**МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ**Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**МОСКОВСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**СВЯЗИ И ИНФОРМАТИКИ**

Кафедра «Информационная безопасность»

**ЛАБОРАТОРНАЯ РАБОТА**

**№6**

по дисциплине «Объектно-ориентированное программирование систем защиты информации»

на тему:

**«Одиночное и множественное наследование. Виртуальные и чисто виртуальные функции. Абстрактные и конкретные классы»**

Вариант №11

Выполнил студент группы БПЗ1802

Каплин П.А.

Проверил:

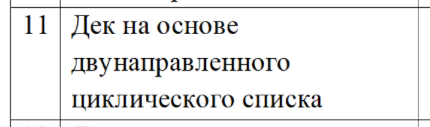
Полковников М. В.

Москва 2021 г

# 1. ЦЕЛЬ РАБОТЫ

Овладеть навыками создания базовых классов с виртуальными и чисто виртуальными функциями, а также производных классов с переопределением указанных функций.

# 2. ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ

  
Рисунок 1 – индивидуальное задание.

# 3. ПРОГРАММНЫЙ КОД

## Задание №1

Листинг 1 – Исходный код файла Deque.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  template<typename T>  class Base  {  public:  virtual void PushFront(T element) {}  virtual void PushBack(T element) {}  virtual T PopFront() { return 0; }  virtual T PopBack() { return 0; }  virtual int GetSize() { return 0; }  virtual T PeekFront() { return 0; }  virtual T PeekBack() { return 0; }  virtual void GetFirst() {}  };  template<typename T>  class DequeTemp: Base<T>  {  T\* arr;  int count;  int end = 0, size = 0;  public:  DequeTemp(int count)  : count(count)  {  this->count = count;  arr = new T[count];  }  DequeTemp(const DequeTemp& other)  {  arr = new T[other.count];  count = other.count;  end = other.end;  size = other.size;  for (int i = 0; i < count; ++i)  {  arr[i] = other.arr[i];  }  }  DequeTemp(DequeTemp&& other)  {  arr = other.arr;  count = other.count;  other.arr = nullptr;  }  ~DequeTemp()  {  delete[] arr;  }  DequeTemp& operator=(const DequeTemp& other)  {  if (this == &other)  {  return \*this;  }  delete[] arr;  arr = new T[other.count];  count = other.count;  for (int i = 0; i < count; ++i)  {  arr[i] = other.arr[i];  }  }  DequeTemp& operator=(DequeTemp&& other)  {  if (this == &other)  {  return \*this;  }  delete[] arr;  arr = other.arr;  count = other.count;  other.arr = nullptr;  }  int GetSize()  {  return this->size;  }  void PushFront(T element)  {  if (size <= count)  {  int\* tmp = new int[size];  for (int i = 0; i <= size; i++)  {  tmp[i] = arr[i];  }  for (int i = 0; i <= size; i++)  {  arr[i + 1] = tmp[i];  }  arr[0] = element;  size++;  end++;  }  }  void PushBack(T element)  {  if (size <= count)  {  end += 1;  arr[end] = element;  size++;  }  }  T PopFront()  {  if (size <= count)  {  for (int i = 0; i < end; i++)  {  arr[i] = arr[i + 1];  }  arr[end] = 0;  end--;  size--;  return 0;  }  }  T PopBack()  {  if (size <= count)  {  arr[end] = 0;  end--;  size--;  return 0;  }  }  T PeekFront()  {  return arr[0];  }  T PeekBack()  {  return arr[end];  }  void GetFirst(T element)  {  arr[0] = element;  size++;  }  friend ostream& operator<<(ostream& stream, DequeTemp& a)  {  for (int i = 0; i < a.size; i++)  stream << a.arr[i] << " ";  return stream;  }  }; |

Листинг 2 – Исходный код файла Deque\_Task1.cpp

|  |
| --- |
| #include <iostream>  #include "Deque.cpp"  using namespace std;  int main()  {  int arr, pos, element;  cout << "Deque Test Program" << endl << "Deque Length = ";  cin >> arr;  DequeTemp<int> a(arr);  cout << "Deque First Element = ";  cin >> element;  a.GetFirst(element);  cout << "Deque Fill Element" << endl;  while (a.GetSize() < arr)  {  cout << "[0] - Add element to Front Position" << endl << "[1] - Add element to Backward Position" << endl;  cin >> pos;  if (pos == 0)  {  cout << "Element " << a.GetSize() + 1 << " / " << arr << ": ";  cin >> element;  a.PushFront(element);  }  else if (pos == 1)  {  cout << "Element " << a.GetSize() + 1 << " / " << arr << ": ";  cin >> element;  a.PushBack(element);  }  }  cout << endl << "Deque: " << a << endl;  system("pause");  while (true)  {  cout << "Deque Main Menu" << endl << "[1] - Show Deque first element" << endl << "[2] - Show Deque last element" << endl << "[3] - Delete Front Deque Element"  << endl << "[4] - Delete Last Deque Element" << endl << "[5] - Exit" << endl << "Selected Item = ";  cin >> pos;  switch (pos)  {  case 1:  {  cout << a.PeekFront() << endl;  break;  }  case 2:  {  cout << a.PeekBack() << endl;  break;  }  case 3:  {  a.PopFront();  cout << endl << "Deque: " << a << endl;  break;  }  case 4:  {  a.PopBack();  cout << endl << "Deque: " << a << endl;  break;  }  case 5:  {  return 0;  }  default:  {  cout << "Error. Selected Item not exist.";  }  }  }  } |

Листинг 3 – Исходный код файла Deque2.cpp

|  |
| --- |
| #include <iostream>  using namespace std;  template<typename T>  class Base  {  public:  virtual void PushFront(T element) {}  virtual void PushBack(T element) {}  virtual void PopFront() { }  virtual void PopBack() { }  virtual int GetSize() { return 0; }  virtual T PeekFront() { return 0; }  virtual T PeekBack() { return 0; }  virtual void GetFirst() {}  };  template<typename T>  class DequeTemp : Base<T>  {  public:  DequeTemp()  {  head\_ = NULL;  tail\_ = NULL;  size\_ = 0;  }  DequeTemp(const DequeTemp& other)  {  head\_ = NULL;  tail\_ = NULL;  size\_ = 0;  Node<T>\* temp = new Node<T>();  while (temp->pNext\_ != other.tail\_)  {  PushBack(temp->data\_);  temp = temp->pNext\_;  }  PushBack(temp->data\_);  }  DequeTemp(DequeTemp&& other)  {  head\_ = NULL;  tail\_ = NULL;  size\_ = 0;  Node<T>\* temp = new Node<T>();  while (temp->pNext\_ != other.tail\_)  {  PushBack(other.deQueue());  temp = temp->pNext\_;  }  PushBack(temp->data\_);  }  ~DequeTemp()  {  while (size\_)  PopFront();  }  int GetSize()  {  return size\_;  }  void PushFront(T element)  {  if (head\_ == NULL)  {  head\_ = new Node<T>(element);  tail\_ = head\_;  }  else  {  head\_->pPrev\_ = new Node<T>(element, head\_, NULL);  head\_ = head\_ -> pPrev\_;  }  size\_++;  }  void PushBack(T element)  {  if (head\_ == NULL)  {  head\_ = new Node<T>(element);  tail\_ = head\_;  }  else  {  tail\_->pNext\_ = new Node<T>(element, NULL, tail\_);  tail\_ = tail\_->pNext\_;  }  size\_++;  }  void PopFront()  {  Node<T>\* temp = head\_;  head\_ = head\_->pNext\_;  head\_->pPrev\_ = NULL;  delete temp;  size\_--;  }  void PopBack()  {  Node<T>\* temp = tail\_;  tail\_ = tail\_->pPrev\_;  tail\_->pNext\_ = NULL;  delete temp;  size\_--;  }  T PeekFront()  {  return head\_->data\_;  }  T PeekBack()  {  return tail\_->data\_;  }  DequeTemp<T>& operator=(const DequeTemp<T>& other)  {  if (this == &other)  return \*this;  head\_ = NULL;  tail\_ = NULL;  size\_ = 0;  Node<T>\* temp = new Node<T>();  while (temp->pNext\_ != other.tail\_)  {  PushBack(temp->data\_);  temp = temp->pNext\_;  }  PushBack(temp->data\_);  }  DequeTemp<T>& operator=(DequeTemp<T>&& other)  {  if (this == &other)  return \*this;  head\_ = NULL;  tail\_ = NULL;  size\_ = 0;  Node<T>\* temp = new Node<T>();  while (temp->pNext\_ != other.tail\_)  {  PushBack(other.PopFront());  temp = temp->pNext\_;  }  PushBack(temp->data);  }  friend ostream& operator<<(ostream &out, DequeTemp &q)  {  Node<T>\* temp = q.head\_;  while (temp->pNext\_ != NULL)  {  int test = temp->data\_;  out << temp->data\_ << " ";  temp = temp->pNext\_;  }  out << temp->data\_ << " ";  return out;  }  private:  template<typename T>  class Node  {  public:  Node(T data, Node\* pNext = NULL, Node\* pPrev = NULL)  {  data\_ = data;  pNext\_ = pNext;  pPrev\_ = pPrev;  }  Node\* pNext\_;  Node\* pPrev\_;  T data\_;  };  Node<T>\* head\_;  Node<T>\* tail\_;  int size\_;  }; |

Листинг 4 – Исходный код файла main.cpp

|  |
| --- |
| #include <iostream>  #include "Deque2.cpp"  using namespace std;  int main()  {  setlocale(LC\_ALL, "rus");  DequeTemp<int>\* a = new DequeTemp<int>;  DequeTemp<int>\* aclass;  a->PushFront(1);  a->PushFront(2);  a->PushBack(3);  a->PushBack(4);  cout << \*a << endl;  cout << "First = " << a->PeekFront() << endl;  cout << "Back = " << a->PeekBack() << endl;  cout << "Size = " << a->GetSize() << endl;  a->PopFront();  cout << "PopFrontDeque: "<< \*a << endl;  a->PopBack();  cout << "PopBackDeque: " << \*a << endl;  aclass = a;  cout << "New Aclass = " << \*aclass << endl;  system("pause");  } |

# 4. СКРИНШОТЫ

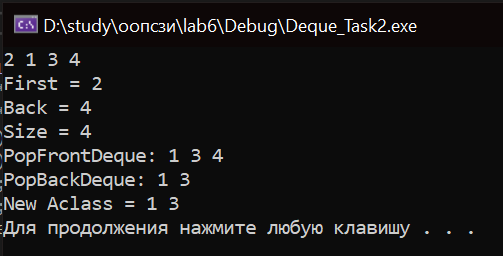


Рисунок 2 – скриншот выполнения программы Deque\_Task2.exe.

<https://github.com/PaSHe4ka1809/oopszi>

# 5. ВЫВОД

В данной лабораторной работе я овладел навыками создания базовых классов с виртуальными и чисто виртуальными функциями, а также производных классов с переопределением указанных функций.