**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра АПУ**

**ОТЧЕТ**

**по лабораторной работе №1**

**по дисциплине «Программирование»**

| Студент гр. 3391 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Николаев В.Ю. |
| --- | --- | --- |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Миненко М.В. |

Санкт-Петербург

2024 г.

# Исходная формулировка

Требуется разработать программу обеспечивающую:

* построение линейного односвязного списка типа «стек» с заполнением его узлов вводимыми с клавиатуры числами;
* вывод на экран в обратном порядке всех положительных чисел, хранимых в стеке.

# Ход работы программы

В начале программы создаётся стек чисел типа int:

| 6 | Stack<int> s; | . |
| --- | --- | --- |

Затем выводится сообщение о вводе чисел и числа считываются из консоли до тех пор, пока не будет введён 0:

| 8  9  10  11  12  13  14  15  16 | std::cout << "Enter the numbers you want to add to the stack. Enter 0 to stop." << std::endl;  int temp = 1;  while (temp != 0)  {  std::cin >> temp;  if (temp != 0)  s.push(temp);  } | . |
| --- | --- | --- |

После этого выводится сообщение о выводе положительных чисел в обратном порядке и сами числа в обратном порядке.

| 18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33 | std::cout << "Here are all the positive numbers in the stack, displayed in the order they were entered." << std::endl;  bool numberPrinted = false;  while (!s.empty())  {  temp = s.top();  if (temp > 0)  {  std::cout << temp << " ";  numberPrinted = true;  }  s.pop();  }  if (numberPrinted)  std::cout << std::endl; |
| --- | --- |

# Текст программы

| Stack.h | |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | #pragma once  template <typename T>  class Stack  {  private:  struct Node  {  T data;  Node\* next;  Node() : data(0), next(nullptr) {}  Node(T data) : data(data), next(nullptr) {}  ~Node()  {  delete next;  }  };  Node\* head;  public:  Stack() : head(nullptr) {}  void push(T data)  {  Node\* newNode = new Node(data);  newNode->next = head;  head = newNode;  }  void pop()  {  if (empty())  {  std::cout << "Stack is empty" << std::endl;  return;  }  Node\* temp = head;  head = head->next;  temp->next = nullptr;  delete temp;  }  T top()  {  if (empty())  {  std::cout << "Stack is empty" << std::endl;  return -1;  }  return head->data;  }  bool empty()  {  return head == nullptr;  }  ~Stack()  {  delete head;  }  }; |
| Main.cpp | |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36 | #include <iostream>  #include "Stack.h"  int main()  {  Stack<int> s;  std::cout << "Enter the numbers you want to add to the stack. Enter 0 to stop." << std::endl;  int temp = 1;  while (temp != 0)  {  std::cin >> temp;  if (temp != 0)  s.push(temp);  }  std::cout << "Here are all the positive numbers in the stack, displayed in the order they were entered." << std::endl;  bool numberPrinted = false;  while (!s.empty())  {  temp = s.top();  if (temp > 0)  {  std::cout << temp << " ";  numberPrinted = true;  }    s.pop();  }  if (numberPrinted)  std::cout << std::endl;  return 0;  } |

# Результаты работы программы

