp = cop.head->next;

while (temp)

{

el = new elem;

tail->next = el;

tail=el;

tail->next = 0;

tail->number = temp->number;

tail->name = temp->name;

temp = temp->next;

}

}

~product()

{

elem \*temp = head;

while (head)

{

temp = head;

head = head->next;

delete temp;

}

}

void vvod(elem \*temp);

void add\_el();

void vivod();

void change();

int sozdanie();

void del();

void poisk();

};

void product::vvod(elem \*temp)

{

cout << "Номер товара:" << endl;

cin >> temp->number;

cout << "Название товара:" << endl;

cin >> temp->name;

}

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;

}

vvod(tmp);

}

int product :: sozdanie()

{

char key;

elem \*newE;

newE = new elem;

vvod(newE);

newE->next = head;

head = newE;

tail = newE;

do

{

cout << "Добавить еще данные?" << endl;

cout << "1 - Да" << endl;

cout << "2 - Нет" << endl;

cin >> key;

switch (key)

{

case '1': add\_el(); break;

case '2': return 0;

}

}

while (1);

}

void product :: add\_el()

{

elem \*newE = new elem;

vvod(newE);

if (!head)

{

newE -> next = tail;

tail = newE;

}

else

{

newE -> next = head;

head -> prev = newE;

}

head = newE;

head -> prev = NULL;

}

void product :: vivod()

{

elem \*temp;

temp = head;

if (head)

{

while (temp)

{

cout << " " << temp->number<< " " << temp->name << 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 :: poisk() {

elem\* temp2=head;

int index, i;

cout << "Введите номер для поиска:" << endl;

cin >> index;

while(temp2)

{

if (temp2->number==index) i++;

temp2=temp2->next;

}

if (i!=0) cout << "Найдено" << endl;

else cout << "Не найдено" << endl;

}

int main()

{

setlocale(0, "rus");

char key1, key2;

product p1;

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

p1.sozdanie();

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 << "| Номер "<<" Название |" << endl;

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

cout << "| Оригинал: |" << endl;

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

p1.vivod();

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

cout << "| Копия: |" << endl;

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

p2.vivod();

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.add\_el(); break;

case'2':p2.add\_el(); 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.poisk();

system("pause");

system("cls");

break;

}

case '6': return 0;

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

}

}

}

4 ТЕСТИРОВАНИЕ

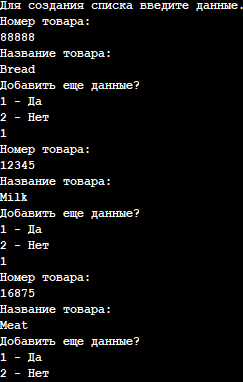


Рисунок 1 – Создание списка

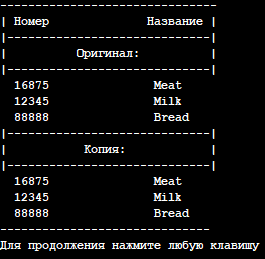


Рисунок 2 – Просмотр созданного списка

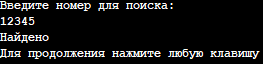


Рисунок 3 – Функция поиска по списку (элемент найден)

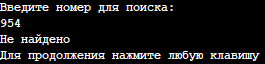


Рисунок 4 – Функция поиска по списку (элемент не найден)

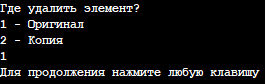


Рисунок 5 – Функция удаления элемента

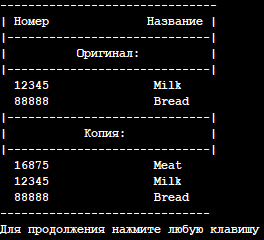


Рисунок 6 – Результат удаления элемента

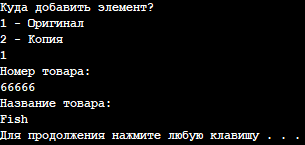


Рисунок 7 – Функция добавления элемента

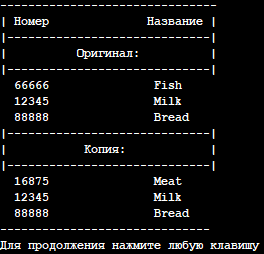


Рисунок 8 – Результат добавления элемента

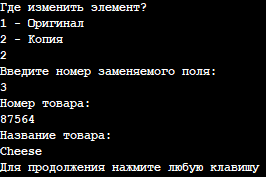


Рисунок 9 – Функция замены элемента

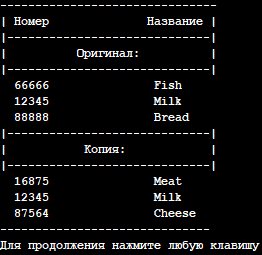


Рисунок 10 – Результат замены элемента

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, содержащий указатель на динамический тип как поле данных. Для этого класса были описаны конструкторы, деструкторы и функции добавления элементов, удаления и проверки на наличие товара на складе. Был создан экземпляр полученного класса и его копия, над которыми были использованы функции печати и изменения данных, и последующего их сравнения.