

ГОУ ВПО «ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ»  
ФИЗИКО-ТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ  
Кафедра Компьютерных технологий



УТВЕРЖДАЮ:

Проректор по научно-методической  
и учебной работе

Е.И. Скафа

«22» апреля 2020 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ  
«ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ  
НИЗКОГО УРОВНЯ»

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Профиль подготовки: Информатика и вычислительная техника

Образовательная программа: бакалавриат

Квалификация: академический бакалавр

Форма обучения: очная, очно-заочная, заочная, в том числе с ускоренным сроком обучения

Донецк 2020

УТВЕРЖДАЮ:

Декан физико-технического факультета

Фоменко С.А.

«17» апреля 2020 г.

М.П.



Программа учебной дисциплины «**Программирование на языках низкого уровня**» составлена на основе Государственного образовательного стандарта высшего профессионального образования (ГОС ВПО) по направлению подготовки 09.03.01 Информатика и вычислительная техника, утверждённого приказом Министерства образования и науки ДНР от «21» января 2016 г. №31»; «Порядка организации учебного процесса в образовательных организациях высшего профессионального образования Донецкой Народной Республики», утверждённого приказом Министерства образования и науки ДНР №1171 от «10» ноября 2017 г.»; учебного плана и основной образовательной программы высшего профессионального образования направления подготовки 09.03.01 Информатика и вычислительная техника, разработанных в ГОУ ВПО «Донецкий национальный университет».

Разработчики:

Старший преподаватель кафедры компьютерных технологий

Старший преподаватель кафедры компьютерных технологий

Котенко В.Н.

Котенко Ю.В.

Программа учебной дисциплины утверждена на заседании кафедры компьютерных технологий

Протокол № 12 от «2» апреля 2020 г.

Зав. кафедрой компьютерных технологий

Ермоленко Т.В.

Программа учебной дисциплины одобрена учебно-методической комиссией физико-технического факультета

Протокол № 5 от «15» апреля 2020 г.

Председатель учебно-методической комиссии физико-технического факультета

Котенко В.Н.

## 1. ОБЛАСТЬ ПРИМЕНЕНИЯ И МЕСТО ДИСЦИПЛИНЫ В УЧЕБНОМ ПРОЦЕССЕ

Учебная дисциплина «Программирование на языках низкого уровня» относится к вариативной части профессионального блока и состоит из двух содержательных модулей: модуль 1 – «Макросредства и функции языка Ассемблер», модуль 2 – «Консольные и оконные приложения».

Основывается на базе дисциплин: «Основы программирования», «Архитектура ЭВМ и микроконтроллеров», «ЭВМ и периферийные устройства». Является основой для изучения дисциплин: «Операционные системы».

## 2. СТРУКТУРА ДИСЦИПЛИНЫ

<i>Характеристика учебной дисциплины</i>				
Направление подготовки	09.03.01 Информатика и вычислительная техника			
Профиль	Информатика и вычислительная техника			
Образовательная программа	Бакалавриат			
Квалификация	Академический бакалавр			
Количество содержательных модулей	2			
Дисциплина базовой / вариативной части образовательной программы	Профессиональный блок. Вариативная часть			
Формы контроля (МК, экзамен, зачет)	Модульный контроль, зачёт			
Показатели	очная форма обучения		заочная форма обучения	
	нормат. срок	ускор. срок	нормат. срок	ускор. срок
Количество зачётных единиц (кредитов)	3	3	3	3
Год подготовки	3	2	3	2
Семестр	5	3	5	3
Количество часов	108	108	108	108
- лекционных	18	18	4	4
- практических, семинарских				
- лабораторных	36	36	10	10
- самостоятельной работы	54	54	94	94
в т. ч. индивидуальное задание				
Недельное количество часов, т. ч.	6	6	6	6
аудиторных	3	3	0.78	0.78

### 3. ОПИСАНИЕ ДИСЦИПЛИНЫ

#### Цели и задачи.

**Цель** – формирование знаний студента о фундаментальных понятиях, общих принципах организации и функционирования программ на машинно-ориентированных языках, методах и средствах проектирования и построения программ на языках низкого уровня программирования.

**Задачи** – усвоение теоретических основ и приобретение практических навыков по проектированию программ на языках низкого уровня в соответствии с техническим заданием; программированию на машинно-ориентированных языках; формированию понимания сущности процесса программирования на низком уровне; по современным инструментальным средствам низкоуровневого программирования; отладке программ на языках низкого уровня; составлению отчёта по выполненному заданию.

#### **Требования к результатам освоения дисциплины:**

Процесс изучения дисциплины направлен на формирование элементов следующих компетенций в соответствии с ГОС ВПО по данному направлению подготовки (профилю):

##### **а) общекультурных (ОК):**

способностью работать в коллективе, толерантно воспринимать социальные, этнические, конфессиональные и культурные различия (ОК-6);  
способностью к самоорганизации и самообразованию (ОК-7).

##### **б) общепрофессиональных (ОПК):**

основательная подготовка по математике для использования математического аппарата при решении прикладных и научных задач в области компьютерной инженерии (ОПК-1);  
знание современных методов построения и анализа алгоритмов, основ численных методов и умение их использовать на практике (ОПК-4).

##### **в) профессиональных (ПК):**

##### **проектно-конструкторская деятельность:**

знать архитектуру компьютеров, уметь применять их в процессе эксплуатации (ПК-1);  
пользоваться методиками использования программных средств для решения практических задач (ПК-2);  
использовать и самостоятельно разрабатывать интерфейсы взаимодействия человека и ЭВМ (ПК-3);

знание принципов программирования, средств современных языков программирования, структур данных (ПК-5);

знание архитектуры компьютеров (ПК-6);

##### **проектно-технологическая деятельность:**

знание особенностей системного программирования, владение методами и средствами разработки элементов системных программ (ПК-10);

знание особенностей построения системного программного обеспечения и общих принципов организации и функционирования операционных систем (ПК-11);

знание методологических принципов построения современных компьютерных систем разной организации для высокопродуктивной обработки информации (ПК-12);

знание теоретических (логических и арифметических) основ построения современных компьютеров и умение их использовать при решении профессиональных задач (ПК-13);

знание современных технологий и инструментальных способов разработки сложных программных систем (инженерии программного обеспечения), умение их использовать на всех этапах жизненного цикла программ (ПК-14);

##### **научно-исследовательская деятельность:**

базовые знания научно-методических основ и стандартов в области компьютерной инженерии, проводить эксперимент по проверке корректности решений, рассчитывать экономическую эффективность (ПК-15);

умение готовить и проводить доклады с использованием современных компьютерных средств, писать научно-технические отчёты, оформлять результаты исследований в виде статей (ПК-16);

**педагогическая деятельность:**

готовить конспекты лекций, проводить повышение квалификации сотрудников (ПК-17);

**сервисно-эксплуатационная деятельность:**

инсталлировать, настраивать и сопровождать программное и аппаратное обеспечение информационных и автоматизированных систем (ПК-21).

**В результате изучения учебной дисциплины студент должен**

**Знать:**

- современные языки программирования низкого уровня;
- основы современных концепций и технологий программирования на языках низкого уровня;
- технологии отладки программ на машинно-ориентированных языках.

**Уметь:**

- выполнять разработку элементов системного и прикладного программного обеспечения на языках низкого уровня;
- использовать возможности современных языков программирования низкого уровня;
- обосновывать выбор языка программирования низкого уровня в процессе разработки прикладного программного обеспечения;
- использовать макросредства и функции языков низкого уровня для создания консольных и оконных приложений.

**Владеть:**

- навыками разработки программного обеспечения на машинно-ориентированных языках;
- основами методики инсталляции средств разработки программ, ассемблирования, линковки и отладки программ на языках низкого уровня.

#### **4. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ И ФОРМЫ ОРГАНИЗАЦИИ УЧЕБНОГО ПРОЦЕССА**

<b>Порядковый номер и тема</b>	<b>Краткое содержание темы</b>
	<b><i>Содержательный модуль 1.</i></b> <b>Консольные приложения</b>
<b><i>Тема 1.</i></b> Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера	Преимущества знания низкоуровневых языков. Основные сведения об операционной системе Windows. Память в Windows. Исполняемые компоненты Windows. Системные библиотеки и подсистемы. Модель вызова функций в Win32. Выполнение программ в Win32. Типы структур программ Windows. Компиляторы Ассемблера. Ассемблеры для операционной системы DOS. Ассемблеры для операционной системы Windows. Ассемблеры для операционной системы Linux. Ассемблер для операционной системы ReactOS. Переносимые ассемблеры. AVR ассемблер.
<b><i>Тема 2.</i></b> Структура программы. Компиляция и компоновка	Использование визуальной интегрированной среды разработки приложений для создания исполняемого файла. Компиляция и компоновка программы с использованием командной строки и командного файла. Структура программы. Макродиректива Invoke. Форматированный вывод.

<b>Тема 3.</b> Процедуры	Процедуры. Команды работы со стеком. Синтаксис процедуры. Вызов и возврат из процедуры. Передача параметров в процедуру. Передача результата процедуры. Сохранение регистров в процедуре. Локальные данные процедур. Рекурсивные процедуры.
<b>Тема 4.</b> Консольные приложения	Понятие консольного приложения. Минимальное консольное приложение. Функции работы с консолью. Создание и освобождение консоли. Получение дескриптора устройства и установка заголовка окна. Вывод в консоль и чтение из буфера консоли. Определение размеров окна консоли, установка позиции курсора и атрибутов символов. Получение информации о клавиатуре и мыши. Пример консольного приложения.
	<b>Содержательный модуль 2.</b> <b>Оконные приложения</b>
<b>Тема 5.</b> Оконные приложения	Сообщения и их структура. Оконные сообщения и функции работы с окнами. Минимальное оконное приложение.
<b>Тема 6.</b> Элементы управления окна	Кнопка. Поле редактирования. Статический текст. Пример использования элементов управления.
<b>Тема 7.</b> Ресурсы приложений	Понятие ресурса. Стандартные и нестандартные ресурсы. Подключение ресурсов к исполняемому файлу. Создание собственной иконки приложения. Подключение меню к окну.
<b>Тема 8.</b> Работа с файлами в системе Windows.	Работа с файлами в системе Windows. Создание, открытие файла и закрытие файла. Удаление файла. Установка текущей файловой позиции. Получение размера файла. Чтение данных из файла. Запись данных в файл. Пример работы с файлами.

Курс дисциплины «Программирование на языках низкого уровня» предусматривает следующие **формы организации учебного процесса**:

- 1) лекции;
- 2) лабораторные занятия;
- 3) самостоятельная работа студента.

Электронные материалы по всем формам организации учебного процесса размещены на сайте KOTENKO.EU5.NET (<https://sites.google.com/site/kotenko1967/>).

По источнику передачи и восприятия учебной информации используются словесные (лекция, беседа), наглядные (иллюстрация, демонстрация), практические (исследования, упражнения, лабораторные работы) методы.

По характеру познавательной деятельности студентов используются объяснительно-иллюстративные и репродуктивные методы, проблемное преподавание, частично-поисковый и исследовательский методы.

В зависимости от основной дидактической цели и задач используются методы устного изложения знаний, закрепление учебного материала, самостоятельной работы студентов по осмыслению и усвоению нового материала, работы по применению знаний на практике и выработке умений и навыков, проверки и оценки знаний, умений и навыков.

Используются следующие методы контроля:

- 1) устный контроль (экспресс-опрос на лекциях);
- 2) проверка конспектов;
- 3) защита лабораторных работ;
- 4) проверка самостоятельных работ;
- 5) модульная контрольная работа (дидактическое тестирование);
- 6) итоговый тест.

## Тематический план

Названия содержательных модулей и тем	Содержательный модуль 1																			
	Количество часов																			
	Очная форма										Заочная форма									
	Нормативный срок обучения					Ускоренный срок обучения					Нормативный срок обучения					Ускоренный срок обучения				
	всего	В Т. Ч.				всего	В Т. Ч.				всего	В Т. Ч.				всего	В Т. Ч.			
		лекции	практические	лабораторные	самостоятельная работа		лекции	практические	лабораторные	самостоятельная работа		лекции	практические	лабораторные	самостоятельная работа		лекции	практические	лабораторные работы	самостоятельная работа
<b>Тема 1.</b> Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера	12	2		4	6	12	2		4	6	12	0,5		1	10,5	12	0,5		1	10,5
<b>Тема 2.</b> Структура программы. Компиляция и компоновка	13	2		4	7	13	2		4	7	13	0,5		1	11,5	13	0,5		1	11,5
<b>Тема 3.</b> Процедуры	13	2		4	7	13	2		4	7	13	0,5		1	11,5	13	0,5		1	11,5
<b>Тема 4.</b> Консольные приложения	16	3		6	7	16	3		6	7	16	0,5		2	13,5	16	0,5		2	13,5
<b>Итого содержательному модулю 1</b>	54	9		18	27	54	9		18	27	54	2		5	47	54	2		5	47

	Содержательный модуль 2																						
Названия содержательных модулей и тем	Количество часов																						
	Очная форма												Заочная форма										
							на базе общего среднего образования						на базе среднего профессионального образования						на базе высшего профессионального образования				
	всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.					всего	В Т. Ч.			
		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные	самостоятельная работа	индивидуальная работа		лекции	практические	лабораторные работы	самостоятельная работа
Тема 5. Оконные приложения	12	2		4	6		12	2		4	6		12	0,5		1	10,5		12	0,5		1	10,5
Тема 6. Элементы управления окна	13	2		4	7		13	2		4	7		13	0,5		1	11,5		13	0,5		1	11,5
Тема 7. Ресурсы приложений	13	2		4	7		13	2		4	7		13	0,5		1	11,5		13	0,5		1	11,5
Тема 8. Работа с файлами в системе Windows	16	3		6	7		16	3		6	7		16	0,5		2	13,5		16	0,5		2	13,5
Итого по содержательному модулю 2	54	9		18	27		54	9		18	27		54	2		5	47		54	2		5	47
Всего часов	108	18		36	54		108	18		36	54		108	4		10	94		108	4		10	94



## 5. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ДЛЯ ПРОВЕДЕНИЯ ЛЕКЦИОННЫХ, ПРАКТИЧЕСКИХ И ЛАБОРАТОРНЫХ ЗАНЯТИЙ

### Темы лекционных занятий

<i>№ п/п</i>	<i>Название темы</i>	<i>Количество часов</i>
1.	Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера	2
2.	Структура программы. Компиляция и компоновка	2
3.	Процедуры	2
4.	Консольные приложения	3
5.	Оконные приложения	2
6.	Элементы управления окна	2
7.	Ресурсы приложений	2
8.	Работа с файлами в системе Windows	3
	<b>ВСЕГО</b>	<b>18</b>

### Темы лабораторных занятий

<i>№ п/п</i>	<i>Название темы</i>	<i>Количество часов</i>
1.	Компиляция и компоновка программы	4
2.	Диалоговая структура программы. Форматированный вывод	4
3.	Процедуры	4
4.	Консольные приложения	6
5.	Оконные приложения	4
6.	Элементы управления окна	4
7.	Ресурсы приложений	4
8.	Работа с файлами в системе Windows	6
	<b>ВСЕГО</b>	<b>36</b>

## 6. МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ОРГАНИЗАЦИИ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ

### Организация самостоятельной работы студентов

Самостоятельная работа студентов по курсу «Программирование на языках низкого уровня» предусматривает:

- систематическое посещение лекционных занятий, ведение конспекта лекций;
- повседневное изучение лекционного материала и содержания технической литературы, рекомендуемые этой программой и рабочим учебным планом;
- добросовестную подготовку к лабораторным занятиям;
- самостоятельную разработку алгоритмов и текстов программ лабораторных работ.
- своевременное и качественное оформление отчётов по лабораторным работам.

<b>№ n/n</b>	<b>Название темы</b>	<b>Количество часов</b>
1.	Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера	6
2.	Структура программы. Компиляция и компоновка	7
3.	Процедуры	7
4.	Консольные приложения	7
5.	Оконные приложения	6
6.	Элементы управления окна	7
7.	Ресурсы приложений	7
8.	Работа с файлами в системе Windows	7
	<b>ВСЕГО</b>	<b>54</b>

## 7. ИНДИВИДУАЛЬНЫЕ ЗАДАНИЯ

Индивидуальные задания предусмотрены к каждой теме и полностью приведены в методических указания по выполнению и оформлению лабораторных работ к курсу «Программирование на языках низкого уровня».

Ниже приводится по одному примеру индивидуального задания из каждой темы:

1. Напишите и скомпилируйте первую программу на Masm под Windows с использованием среды разработки MS Visual Studio 2015.

2. Выполнить вычисление выражений в соответствии с номером варианта и осуществить форматированный вывод исходных данных и результатов расчетов, используя функции `wsprintfA` и `MessageBoxA`.

$$a) \quad x = \frac{1970 - 1935}{5} \times 5 - 5 + 548$$

$$b) \quad y = \frac{2210 - 210}{29 + 31 - 10} + \frac{1910 - 810}{100} \times 2$$

3. Написать программу, главная часть которой вызывает две процедуры, которые выполняют вычисления. Параметры в первую процедуру передаются через стек, результат возвращается через регистр EAX. Соглашение о вызовах – `stdcall`. Параметры во вторую процедуру передаются через стек, включая адрес ячейки памяти для результата, результат возвращается по адресу. Соглашение о вызовах – `cdecl`. Значения исходных параметров A, B, X, X1, X2, A1, A2 задать произвольно.

Процедура №1		Процедура №2	
Передаваемые параметры	Возвращаемые значения	Передаваемые параметры	Возвращаемые значения
Z = (A + B)/10 X	Y = 0, если Z > 0 или X < 0 Y = 1, если Z = 0 Y = 2, в противном случае	Y1 = A1 + A2 – 6	Y2 = 0, если Y1 > 0 Y2 = 1, если Y1 <= 0

4. Создать консольное приложение, выполняющее следующие действия: ввести строку, в которой между словами один или несколько пробелов. Сжать строку, оставив между словами по одному пробелу. Вывести сжатую строку и количество лишних пробелов.

5. Используя обработку сообщений `WM_KEYDOWN`, `WM_RBUTTONDOWN`, `WM_LBUTTONDOWN`, `WM_TIMER` и т.д. и API-функции `ShowWindow`, `CloseWindow`, `MoveWindow`, `SetWindowText`, `GetWindowTextLength`, `GetClientRect`, `GetTitleBarInfo`,

GetWindowPlacement, SetWindowPlacement, WindowFromPoint, AnimateWindow, SetTimer, KillTimer и т.д. создать оконное приложение, осуществляющее следующие действия: изменение текста заголовка окна на заданный текст по двойному щелчку правой кнопки в клиентской области окна Обратную замену заголовка осуществить по нажатию клавиши F1.

6. Создать оконное приложение с элементами управления «Кнопки», «Поля редактирования», «Статический текст»: калькулятор, выполняющий операции сложения, вычитания и умножения.

7. Создать оконное приложение с собственной иконкой и главным меню, содержащим пункты: «Задания», «Решения», «Помощь», «Выход».

Пункт меню «Задания» содержит подпункты: «Задание для лабораторной работы № 5», «Задание для лабораторной работы № 6», выбор которых приводит к отображению окна с информацией о заданиях на пятую и шестую лабораторные работы в соответствии с номером варианта.

Пункт меню «Решения» содержит подпункты: «Решение для лабораторной работы № 5», «Решение для лабораторной работы № 6», выбор которых приводит к отображению окон для решения заданий пятой и шестой лабораторных работ в соответствии с номером варианта.

Выбор пункта меню «Помощь» приводит к выводу на экран окна с кратким описанием работы приложения (две-три строки).

Выбор пункта меню «Выход» приводит к завершению работы программы.

8. Написать программу копирования содержимого одного файла в другой файл. Имя файла для работы программы передавать в командной строке.

## **8. КОНТРОЛЬНЫЕ ВОПРОСЫ К ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ**

1. Преимущества знания низкоуровневых языков.
2. Основные сведения об операционной системе Windows. Память в Windows. Исполняемые компоненты Windows.
3. Системные библиотеки и подсистемы.
4. Модель вызова функций в Win32. Выполнение программ в Win32.
5. Типы структур программ Windows.
6. Компиляторы Ассемблера.
7. Использование визуальной интегрированной среды разработки приложений для создания исполняемого файла.
8. Компиляция и компоновка программы с использованием командной строки и командного файла.
9. Структура программы.
10. Макродиректива Invoke.
11. Форматированный вывод.
12. Команды работы со стеком.
13. Синтаксис процедуры. Вызов и возврат из процедуры.
14. Передача параметров в процедуру.
15. Передача результата процедуры.
16. Сохранение регистров в процедуре.
17. Локальные данные процедур.
18. Минимальное консольное приложение.
19. Создание и освобождение консоли, получение дескриптора устройства и установка заголовка окна.
20. Вывод в консоль и чтение из буфера консоли.
21. Определение размеров окна консоли, установка позиции курсора и атрибутов символов.
22. Получение информации о клавиатуре и мыши.
23. Сообщения и их структура.

24. Оконные сообщения и функции работы с окнами.
25. Минимальное оконное приложение.
26. Элементы управления окна. Кнопка.
27. Поле редактирования. Статический текст.
28. Понятие ресурса. Стандартные и нестандартные ресурсы.
29. Подключение ресурсов к исполняемому файлу.
30. Создание собственной иконки приложения.
31. Подключение меню к окну.
32. Создание, открытие, закрытие и удаление файла.
33. Установка текущей файловой позиции. Получение размера файла.
34. Чтение данных из файла.
35. Запись данных в файл.

## 9. ОБРАЗЕЦ МОДУЛЬНОГО КОНТРОЛЯ

ГОУ ВПО «Донецкий национальный университет»

Физико-технический факультет

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Программа подготовки бакалавриат

Семестр 3

Учебная дисциплина Программирование на языках низкого уровня

### МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №1

#### ВАРИАНТ №1

*1. К основным системным библиотекам, содержащим API-функции, относятся:*

- a) bootstr.dll, comdlg32.dll, lsasrv.dll;
- b) mciavi32.dll, ntvdmd.dll, odbccjt32.dll;
- c) rasapi32.dll, secur32.dll, tapi32.dll;
- d) user32.dll, gdi32.dll, kernel32.dll;
- e) vspell32.dll, winsrv.dll, xwtpw32.dll.

*2. Адресное пространство любого процесса состоит из:*

- 1) стека и регистровой памяти;
- 2) памяти процесса и памяти системы;
- 3) памяти портов ввода-вывода и памяти системы;
- 4) памяти регистров сегментов, памяти регистров указателей и индексов;
- 5) памяти процесса, памяти регистра флагов.

*3. Модель памяти flat – метод организации адресного пространства, предполагающий:*

- 1) сегментированную конфигурацию программы;
- 2) раздельное хранение кода и данных;
- 3) указание начального адреса выполнения программы;
- 4) несегментированную конфигурацию программы;
- 5) очистку стека вызывающей программой.

#### 4. Секция – это:

- 1) структура, объединяющая части адресного пространства в единое целое;
- 2) главный модуль, с которого начинается выполнение программы;
- 3) фрагмент кода, содержащий основную часть программы;
- 4) блок программы, содержащий данные, код, метки;
- 5) фрагмент программы, предполагающий сегментированную организацию памяти.

#### 5. Согласно модели вызова системных функций *stdcall*:

- 1) параметры функций передаются через стек в прямом порядке, стек очищает вызываемая функция, результат будет содержаться в регистре EAX;
- 2) параметры функций передаются через стек в прямом порядке, стек очищает вызывающая программа, результат будет содержаться в регистре EAX;
- 3) параметры функций передаются через стек в обратном порядке, стек очищает вызываемая функция, результат будет содержаться в регистре EAX;
- 4) параметры функций передаются через стек в обратном порядке, стек очищает вызывающая программа, результат будет содержаться в регистре EBX;
- 5) параметры функций передаются через стек в прямом порядке, стек очищает вызывающая программа, результат будет содержаться в ячейке памяти.

#### 6. Главная задача *ML.exe* – это:

- 1) создать готовую программу под конкретную операционную систему и её формат;
- 2) скомпоновать программу, т.е. создать *dll*-файлы;
- 3) пошагово отладить исходный программный код;
- 4) скомпоновать объектные файлы и библиотеки в исполняемые модули определенного формата;
- 5) перевести все команды ассемблера из текстового исходного кода в байты машинных команд;
- 6) подключить к программе указанные API-функции.

#### 7. Укажите вид командной строки *LINK.exe*:

- 1) `LINK [ключи] список_файлов [/link<ключи_линковщика>]`
- 2) `LINK [ключи] [входные_obj-файлы] [@файл_опций]`
- 3) `LINK [/link<ключи_линковщика>]список_файлов [ключи]`
- 4) `LINK [входные_dll-файлы][@файл_опций] [ключи]`
- 5) `LINK [входные_dll-файлы] [ключи][@файл_опций]`

#### 8. Укажите верное утверждение:

- 1) директива `.IF` повторяет выполнение *<фрагмент\_программы>*, пока условие истинно;
- 2) директива `.REPEAT` выполняет *<фрагмент\_программы\_1>*, если логическое выражение истинно, и *<фрагмент\_программы\_2>*, если оно ложно;

- 3) директива .IF выполняет <фрагмент\_программы\_1>, если логическое выражение истинно, и <фрагмент\_программы\_2>, если оно ложно;
- 4) директива .WHILE повторяет выполнение <фрагмент\_программы>, пока условие не истинно;
- 5) директива .BREAK повторяет выполнение <фрагмент\_программы\_2>, пока условие не истинно.

9. Для чего используется макродиректива INVOKE?

- 1) создания дескриптора окна консольного приложения;
- 2) помещения в стек содержимого всех регистров;
- 3) вызова API-функций с проверкой количества и типов параметров;
- 4) создания dll-библиотек с проверкой количества и типов параметров;
- 5) вывода параметров функции на экран.

10. Соглашение *stdcall* предусматривает сохранение содержимого регистров... и результат выполнения функции будет содержаться в регистре....:

- 1) EAX, EBX, ECX, EDX, результат в EAX
- 2) ESI, ECX, EAX, результат в EDX
- 3) EAX, ESP, EBP, результат в EBX
- 4) EDX, EAX, результат в EAX
- 5) EBX, EBP, ESI, EDI, результат в EAX

11. Какой шестнадцатеричный код будет содержаться в регистре EDX в результате выполнения следующего фрагмента кода?

```
mov ebx, -104
add ebx, 4
neg ebx
mov eax, 214
cdq
idiv ebx
```

- 1) 0038H
- 2) 000EH
- 3) 00BEN
- 4) 0014H
- 5) 00C8H
- 6) 0002H

12. Что делает команда POP?

- 1) кладёт данные на вершину стека, значение регистра ESP уменьшает на размер операнда;
- 2) берёт данные из стека, помещает в указанный регистр или ячейку памяти, значение регистра ESP увеличивает на размер операнда;

- 3) берёт данные из стека, помещает в указанный регистр или ячейку памяти, значение регистра ESP уменьшает на размер операнда;
- 4) кладёт данные на вершину стека, значение регистра ESP увеличивает на размер операнда;
- 5) восстанавливает из стека содержимое всех регистров общего назначения;
- 6) берёт данные из стека, помещает в указанный регистр или ячейку памяти, значение регистра ESP не изменяет.

*13. Укажите правильный формат команды помещения данных в стек:*

- 1) pop <операнд>
- 2) mov <операнд1>, <операнд2>
- 3) push <операнд>
- 4) push <операнд1>, <операнд2>
- 5) mov <операнд>
- 6) pop <операнд1>, <операнд2>

*14. В каком регистре хранится адрес вершины стека?*

- 1) EAX
- 2) ESI
- 3) EBP
- 4) ESP
- 5) EDI
- 6) EBX

*15. Какая команда сохраняет в стеке все 32 бита регистра флагов:*

- 1) pushfd
- 2) pushd
- 3) popa
- 4) popad
- 5) popf

*16. Какая функция используется для получения дескриптора стандартного устройства?*

- 1) CallWindowProcA
- 2) DispatchMessageA
- 3) EnumDisplayDevicesW
- 4) GetClassInfoW
- 5) GetStdHandle
- 6) GetMenuItemID

*17. Для создания своей консоли необходимо использовать функцию:*

- 1) AttachConsole
- 2) AllocConsole
- 3) PeekConsoleInput
- 4) SetConsoleCP

5) SetConsoleMode

6) CreateConsoleScreenBuffer

18. Написать программу, главная часть которой вычисляет значение выражения:  $Z = M - N/5$  и передаёт через стек в процедуру пять параметров: адрес ячейки памяти  $Y$  для результата, значение  $Z$ ,  $Q$ ,  $P1$  и  $P2$ . Соглашение о вызовах cdecl.

Значения исходных параметров:  
 $M = 20$ ,  $N = 5$ ,  $Q = -3$ ,  $P1 = 50$ ,  $P2 = 10$ .

Возвращаемые значения:

$Y = P1 + P2$ , если  $Z * Q < 1$

$Y = P1 - P2$ , если  $Z * Q > 1$

$Y = 0$ , если  $Z * Q = 1$

Очистить стек.

Утверждено на заседании кафедры компьютерных технологий,  
 протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой  
 Преподаватель

Ермоленко Т.В.  
 Котенко В.Н.

#### Критерии оценивания модульного контроля

Номер задания	Количество баллов
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	8
<b>Всего</b>	<b>25</b>



ГОУ ВПО «Донецкий национальный университет»

Физико-технический факультет

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Программа подготовки бакалавриат

Семестр 3

Учебная дисциплина Программирование на языках низкого уровня

## МОДУЛЬНАЯ КОНТРОЛЬНАЯ РАБОТА №2

### ВАРИАНТ №1

*1. Что является главным элементом оконного приложения?*

- 1) сообщение;
- 2) окно;
- 3) кнопка;
- 4) список;
- 5) рабочий стол Windows.

*2. Что является отличительным признаком сообщения?*

- 1) ASCII-код сообщения;
- 2) дескриптор сообщения;
- 3) код, которому сопоставляется символическая константа;
- 4) текст сообщения;
- 5) функция сообщения.

*3. Код системных сообщений лежит в диапазоне:*

- 1) 03FFH – 07FFH;
- 2) 0000 – 0FFFFH;
- 3) 03FFH – 0FFFFH;
- 4) 0001 – 03FFH;
- 5) 07FFH – 0FFFFH.

*4. В очередь какого потока ставится сообщение из системной очереди:*

- 1) потока операционной системы;
- 2) потока, активизировавшего рабочий стол Windows;
- 3) потока дочернего процесса текущего процесса;
- 4) первого из потоков запущенных приложений;
- 5) потока, активизировавшего текущее окно.

*5. С помощью какой структуры сообщения передаются в приложения?*

- 1) HWND\_MSG;
- 2) MSG\_MAIN;
- 3) MGSP\_MAIN;
- 4) WINMAIN;
- 5) MSG.

6. Сообщения от кнопок имеют префикс?

- 1) WM\_
- 2) EM\_
- 3) LB\_
- 4) BM\_
- 5) BS\_

7. Дескрипторы объектов и системных ресурсов Window находятся в подключаемом файле:

- 1) hwnd.inc;
- 2) system.inc;
- 3) windows.inc;
- 4) kernel.inc;
- 5) hsystem.inc.

8. Для всех окон верхнего уровня родительским окном является:

- 1) окно последнего запущенного приложения;
- 2) рабочий стол Windows;
- 3) окно первого запущенного приложения;
- 4) окно, над которым расположен курсор;
- 5) окно с максимальными размерами.

9. Что делает функция *SetWindowText*?

- 1) копирует текст строки из буфера в заголовок окна. Если окно – элемент управления, текст из буфера копируется в элемент управления;
- 2) возвращает длину (количество символов) текста строки заголовка для окна, если окно имеет область заголовка;
- 3) копирует текст из буфера в клиентскую область окна;
- 4) устанавливает позицию курсора в клиентской области окна для вывода текста на экран;
- 5) выводит текст сообщения, посланного системой или устройством на экран.

10. При нажатии кнопки система генерирует сообщение:

- 1) WS\_CHILD;
- 2) WM\_CREATE;
- 3) WM\_COMMAND;
- 4) WS\_CAPTION;
- 5) WM\_BUTTON.

11. С помощью какой функции осуществляется создание окна:

- 1) RegisterWindow;
- 2) SetWindow;
- 3) CreateWindowState;
- 4) CreateWindowEx;
- 5) CreateMDIWindowA.

*12. Функция `GetModuleHandleA`:*

- 1) извлекает сообщение из очереди;
- 2) получает дескриптор окна;
- 3) получает дескриптор экземпляра приложения;
- 4) регистрирует класс окна;
- 5) устанавливает состояние показа модуля приложения.

*13. Создание элементов управления окна осуществляется функциями:*

- 1) `CreateWindow` и `CreateWindowEx`;
- 2) `CreateButton`, `CreateStatic` и `CreateEdit`;
- 3) `CreateButton`, `CreateComboBox` и `CreateListBox`;
- 4) `RegisterButton`, `RegisterStatic` и `RegisterEdit`;
- 5) `RegisterButton`, `RegisterComboBox` и `RegisterListBox`.

*14. Как понять, что текущее сообщение `WM_COMMAND` было послано меню, а не элементом управления окна?*

- 1) по параметру `lParam = 1`;
- 2) по параметру `wParam = 1`;
- 3) по параметру `wParam = 0`;
- 4) по ID меню и коду уведомления;
- 5) по параметру `lParam = 0`.

*15. При создании элементов управления окна обязательно указание стиля:*

- 1) `WS_GROUP`;
- 2) `WS_OVERLAPPED`;
- 3) `WS_CHILD`;
- 4) `WS_SYSMENU`;
- 5) `WS_DLGFRAME`.

*16. Ресурсы программы:*

- 1) определяются в самом начале файла программы;
- 2) определяются в секции данных файла программы;
- 3) определяются в рамках оконной функции `WndProc`;
- 4) создаются отдельно от файлов программы и добавляются в исполняемый файл при линковке программы;
- 5) определяются в \*.bat файле.

*17. Компиляция ресурсов осуществляется:*

- 1) компилятором ресурсов `rc.exe`;
- 2) компилятором ресурсов `link.exe`;
- 3) компилятором ресурсов `ml.exe`;
- 4) компилятором ресурсов `ml.exe` и `link.exe`;
- 5) компилятором ресурсов `res.exe` и `link.exe`.

18. Редактором ресурсов создан файл *ICON.RC*, в котором записана строка *ICON\_MAIN ICON ICON.ICO*. В какую секцию и какую строку нужно добавить, чтобы подключить ресурс к приложению?

- 1) в секцию кода строку *IconName dw 'ICON.ICO',0;;*
- 2) в секцию данных строку *IconName dw 'ICON.ICO',0;*
- 3) в секцию кода строку *IconName db 'ICON\_MAIN',0;*
- 4) в секцию данных строку *IconName db 'ICON\_MAIN',0;*
- 5) в секцию стека строку *IconName db 'ICON',0.*

19. Главное меню программы:

- 1) сетевая структура;
- 2) одноуровневая структура;
- 3) трехуровневая структура;
- 4) древовидная структура.
- 5) матричная структура.

20. Главное меню программы:

- 1) не отображается на экране и не содержит ни одного элемента;
- 2) отображается на экране и содержит перечень идентификаторов действия;
- 3) отображается на экране и содержит указанное количество элементов;
- 4) не отображается на экране и содержит указанное количество элементов;
- 5) отображается на экране и не содержит ни одного элемента.

21. У каких объектов есть характеристика – идентификатор действия?

- 1) *MENUITEM*-ы;
- 2) *POPUP*-меню;
- 3) элементы главного меню;
- 4) активные окна;
- 5) элементы управления окна.

22. Ключевое слово *POPUP*:

- 1) специфицирует *MENUITEM*-ы;
- 2) специфицирует элементы одного уровня меню;
- 3) специфицирует элементы главного меню;
- 4) специфицирует обычный элемент меню;
- 5) специфицирует всплывающее меню.

23. Вызов функции *TranslateMessage()* необходим:

- 1) во всех приложениях;
- 2) во всех приложениях, работающих с манипулятором «мышь»;
- 3) в приложениях, которые обрабатывают ввод данных с клавиатуры;
- 4) в приложениях, обрабатывающих ввод данных с клавиатуры и мыши;
- 5) не является обязательным.

24. Вызов функции *CreateFileA* возвращает:

- 1) полное имя файла;
- 2) сокращенное имя файла;
- 3) дескриптор файла;
- 4) режим открытия файла;
- 5) режим создания файла.

25. *Закрытие файла осуществляется функцией с передачей параметра:*

- 1) CloseFile, параметр «полное имя файла»;
- 2) CloseFile, параметр «краткое имя файла»;
- 3) CloseHandle, параметр «полное имя файла»;
- 4) CloseHandle, параметр «краткое имя файла»;
- 5) CloseHandle, параметр «описатель файла».

Утверждено на заседании кафедры компьютерных технологий,  
протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой  
Преподаватель

Ермоленко Т.В.  
Котенко В.Н.

#### Критерии оценивания модульного контроля

<i>Номер задания</i>	<i>Количество баллов</i>
1	1
2	1
3	1
4	1
5	1
6	1
7	1
8	1
9	1
10	1
11	1
12	1
13	1
14	1
15	1
16	1
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
<b><i>Всего</i></b>	<b>25</b>

## 10. ОБРАЗЕЦ ЭКЗАМЕНАЦИОННОГО БИЛЕТА

Экзамен не предусмотрен программой.

## 11. ОБРАЗЕЦ ТЕСТОВОГО ЗАДАНИЯ

1. К основным системным библиотекам, содержащим API-функции, относятся:

- 1) bootstr.dll, comdlg32.dll, lsasrv.dll;
- 2) mciavi32.dll, ntvdmd.dll, odbccjt32.dll;
- 3) rasapi32.dll, secur32.dll, tapi32.dll;
- 4) user32.dll, gdi32.dll, kernel32.dll;
- 5) vspell32.dll, winsrv.dll, xwtpw32.dll.

2. Код системных сообщений лежит в диапазоне:

- 1) 03FFH – 07FFH;
- 2) 0000 – 0FFFFH;
- 3) 03FFH – 0FFFFH;
- 4) 0001 – 03FFH;
- 5) 07FFH – 0FFFFH.

3. У каких объектов есть характеристика – идентификатор действия?

- 1) MENUITEM;
- 2) POPUP-меню;
- 3) элементы главного меню;
- 4) активные окна;
- 5) элементы управления окна.

## 12. КРИТЕРИИ ОЦЕНИВАНИЯ

*Распределение баллов, которые могут получить студенты  
в процессе изучения дисциплины*

	Содержательный модуль №1							Содержательный модуль №2						Всего	
	Лабораторные работы				Конс-пект	Мод. контр. работа	Всего С.М. №1	Лабораторные работы				Конс-пект	Мод. контр. работа		Всего С.М. №2
	№1	№2	№3	№4				№5	№6	№7	№8				
Макс. балл	5	5	5	5	5	25	50	5	5	5	5	5	25	50	100

Согласно модульному принципу организации учебного процесса содержание дисциплины «Программирование на языках низкого уровня» включает в себя два зачётных модуля. Каждый зачётный модуль состоит из теоретического материала и практических задач, выполнение которых требует овладения теорией в указанном в модуле объёме.

К первому модульному контролю студент должен защитить 4 лабораторные работы. За первую, вторую, третью и четвертую лабораторные работы студент может получить по 5 баллов. В 5 баллов оценивается ведение конспекта лекций.

На первом модульном контроле студент имеет возможность получить 25 баллов, ответив правильно на 17 тестовых вопросов, каждый из которых оценивается в 1 балл, и решив одну практическую задачу, которая оценивается в 8 баллов.

Ко второму модульному контролю студент должен защитить 4 лабораторные работы. За пятую, шестую, седьмую и восьмую лабораторные работы студент может получить по 5 баллов. В 5 баллов оценивается ведение конспекта лекций.

На втором модульном контроле студент имеет возможность получить 25 баллов, ответив правильно на 25 тестовых вопросов, каждый из которых оценивается в 1 балл.

Оценка за овладение курса выставляется по следующим критериям:

– Оценку «отлично» заслуживает студент, который обнаружил глубокие знания при ответах на теоретические вопросы по темам курса, а также выполнил практические задания в полном объеме и набрал более 90 баллов.

– Оценку «хорошо» заслуживает студент, сделавший ошибки в теоретических или практических ответах, которые могут быть интерпретированы как малозначительные для вопросов, которые рассматривались. Студент должен набрать более 75 баллов.

– Оценку «удовлетворительно» заслуживает студент, который выполнил задания неполно и с ошибками, но при этом набрал более 60 баллов.

– Оценку «неудовлетворительно» заслуживает студент, который не выполнил большинства теоретических и практических задач и набрал менее 60 баллов.

#### ***Шкала соответствия баллов национальной шкале***

<b>Оценка по шкале ECTS</b>	<b>Оценка по 100-балльной шкале</b>	<b>Оценка по государственной шкале (экзамен, дифференцированный зачет)</b>	<b>Оценка по государственной шкале (зачет)</b>
<b>A</b>	90-100	5 (отлично)	зачтено
<b>B</b>	80-89	4 (хорошо)	зачтено
<b>C</b>	75-79	4 (хорошо)	зачтено
<b>D</b>	70-74	3 (удовлетворительно)	зачтено
<b>E</b>	60-69	3 (удовлетворительно)	зачтено
<b>FX</b>	35-59	2 (неудовлетворительно) с возможностью повторной сдачи	не зачтено
<b>F</b>	0-34	2 (неудовлетворительно) с возможностью повторной сдачи при условии обязательного набора дополнительных баллов	не зачтено

### **13. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА**

Лекционные занятия проводятся в аудитории, оснащенной мультимедийной техникой и доской.

Лабораторные занятия проводятся в компьютерном классе, оборудованном компьютерами с лицензионным программным обеспечением, доступом к сети Интернет, столами и доской.

#### 14. РЕКОМЕНДОВАННАЯ ЛИТЕРАТУРА

№ п/п	Наименование	Кол-во экземпляров в библиотеке ДонНУ	Наличие электронной версии в ЭБС
<i>Основная литература</i>			
1.	Котенко В.Н. «Программирование на языках низкого уровня»: курс лекций / В. Н. Котенко, Ю.В. Котенко. – Донецк: ГОУ ВПО «ДонНУ», 2016. – 85 с.	100	Да
2.	Котенко В.Н. Методические указания к выполнению и оформлению лабораторных работ к курсу «Программирование на языках низкого уровня» / В.Н. Котенко, Ю.В. Котенко. - Донецк: ГОУ ВПО «ДонНУ», 2016. – 71 с.	100	Да
<i>Дополнительная литература</i>			
3.	Пирогов В.Ю. Ассемблер для Windows. 4-е издание / В.Ю. Пирогов. – СПб.: БХВ-Петербург., 2015. – 875 с.	35	Да

#### 15. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. Котенко В. Н. Лекции по дисциплине «Программирование на языках низкого уровня». URL: [https://sites.google.com/site/kotenko1967/2\\_LLPL/2\\_lekcii\\_LLPL](https://sites.google.com/site/kotenko1967/2_LLPL/2_lekcii_LLPL) (дата обращения 01.03.2020 г.)

2. Котенко В. Н., Котенко Ю.В. Лабораторные работы по дисциплине «Программирование на языках низкого уровня». URL: [https://sites.google.com/site/kotenko1967/2\\_LLPL/3\\_laboratornye\\_raboti\\_LLPL](https://sites.google.com/site/kotenko1967/2_LLPL/3_laboratornye_raboti_LLPL) (дата обращения 01.03.2020 г.)

3. Котенко В.Н. Программирование на языках низкого уровня. Дистанционный курс в системе Moodle. URL: <http://dl.donnu.ru/course/info.php?id=77> (дата обращения 02.03.2020 г.)

4. Котенко В.Н. Группа ВКонтакте. [https://vk.com/low\\_level\\_program](https://vk.com/low_level_program)

5. Котенко В.Н. Облако Mail.ru. <https://cloud.mail.ru/public/3rqr/5Q228H4J6>

6. Assembler. URL: <http://www.chitalkino.ru/assembler/> (дата обращения 02.03.2020 г.)

7. Аблязов Р. 3. Программирование на ассемблере на платформе x86-64 URL: [https://codernet.ru/books/assembler/programmirovanie\\_na\\_assemblere\\_na\\_platforme\\_x86\\_64/](https://codernet.ru/books/assembler/programmirovanie_na_assemblere_na_platforme_x86_64/) (дата обращения 03.03.2020 г.)

8. Assembler для Windows. URL: <http://i-assembler.ru/> (дата обращения 03.03.2020 г.)

9. Программирование на языке ассемблера.

URL: <http://natalia.appmat.ru/c&c++/assembler.html> (дата обращения 04.03.2020 г.)

10. Структура оконного приложения URL: <http://prog-cpp.ru/winmain/> (дата обращения 04.03.2020 г.)

11. Элементы управления окна URL: <http://prog-cpp.ru/winelements/> (дата обращения 04.03.2020 г.)

12. The MASM32 SDK. URL: <http://www.masm32.com/> (дата обращения 05.03.2020 г.)

#### 18. Программное обеспечение

1. Microsoft Visual Studio 2015 и старших версий.

2. Microsoft Macro Assembler версии 6.15 или более старших версий.



Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2020 год.

Протокол № 12 от «2» апреля 2020 г.

Заведующий кафедрой

Ермоленко Т.В.

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2021 год.

Протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 2021 г.

Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2022 год.

Протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 2022 г.

Заведующий кафедрой

Рабочая программа рассмотрена и переутверждена на заседании кафедры компьютерных технологий с изменениями (без изменений) на 2023 год.

Протокол № \_\_\_\_ от «\_\_\_\_» \_\_\_\_\_ 2023 г.

Заведующий кафедрой