Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра Программной Инженерии

Лабораторная работа 8

По дисциплине «Основы алгоритмизации и программирования»

На тему «Полустатические структуры данных: очереди»

Выполнил:

Студент 1 курса 3 группы

Шатерник Г.И.

Преподаватель: асс. Андронова М.В.

2023, Минск

Вариант 15

1. Создать проект из нескольких файлов, демонстрирующий работу с очередью. В соответствии со своим вариантом выполнить задание из таблицы, представленной ниже. Разработать меню и реализовать все операции с очередью через функции. Максимальный размер очереди ввести с клавиатуры.
2. Создать очередь для символов. Создать функции для ввода, вывода, удаления и определения размера очереди. Ввести эталонный символ. Вводить символы с экрана в очередь до встречи эталонного. При встрече эталонного, удалить два элемента очереди. Вывести размер очереди.

|  |
| --- |
| Программный код |
| #include<iostream>  using namespace std;  struct Number // создаем структуру типа number  {  char info;  Number\* next;  };  int choice, siz, k = 0;  char etal, element; //задаем тип данных для эталонного элемента  void EnNumber(Number\*\* pnum, char element); // обьявляем функции и их параметры  void PrnN(Number\*\* pnum);  void Clear(Number\*\* pnum);  void deltwo(Number\*\* pnum);  int main()  {  Number\* num = new Number();  setlocale(0, "");  cout << "Введите максимальный размер очереди: ";  cin >> siz;  cout << "Введите эталонный символ";  cin >> etal;  cout << "Выберите команду:" << endl;  cout << "1 - Ввод эллемента" << endl;  cout << "2 - Вывод очереди" << endl;  cout << "3 - Очистка очереди" << endl;  cout << "4 - Размер очереди" << endl;  cout << "5 - Выход" << endl;  cin >> choice;  do {  switch (choice)  {  case 1: if (siz != k)  {  cout << "Введите элемент: ";  cin >> element;  if (element == etal)  {  cout << "Встречен эталонный символ. Будут удалены 2 символа из очереди" << endl;  deltwo(&num);  cout << "Что дальше? ";  cin >> choice;  break;  }  EnNumber(&num, element);  }  else  {  cout << "Максимальный размер очереди достигнут!" << endl;  cout << "---------------------" << endl;  cout << "Что дальше? ";  cin >> choice;  }  break;  case 2:  PrnN(&num);  break;  case 3:  Clear(&num);  break;  case 4: cout << "Размер очереди: " << k << endl;  cout << "---------------------" << endl;  cout << "Что дальше? ";  cin >> choice;  break;  }  } while (choice != 5);  system("pause");  }  //Функция добавления элемента в очередь:  void EnNumber(Number\*\* pnum, char element)  {  if (\*pnum) //если очередь не пустая  {  Number\* pN = new Number;  Number\* pQ;  pQ = (\*pnum);  while (pQ->next)  pQ = pQ->next; //перейти в конец  pN->info = element; //заполнить поле  pN->next = NULL;  pQ->next = pN; //добавить в конец  k++;  }  else  {  (\*pnum) = new Number;  (\*pnum)->info = element; //заполнить поле  (\*pnum)->next = NULL;  }  cout << "---------------------" << endl;  cout << "Что дальше? ";  cin >> choice;  }  // Функция вывода на экран:  void PrnN(Number\*\* pnum)  {  if ((\*pnum) != NULL)  {  Number\* t = \*pnum;  while (t->next)  {  t = t->next;  cout << t->info << " ";  }  cout << endl;  }  else {  cout << "В очереди нет эллементов!" << endl;  }  cout << "---------------------" << endl;  cout << "Что дальше? ";  cin >> choice;  }  void Clear(Number\*\* pnum)  {  Number\* pDel = \*pnum;  while ((\*pnum) != NULL)  {  pDel = \*pnum;  \*pnum = (\*pnum)->next; //перейти к след.  delete pDel; //освободить элемент  }  cout << "Очередь очищена!" << endl;  cout << "---------------------" << endl;  cout << "Что дальше? ";  cin >> choice;  }  void deltwo(Number\*\* pnum)  {  int i = 0;  Number\* t;  t = new Number;  while (i < 2)  {  t = \*pnum;  \*pnum = (\*pnum)->next;  delete t;  i++;  }  } |
| Вывод: |