МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

“БРЕСТСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ”

**ИНТЕЛЕКТУАЛЬНЫЕ ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

ОТЧЁТ

По лабораторной работе № \_\_

Выполнил:

Студент группы ИИ-22

Копанчук Евгений Романович

Проверил\_:

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

Брест – 202\_\_

**Ход работы**

**ЧАСТЬ 1**

**ТАБЛИЦА UML**

|  |  |  |  |
| --- | --- | --- | --- |
| << class >>  STACK |  | << class >>  QUEUE | |
| -+T\* arr  -int size |  | -T\* arr  -int size | |
| << Конструкторры >>  +STACK()  +~STACK()  << Методы >>  +void PUSH(T elem)  +T POP()  +void PRINT() |  | << Конструкторры >>  +QUEUE()  +~QUEUE()  << Методы >>  +void PUSH(T elem)  +T POP()  +void PRINT() | |
| Класс для реализации стека с помощью массива. |  | Класс для реализации очереди с помощью массива. | |
|  |  | |  |
| << class >>  TYPES |  | |
| -T1 a  -T2 b  -T3 c |  | |
| << Конструкторры >>  +TYPES(T1 a, T2 b, T3 c)  << Методы >>  +void SHOW() |  | |
| Класс для хранения и вывода трёх типов. |  | |

**MAIN и ВЫВОД**

|  |  |
| --- | --- |
| int main()  {  STACK<int> stack;  for (int i = 0; i < 10; i += 2)  stack.PUSH(i);  stack.PRINT();  stack.POP();  stack.PRINT();  QUEUE<int> queue;  for (int i = 0; i < 10; i += 2)  queue.PUSH(i);  queue.PRINT();  queue.POP();  queue.PRINT();  TYPES<int, char, string> types(324, 'a', "hello");  types.SHOW();  } |  |

**РЕАЛИЗАЦИИ МЕТОДОВ**

|  |  |
| --- | --- |
| **STACK** | **QUEUE** |
| template <typename T>  STACK<T>::STACK() {  size = 0;  arr = new T[size];  }  template <typename T>  STACK<T>::~STACK() {  delete[] arr;  }  template <typename T>  void STACK<T>::PUSH(T elem){  T\* buf = new T[size];  for (int i = 0; i < size; i++)  buf[i] = arr[i];  delete[] arr;  arr = new T[size + 1];  for (int i = 0; i < size; i++)  arr[i] = buf[i];  size++;  arr[size - 1] = elem;  }  template <typename T>  T STACK<T>::POP() {  size--;  T elem = arr[size];  T\* buf = new T[size];  for (int i = 0; i < size; i++)  buf[i] = arr[i];  delete[] arr;  arr = new T[size];  for (int i = 0; i < size; i++)  arr[i] = buf[i];  return elem;  }  template <typename T>  void STACK<T>::PRINT() {  cout << "STACK: ";  for (int i = 0; i < size; i++)  cout << arr[i] << ' ';  cout << endl;  } | template <typename T>  QUEUE<T>::QUEUE() {  size = 0;  arr = new T[size];  }  template <typename T>  QUEUE<T>::~QUEUE() {  delete[] arr;  }  template <typename T>  void QUEUE<T>::PUSH(T elem) {  T\* buf = new T[size];  for (int i = 0; i < size; i++)  buf[i] = arr[i];  delete[] arr;  arr = new T[size + 1];  for (int i = 0; i < size; i++)  arr[i] = buf[i];  size++;  arr[size - 1] = elem;  }  template <typename T>  T QUEUE<T>::POP() {  size--;  T elem = arr[0];  T\* buf = new T[size];  for (int i = 0; i < size; i++)  buf[i] = arr[i] + 1;  delete[] arr;  arr = new T[size];  for (int i = 0; i < size; i++)  arr[i] = buf[i];  return elem;  }  template <typename T>  void QUEUE<T>::PRINT() {  cout << "QUEUE: ";  for (int i = 0; i < size; i++)  cout << arr[i] << ' ';  cout << endl;  } |
| **TYPES** | |
| template<typename T1, typename T2, typename T3>  TYPES<T1, T2, T3>::TYPES(T1 a, T2 b, T3 c) { this->a = a; this->b = b; this->c = c; }  template<typename T1, typename T2, typename T3>  void TYPES<T1, T2, T3>::SHOW() { cout << a << ' ' << b << ' ' << c << endl; } | |