Лабораторная работа №1

По предмету «ПРОГРАММИРОВАНИЕ НА языке с++»

На тему «Динамические структуры данных. Динамические массивы»

Подготовил: Азиев Дамир (гр. 228-17)

Принял(-а): Усманова С.Ю

2018

## Цель работы:

Научиться создавать простейшие формы Windows Forms, работать с графическими компонентами управления TextBox, Label, DataGridView. Вспомнить алгоритмы работы с динамическими структурами данных и динамическими массивами в частности.

## Задание №14:

## Сформировать массив из случайно заданных элементов в диапазоне [-1000,1000]. Найти последний положительный элемент массива (его значение и индекс).

## Использованные компоненты:

Для выполнения данной работы мною были использованы следующие компоненты:

|  |
| --- |
| this->tlp\_page = (gcnew System::Windows::Forms::TableLayoutPanel());  this->dgv\_Array = (gcnew System::Windows::Forms::DataGridView());  this->tlp\_btns = (gcnew System::Windows::Forms::TableLayoutPanel());  this->pbt\_Generate = (gcnew System::Windows::Forms::Button());  this->pbt\_getLastPositive = (gcnew System::Windows::Forms::Button());  this->lb\_Text = (gcnew System::Windows::Forms::Label());  this->lb\_value = (gcnew System::Windows::Forms::Label());  this->tb\_arX = (gcnew System::Windows::Forms::TextBox());  this->tb\_arY = (gcnew System::Windows::Forms::TextBox());  this->label1 = (gcnew System::Windows::Forms::Label());  this->label2 = (gcnew System::Windows::Forms::Label()); |

**TableLayoutPanel** – представляет панель, в которой содержимое динамически отображается в сетке, состоящей из строк и столбцов. Используется для динамического изменения размеров содержимого при изменении размеров окна.

**DataGridView** – представляет собой таблицу для отображения символьных данных (по умолчанию). В данном проекте используется для отображения динамического массива.  
Для получения строки элементов используется параметр: **DataGridView::Rows** – возвращающий массив строк, перемещаться по которому можно как перебором, так и по индексу.   
Строки в свою очередь, имеют параметр: **Cells** – возвращающий массив ячеек в данной строке, для доступа к ним непосредственно.

**Label** - Представляет стандартную текстовую метку Windows. Позволяет изменять собственное содержимое через параметр **Text.**

**TextBox –** текстовое поле, предназначенное для редактирования и ввода текста. Так же как и у элемента Label текст элемента TextBox можно установить или получить с помощью свойства **Text**.

**Button** - Наиболее часто используемый элемент управления. Обрабатывая событие нажатия кнопки, мы может производить те или иные действия.

Для получения числовых данных из **TextBox** использовалась функция Convert::ToInt32(Object ^obj), которая получается содержимое из любых элементов с текстовым содержимым и преобразует его в целочисленное значение.

## Форма:

|  |  |
| --- | --- |
| При входе: | После генерации массива: |
| При нажатии кнопки Generate array без ввода значений: | После поиска последнего элемента: |
| При нажатии поиска кнопки Get last positive number без сгенерированного массива: | При повторной генерации массива: |

## Код программы:

Обработка кнопки генерации массива:

|  |
| --- |
| private: System::Void pbt\_Generate\_Click(System::Object^ sender, System::EventArgs^ e) {  if (tb\_arX->Text == "" || tb\_arY->Text == "") {  MessageBox::Show("You are not speciafied array size!", "Generation failed");  }  else {  std::srand(std::time(0));  dgv\_Array->Rows->Clear();  int X = Convert::ToInt32(tb\_arX->Text);  int Y = Convert::ToInt32(tb\_arY->Text);  dgv\_Array->ColumnCount = X;  dgv\_Array->RowCount = Y;  for (int dy = 0; dy < Y; ++dy) {  for (int dx = 0; dx < X; ++dx) {  dgv\_Array->Rows[dy]->Cells[dx]->Value = rand() % 2000 - 1000;  }  }  lb\_value->Text = "Not Counted";  }  } |

Обработка кнопки поиска числа:

|  |
| --- |
| private: System::Void pbt\_getLastPositive\_Click(System::Object^ sender, System::EventArgs^ e) {  if (dgv\_Array->RowCount == 0 || dgv\_Array->ColumnCount == 0) {  MessageBox::Show("You are not generated array!", "Processing failed");  }  else {  int X = dgv\_Array->ColumnCount;  int Y = dgv\_Array->RowCount;  int idx = -1,idy = -1;  for (int dy = 0; dy < Y; ++dy) {  for (int dx = 0; dx < X; ++dx) {  int val = Convert::ToInt32(dgv\_Array->Rows[dy]->Cells[dx]->Value);  if (val > 0) {  idx = dx;  idy = dy;  }  }  }  if (idx == -1 || idy == -1) lb\_value->Text = "Not found";  else {  lb\_value->Text = String::Format("{0} ({1};{2})",  dgv\_Array->Rows[idy]->Cells[idx]->Value,  idx+1, idy+1);  }  }  } |