

Министерство науки и высшего образования
Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Донецкий государственный университет»

Физико-технический факультет
Кафедра компьютерных технологий



УТВЕРЖДАЮ
проректор

П.А. Машаров

«29» марта 2024 г.

МП

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ НИЗКОГО УРОВНЯ

Укрупненная группа направлений
подготовки

Программа высшего образования

Направление подготовки

Профиль подготовки

Квалификация

Форма обучения

09.00.00 Информатика и вычислительная
техника

Программа бакалавриата

09.03.01 Информатика и вычислительная
техника

Информатика и вычислительная техника

Бакалавр

Очная, заочная

Рабочая программа адаптирована для лиц
с ограниченными возможностями здоровья и инвалидов

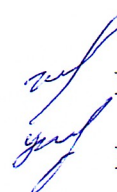
Донецк 2024

Рабочая программа дисциплины «Программирование на языках низкого уровня» для обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника (Профиль подготовки: Информатика и вычислительная техника), составлена на основании Федерального государственного образовательного стандарта высшего образования – бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника, утвержденного приказом Министерства образования и науки Российской Федерации от 19 сентября 2017 г. № 929 (с изм. и доп.), Порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, утвержденного приказом Министерства науки и высшего образования Российской Федерации от 06 апреля 2021 г. № 245 (с изм. и доп.), в соответствии с учебным планом, утвержденным Ученым советом ФГБОУ ВО «ДонГУ» для набора 2024 года.

Разработчики:

Старший преподаватель кафедры
компьютерных технологий

Старший преподаватель кафедры
компьютерных технологий



В.Н. Котенко

Ю.В. Котенко

Рабочая программа одобрена на заседании кафедры компьютерных технологий.
Протокол от 26.03.2024 г. № 12

Заведующий кафедрой



Г.В. Аверин

СОГЛАСОВАНО:

Декан физико-технического факультета
28.03.2024 г.



С.А. Фоменко

Учебно-методическая комиссия физико-технического факультета.
Протокол от 27.03.2024 г. № 2

Председатель



В. Н. Котенко

Руководитель основной профессиональной
образовательной программы,
д-р технических наук, проф.
26.03.2024 г.



Г.В. Аверин

1. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ

1.1. Требования к предварительной подготовке обучающихся, предшествующие и сопутствующие дисциплины, на которых основывается изучение данной:

дисциплины программы бакалавриата: «Основы программирования», «Архитектура ЭВМ», «ЭВМ и периферийные устройства».

1.2. Дисциплины, курсовые работы и практики, для которых освоение данной дисциплины необходимо как предшествующее:

«Операционные системы», «Производственная: технологическая (проектно-технологическая) практика», «Производственная: научно-исследовательская работа», «Производственная: преддипломная практика».

2. ОПИСАНИЕ ДИСЦИПЛИНЫ

2.1. Общая характеристика

Наименование показателя	Значение показателя
Название образовательной программы	09.03.01 Информатика и вычислительная техника (Профиль подготовки: Информатика и вычислительная техника)
Шифр и название в соответствии с учебным планом	Б1.Б.М4.10. Программирование на языках низкого уровня
Часть образовательной программы	Базовая часть
Количество зачетных единиц / всего часов	3 / 108

2.2. Распределение часов по периодам обучения

Форма обучения	курс	семестр	Общее количество часов					Форма контроля
			лекционных	лабораторных	практических	самостоятельной работы	всего	
Очная	3	5	17	34	–	57	108	зачет
Очная, всего			17	34	–	57	108	
Заочная	3	5	2	8	–	98	108	зачет
Заочная, всего			2	8	–	98	108	

3. ЦЕЛИ ДИСЦИПЛИНЫ

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 программирования низкого уровня, 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.

**4. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ
ОСВОЕНИЯ КОМПОНЕНТА ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ, ИХ ИНДИКАТОРЫ
И ПЛАНИРУЕМЫЕ РЕЗУЛЬТАТЫ ОБУЧЕНИЯ**

Компетенции	Индикаторы	Результаты обучения
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	ОПК-8.1. Знать: алгоритмические языки программирования, операционные системы и оболочки, современные среды разработки программного обеспечения	ОПК-8.1.1. Знает алгоритмические языки программирования, синтаксис выбранного языка программирования, особенности программирования на этом языке, стандартные библиотеки языка программирования
		ОПК-8.1.2. Знает операционные системы и оболочки
		ОПК-8.1.3. Знает современные среды разработки программного обеспечения
	ОПК-8.2. Уметь: составлять алгоритмы, писать и отлаживать коды на языке программирования, тестировать работоспособность программы, интегрировать программные модули	ОПК-8.2.1. Умеет составлять алгоритмы
		ОПК-8.2.2. Умеет писать и отлаживать коды на языке программирования
		ОПК-8.2.3. Умеет тестировать работоспособность программы
		ОПК-8.2.4. Умеет применять выбранные языки программирования для написания программного кода, интегрировать программные модули
	ОПК-8.3. Владеть: языком программирования; навыками отладки и тестирования работоспособности программы	ОПК-8.3.1. Владеет навыками программирования
		ОПК-8.3.2. Владеет навыками создания программного кода в соответствии с техническим заданием
		ОПК-8.3.3. Владеет навыками отладки работоспособности программы
		ОПК-8.3.4. Владеет навыками тестирования работоспособности программы
ОПК-9. Способен осваивать методики использования программных средств для решения практических задач	ОПК-9.1. Знать: классификацию программных средств и возможности их применения для решения практических задач	ОПК-9.1.1. Знает классификацию программных средств
		ОПК-9.1.2. Знает возможности применения программных средств для решения практических задач
	ОПК-9.2. Уметь: находить и анализировать техническую документацию по использованию программного средства, выбирать и использовать необходимые функции	ОПК-9.2.1. Умеет находить и анализировать техническую документацию по использованию программного средства
		ОПК-9.2.2. Умеет выбирать необходимые функции программных средств для решения конкретной задачи
		ОПК-9.2.3. Умеет использовать необходимые функции программных средств для решения конкретной задачи

	программных средств для решения конкретной задачи	
	ОПК-9.3. Владеть: способами описания методики использования программного средства для решения конкретной задачи в виде документа, презентации или видеоролика	ОПК-9.3.1. Владеет способами описания методики использования программного средства для решения конкретной задачи в виде документа
		ОПК-9.3.2. Владеет способами описания методики использования программного средства для решения конкретной задачи в виде презентации
		ОПК-9.3.3. Владеет способами описания методики использования программного средства для решения конкретной задачи в виде видеоролика

5. ПРОГРАММА ДИСЦИПЛИНЫ

Наименование темы	Краткое содержание темы (вопросы темы)
<i>Раздел 1. Консольные приложения</i>	
<i>Тема 1.</i> 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.
<i>Тема 2.</i> 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.
<i>Тема 3.</i> 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.
<i>Тема 4.</i> 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.
<i>Раздел 2. Оконные приложения</i>	
<i>Тема 5.</i> Оконные приложения	Сообщения и их структура. Оконные сообщения и функции работы с окнами. Минимальное оконное приложение.
<i>Тема 6.</i>	Кнопка. Поле редактирования. Статический текст. Пример использования элементов управления.

Элементы управления окна	
Тема 7. Ресурсы приложений	Понятие ресурса. Стандартные и нестандартные ресурсы. Подключение ресурсов к исполняемому файлу. Создание собственной иконки приложения. Подключение меню к окну.
Тема 8. 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.

6. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

6.1. Форма обучения – очная, курс – 3, семестр – 5

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС	Всего
Раздел .1 Консольные приложения					
Тема 1. Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера	2	4	–	6	12
Тема 2. Структура программы. Компиляция и компоновка	2	4	–	7	13
Тема 3. Процедуры	2	4	–	7	13
Тема 4. Консольные приложения	3	6	–	7	16
Раздел 2. Оконные приложения					
Тема 5. Оконные приложения	2	4	–	6	12
Тема 6. Элементы управления окна	2	4	–	7	13
Тема 7. Ресурсы приложений	2	4	–	7	13
Тема 8. Работа с файлами в системе Windows.	2	4	–	10	16
ИТОГО ЗА 5 СЕМЕСТР	17	34	–	57	108
ИТОГО ПО КОМПОНЕНТУ ОПОП	17	34	–	57	108

6.2. Форма обучения – заочная, курс – 3, семестр – 5

Наименования разделов и тем	Количество часов				
	Лекц.	Лабор.	Практ.	СРС	Всего
Раздел .1 Консольные приложения					
Тема 1. Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера	0,25	1	–	10,75	12
Тема 2. Структура программы. Компиляция и компоновка	0,25	1	–	11,75	13
Тема 3. Процедуры	0,25	1	–	11,75	13
Тема 4. Консольные приложения	0,25	1	–	14,15	16
Раздел 2. Оконные приложения					
Тема 5. Оконные приложения	0,25	1	–	10,75	12
Тема 6. Элементы управления окна	0,25	1	–	11,75	13
Тема 7. Ресурсы приложений	0,25	1	–	11,75	13
Тема 8. Работа с файлами в системе Windows.	0,25	1	–	14,75	16
ИТОГО ЗА 5 СЕМЕСТР	2	8	–	98	108
ИТОГО ПО КОМПОНЕНТУ ОПОП	2	8	–	98	108

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ (СРЕДСТВА) ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

7.1. Контрольные вопросы

Раздел 1.

Консольные приложения

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

Раздел 2.

Оконные приложения

1. Сообщения и их структура.
2. Оконные сообщения и функции работы с окнами.
3. Минимальное оконное приложение.
4. Элементы управления окна. Кнопка.
5. Поле редактирования. Статический текст.
6. Понятие ресурса. Стандартные и нестандартные ресурсы.
7. Подключение ресурсов к исполняемому файлу.
8. Создание собственной иконки приложения.
9. Подключение меню к окну.
10. Создание, открытие, закрытие и удаление файла.
11. Установка текущей файловой позиции. Получение размера файла.
12. Чтение данных из файла.
13. Запись данных в файл.

7.2. Образец задания на контрольную работу

КОНТРОЛЬНАЯ РАБОТА №1

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

- a) bootstr.dll, comdlg32.dll, lsasrv.dll;
- b) mciavi32.dll, ntvdm.dll, odbccp32.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.

КРИТЕРИИ ОЦЕНИВАНИЯ ЗАДАНИЯ НА КОНТРОЛЬНУЮ РАБОТУ №1

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
Всего	25

КОНТРОЛЬНАЯ РАБОТА №2

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 – 0FFFH;
- 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) MSGP_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_DLGFAME*.

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».

КРИТЕРИИ ОЦЕНИВАНИЯ ЗАДАНИЯ НА КОНТРОЛЬНУЮ РАБОТУ №2

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	1
19	1
20	1
21	1
22	1
23	1
24	1
25	1
Всего	25

7.3. Темы лабораторных работ

Лабораторные работы по дисциплине предназначены для закрепления теоретических знаний и приобретения практических навыков по реализации изученных методов. Темы лабораторных работ перечислены ниже:

1. Исполняемые компоненты Windows. Системные библиотеки. Компиляторы Ассемблера.

2. Структура программы. Компиляция и компоновка.

3. Процедуры.

4. Консольные приложения.

5. Оконные приложения.

6. Элементы управления окна.

7. Ресурсы приложений.

8. Работа с файлами в системе Windows.

Содержание лабораторных работ и методические рекомендации к их выполнению приведены в электронном учебно-методическом комплексе дисциплины и в электронном репозитории учебных курсов ДонГУ.

Индивидуальные задания предусмотрены к каждой лабораторной работе и полностью приведены в учебно-методическом пособии «Программирование на языках низкого уровня».

Ниже приводится по одному примеру индивидуального задания из каждой лабораторной работы:

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

2. Выполнить вычисление выражений в соответствии с номером варианта и осуществить форматированный вывод исходных данных и результатов расчетов, используя функции `wprintfA` 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 $Y1 \leq 0$

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».

8. 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.

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

Требования к оформлению отчета по лабораторной работе:

Отчет оформляют на листах белой бумаги формата А4 с одной стороны компьютерным способом с помощью текстового редактора Microsoft Word. Размеры полей: левое – 25 мм, правое – 15 мм, верхнее и нижнее – 2 мм.

Текст отчета печатается шрифтом Times New Roman размером 14 pt с полуторным междустрочным интервалом и абзацным отступом 1,27 см. Изображение шрифта обычное, выравнивание по ширине строки.

Отчет по лабораторной работе включает в себя:

1. Фамилия, Имя, Отчество (например, «Иванов Иван Иванович»).
2. Название группы (например, «Группа 3 ИВТ-1»).
3. Название дисциплины (например, Дисциплина: «Программирование на языках низкого уровня»).
4. Номер лабораторной работы (например, «Лабораторная работа №1»).
5. Тема лабораторной работы (например, «Тема: «СТРУКТУРА ПРОГРАММЫ. КОМПИЛЯЦИЯ И КОМПОНОВКА»»).
6. Цель лабораторной работы.
7. Контрольные вопросы и ответы на них.
8. Вариант индивидуального задания (например, «Вариант №3»).
9. Задание варианта к лабораторной работе.
10. Код программы для решения индивидуального задания работы.
11. Скриншоты результатов работы программы.

8. РАСПРЕДЕЛЕНИЕ БАЛЛОВ, КОТОРЫЕ ПОЛУЧАЮТ ОБУЧАЮЩИЕСЯ

Общая оценка знаний обучающихся по дисциплине проводится по 100-балльной шкале исходя из максимума, приведенного в таблице ниже. Организационно-учебная работа в аудитории оценивается на основе таких критериев как посещаемость занятий, своевременное и качественное выполнение домашних заданий, активность во время проведения лекционных и практических занятий (участие в обсуждении текущего и пройденного материала, решение задач и т.п.).

8.1. Семестр 5

	Содержательный модуль №1							Содержательный модуль №2							Всего
	Лабораторные работы				Конс-пект	Контр. работа №1	Всего раздел №1	Лабораторные работы				Конс-пект	Контр. работа №2	Всего раздел №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

Содержание дисциплины «Программирование на языках низкого уровня» включает в себя два раздела. Каждый раздел состоит из теоретического материала и практических задач, выполнение которых требует овладения теорией в указанном в модуле объёме.

По первому разделу 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.

За первую контрольную работу студент имеет возможность получить 25 баллов, ответив правильно на 17 вопросов, каждый из которых оценивается в 1 балл, и решив одну практическую задачу, которая оценивается в 8 баллов.

По второму разделу 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.

За вторую контрольную работу студент имеет возможность получить 25 баллов, ответив правильно на 25 вопросов, каждый из которых дает 1 балл.

Всего по первому разделу студент может набрать 50 баллов, по второму разделу 50 баллов. Всего по компоненту ОПОП студент может набрать 100 баллов.

Соответствие баллов оценке

Количество баллов из 100	ECTS	Оценка по пятибалльной шкале	
		Экзамен, дифференцированный зачет	Зачет
90-100	A	отлично	зачтено
80-89	B	хорошо	зачтено
75-79	C		зачтено
70-74	D	удовлетворительно	зачтено
60-69	E		зачтено
35-59	FX	неудовлетворительно	не зачтено
0-34	F		не зачтено

9. ОБЕСПЕЧЕНИЕ ОБРАЗОВАТЕЛЬНОГО ПРОЦЕССА ДЛЯ ЛИЦ С ОГРАНИЧЕННЫМИ ВОЗМОЖНОСТЯМИ ЗДОРОВЬЯ И ИНВАЛИДОВ

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- 1) для слепых и слабовидящих:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств;
 - письменные задания оформляются увеличенным шрифтом.
- 2) для глухих и слабослышащих:
 - лекции оформляются в виде электронного документа;
 - письменные задания выполняются на компьютере в письменной форме;
 - экзамен проводится в письменной форме на компьютере; возможно проведение в форме тестирования.

- 3) для лиц с нарушениями опорно-двигательного аппарата:
 - лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением;
 - письменные задания выполняются на компьютере;
 - экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- 1) для слепых и слабовидящих:
 - в печатной форме увеличенным шрифтом;
 - в форме электронного документа;
- 2) для глухих и слабослышащих:
 - в печатной форме;
 - в форме электронного документа.
- 3) для обучающихся с нарушениями опорно-двигательного аппарата:
 - в печатной форме;
 - в форме электронного документа.

10. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ УЧЕБНОГО ПРОЦЕССА

Учебные занятия проводятся в 4-м (пр. Театральный, 13) учебном корпусе университета.

Для проведения лекционных занятий требуется аудитория, оборудованная меловой или маркерной доской, мультимедийный проектор и экран, ноутбук, комплект учебной мебели для студентов, рабочее место преподавателя, выход в Интернет проводной или с использованием Wi-Fi.

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

Для самостоятельной работы используются текстовые и электронные ресурсы Научной библиотеки университета и других электронных библиотечных баз данных, материально-техническая база учебных лабораторий «Программного обеспечения общего назначения» (ауд. 419), «Специального программного обеспечения» (ауд. 415) и «Программного обеспечения систем искусственного интеллекта» (ауд. 413) кафедры компьютерных технологий.

В процессе обучения студенты имеют возможность использовать учебные материалы по дисциплине «Программирование на языках низкого уровня», размещенные на платформе Moodle Центра дистанционного образования ФГБОУ ВО «ДонГУ».

С использованием ресурсов платформы дистанционного образования также осуществляется текущий контроль знаний студентов на основе тестирования и проверки результатов самостоятельной работы.

11. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

11.1. Основная литература

1. Котенко В.Н. «Программирование на языках низкого уровня» [Электронный ресурс]: курс лекций / В. Н. Котенко, Ю.В. Котенко. – Донецк: ДонГУ, 2023. – 85 с. – Электронные данные (1 файл)
2. Котенко В.Н. «Программирование на языках низкого уровня» [Электронный ресурс]: учебно-методическое пособие / В.Н. Котенко, Ю.В. Котенко. – Донецк: ДонГУ, 2021. – 94 с. – Электронные данные (1 файл)

11.2. Дополнительная литература

3. Пирогов В.Ю. Ассемблер для Windows. 4-е издание / В.Ю. Пирогов. – СПб.: БХВ-Петербург., 2015. – 875 с.

12. ИНФОРМАЦИОННЫЕ РЕСУРСЫ

1. **Национальная электронная библиотека (НЭБ):** федеральная государственная информационная система / Министерство Культуры РФ; Российская государственная библиотека. – Москва, 2019- . – URL: <https://rusneb.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный, подписка. Необходима установка программного обеспечения. – Текст: электронный.
2. **eLIBRARY.RU:** научная электронная библиотека: сайт. – Москва, 2000- . – URL: <https://elibrary.ru> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
3. Научная электронная библиотека «КиберЛенинка»: сайт / Ассоциация «Открытая наука». – Москва, 2014- . – URL: <https://cyberleninka.ru/>. – Режим доступа: свободный. – Текст: электронный.
4. Электронно-библиотечная система «Лань»: [сайт]. – URL: <https://e.lanbook.com> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
5. **ЭБС Юрайт:** электронная библиотечная система: сайт. – Москва, 2013. – URL: <https://biblio-online.ru> (дата обращения: 01.09.2023). – Режим доступа: для авторизов. пользователей. – Текст: электронный.
6. **Электронно-библиотечная система ДонГУ:** сайт / ФГБОУ ВО «ДонГУ». – Донецк, 2016- . – URL: <http://library.donnu.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный. – Текст: электронный.
7. **Электронный каталог** Научной библиотеки ДонГУ: раздел сайта / НБ ДонГУ. – Текст: электронный // ЭБС ДонГУ: сайт. – URL: <http://library.donnu.ru/catalog/> (дата обращения: 01.09.2023). – Режим доступа: поиск свободный, электронные документы – для пользователей ДонГУ.
8. **Электронный архив ДонГУ:** раздел сайта / НБ ДонГУ. – Текст: электронный // ЭБС ДонГУ: сайт. – URL: <http://repo.donnu.ru/> (дата обращения: 01.09.2023). – Режим доступа: свободный.
9. Котенко В. Н. Лекции по дисциплине «Программирование на языках низкого уровня» URL: https://sites.google.com/site/kotenko1967/2_LLPL/2_lekcii_LLPL (дата обращения 15.03.2024 г.) – Режим доступа: свободный. – Текст: электронный.
10. Котенко В. Н. Лабораторные работы по дисциплине «Программирование на языках низкого уровня» URL: https://sites.google.com/site/kotenko1967/2_LLPL/3_laboratornye_raboti_LLPL (дата обращения 15.03.2024 г.) – Режим доступа: свободный. – Текст: электронный.

13. ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

1. Windows 7 PRO (корпоративная лицензия ДонГУ № 46484614)
2. Microsoft Office (корпоративная лицензия ДонГУ № 46472919)
3. Microsoft Visual Studio (лицензия программы Dream Spark для высших учебных заведений)
4. Антивирус Касперского, Adobe Acrobat Reader, xPDF (лицензии GPL, Apache, BSD для свободного программного обеспечения).
5. Microsoft Macro Assembler версии 6.15 или более старших версий.
6. Visual Assembler 1.0 или более старших версий.
7. Эмулятор DosBox версии 0.74 или более старших версий.