# Супер-меню

***Цель работы:*** *развить навыки программирования на языке С++, посредством разработки удобного меню для консольных приложений.*

## 1 Создание проекта

Запустите Visual Studio.

1. Создайте новый проект (рис. 1):

* выберите *Файл → Создать → Проект…* (*File → New → Project…)* (либо нажмите Ctrl + Shift + N);
* выберите *Visual C++ → Консольное приложение Windows (Win32 Console Application)*;
* измените имя проекта (Name) на Menu;
* с помощью кнопки *Обзор… (Browse…)* укажите путь к проекту.

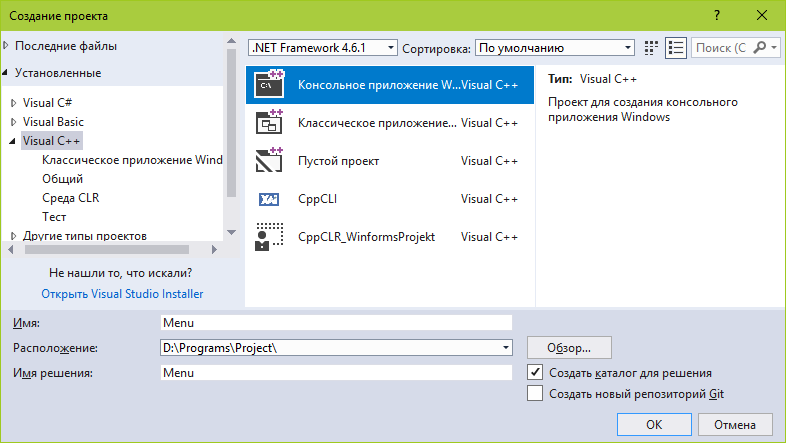


Рисунок 1 – Создание проекта

2. Изменим заголовок окна приложения и добавим задержку вывода на экран:

|  |
| --- |
| SetConsoleTitle(L"Заголовок консольного окна");  \_getch(); |

для чего подключим:

|  |
| --- |
| #include <windows.h>  #include <conio.h> |

**Практическое задание 1. Поменяйте заголовок окна, дабы в нем были отображены ваши фамилии и номер группы.**

## 2 Установка положения курсора

Напишем процедуру, которая устанавливает курсор в заданную позицию:

|  |
| --- |
| HANDLE hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE); // Получаем дескриптор консоли  // Текстовый курсор в точку x,y  void GoToXY(short x, short y)  {  SetConsoleCursorPosition(hStdOut, {x, y});  } |

Чтобы скрыть курсор (или поменять его размер), достаточно написать:

|  |
| --- |
| void ConsoleCursorVisible(bool show, short size)  {  CONSOLE\_CURSOR\_INFO structCursorInfo;  GetConsoleCursorInfo(hStdOut, &structCursorInfo);  structCursorInfo.bVisible = show; // изменяем видимость курсора  structCursorInfo.dwSize = size; // изменяем размер курсора  SetConsoleCursorInfo(hStdOut, &structCursorInfo);  } |

Ее вызов:

|  |
| --- |
| ConsoleCursorVisible(false, 100); |

**Практическое задание 2. Переместите курсор на центр экрана, убедитесь, что он находиться по центру, скройте его.**

## 3 Работа с экраном

Для отчистки экрана достаточно использовать лишь одну команду:

|  |
| --- |
| system("CLS"); |

Она отчищает экран и заполняет его цветом фона установленным в SetConsoleTextAttribute().

## 4 Считывание кода клавиш

Для получения кода клавиши достаточно следующего кода:

|  |
| --- |
| #include <iostream>  #include <conio.h>  using namespace std;  int main()  {  char ch;  ch = \_getch();    if (ch == -32) ch = \_getch(); // Отлавливаем стрелочки  } |

**Практическое задание 3. Добавьте в программу те строчки, которых еще нет в вашей программе, убедитесь в работоспособности кода.**

Для бесконечного считывания кода клавиш:

|  |
| --- |
| #include <iostream>  #include <ConIO.h>  using namespace std;  void main()  {  setlocale(0, "");  char ch;  while (true)  {  ch = \_getch();  if (ch == -32) ch = \_getch(); // Отлавливаем стрелочки  cout << "Код " << (int)ch << endl;  }  } |

**Практическое задание 4. Добавьте в программу те строчки, которых еще нет в вашей программе, убедитесь в работоспособности кода.**

**Узнайте код: нуля, единицы, двойки, клавиш Enter, Escape, пробел.**

Модифицируем нашу программу добавив соответствующий код:

|  |
| --- |
| #include <iostream>  #include <conio.h>  using namespace std;  int main()  {  setlocale(0, "");  char ch;  while (true)  {  ch = \_getch();  if (ch == -32) ch = \_getch(); // Отлавливаем стрелочки  switch (ch)  {  case 27:  exit(0);  case 72:  cout << "Up, Код " << (int)ch << endl;  break;  case 80:  cout << "Down, Код " << (int)ch << endl;  break;  case 75:  cout << "Left, Код " << (int)ch << endl;  break;  case 77:  cout << "Right, Код " << (int)ch << endl;  break;  default:  cout << "Код " << (int)ch << endl;  }  }  } |

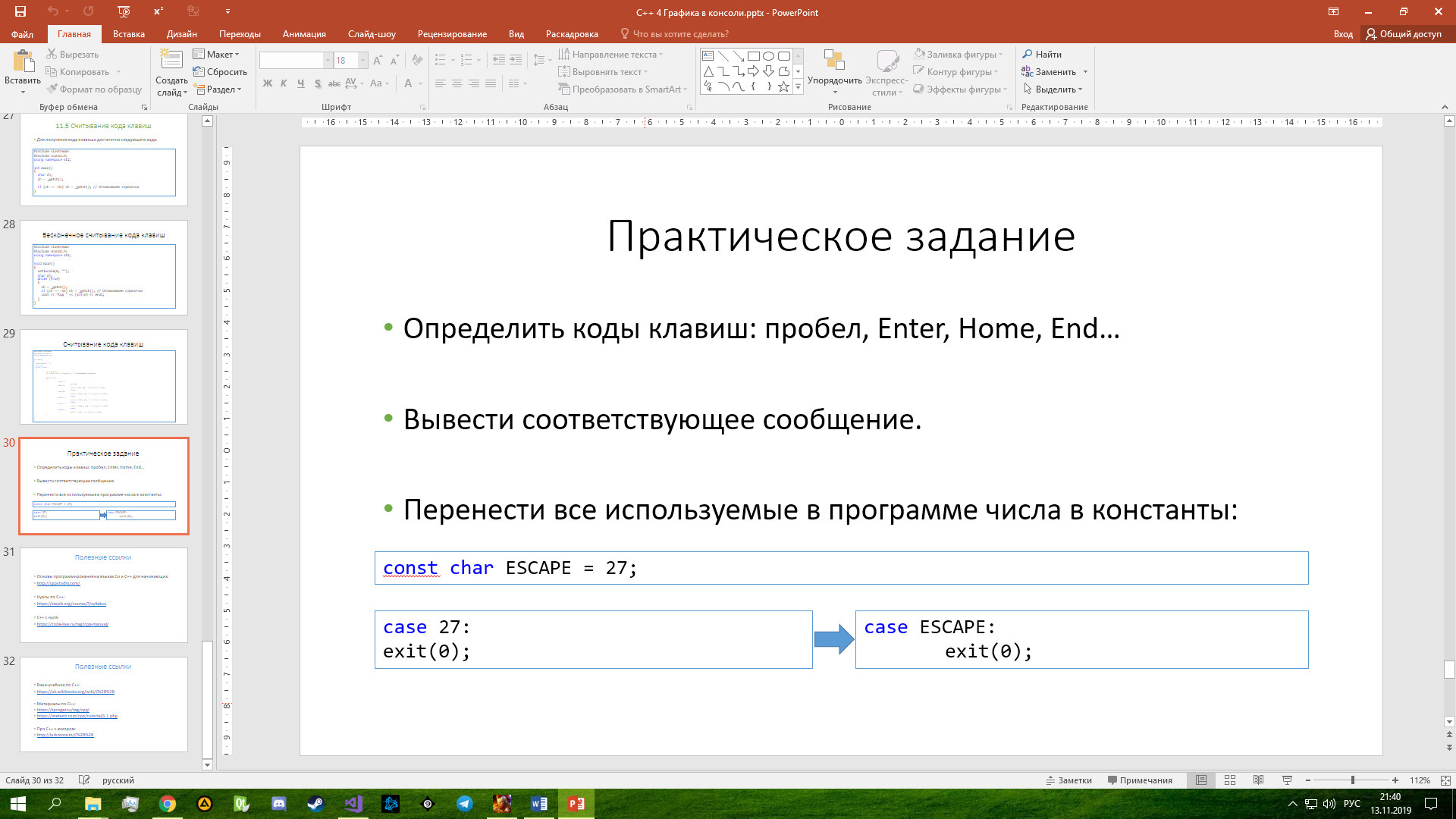
**Практическое задание 5. Добавьте в программу те строчки, которых еще нет в вашей программе, убедитесь в работоспособности кода.**

**Определить коды клавиш: пробел, Enter, Home, End.**

**Вывести соответствующее сообщение.**

**Перенести все используемые в программе числа в константы:**

|  |
| --- |
| const char ESCAPE = 27; |



## 5 Меню

Создадим массив для хранения тем меню:

|  |
| --- |
| string Menu[] = { "Новая игра", "Загрузить", "Настройки", "Выход" }; |

Дабы мы могли с ним нормально работать подключим:

|  |
| --- |
| #include <string> |

Выведем меню на экран (внутри бесконечного цикла):

|  |
| --- |
| int x = 40, y = 12;  GoToXY(x, y);    for (int i = 0; i < size(Menu); i++)  {  GoToXY(x, y++);  cout << Menu[i] << endl;  } |

Если size(Menu) не будет работать, то его можно заменить на

|  |
| --- |
| (sizeof(Menu) / sizeof(\*Menu)) |

Укажем, что первая (нулевая) тема меню сейчас активна (до бесконечного цикла):

|  |
| --- |
| int active\_menu = 0; |

Подсветим ее (в начале цикла с выводом меню):

|  |
| --- |
| if (i==active\_menu) SetConsoleTextAttribute(hStdOut, FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);  else SetConsoleTextAttribute(hStdOut, FOREGROUND\_GREEN); |

Изменим код для стрелочек вверх и вниз:

|  |
| --- |
| case UP:  active\_menu--;  break;  case DOWN:  active\_menu++;  break; |

**Практическое задание 6. Добавьте в программу те строчки, которых еще нет в вашей программе, убедитесь в работоспособности кода.**

**Сделайте так, чтобы при нажатии на стрелочку вниз, когда активен пункт «Выход», становился активен пункт «Новая игра».**

**Соответственно, при нажатии на стрелочку вверх, когда активен пункт «Новая игра», становился активен пункт «Выход».**

Измените код для клавиши Enter:

|  |
| --- |
| case ENTER:  switch (active\_menu)  {  case 0:  system("CLS");  GoToXY(x, y);  SetConsoleTextAttribute(hStdOut, FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_INTENSITY);  cout<< "Пользователь выбрал \""<< Menu[active\_menu] <<"\"";  \_getch();  system("CLS");  break;  case 3:  exit(0);  }  break; |

**Практическое задание 7. Добавьте в программу те строчки, которых еще нет в вашей программе, убедитесь в работоспособности кода.**

**Реализуйте аналогично работу пунктов «Загрузить» и «Настройки».**

**Практическое задание 8. Перенесите написанный вами код в процедуру. Вызывайте ее на выполнение при выборе соответствующего пункта.**