

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА»**

Кафедра «Вычислительная и прикладная математика»

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ
«Машино-зависимые языки программирования»**

Направление подготовки
09.03.04 «Программная инженерия»

Направленность (профиль) подготовки
«Программное обеспечение систем искусственного интеллекта»

Уровень подготовки – бакалавриат

Квалификация выпускника – бакалавр

Форма обучения – очная

Срок обучения – 4 года

Рязань

1. ОБЩИЕ ПОЛОЖЕНИЯ

Оценочные материалы – это совокупность учебно-методических материалов и процедур, предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций, обучающихся целям и требованиям основной образовательной программы в ходе проведения текущего контроля и промежуточной аттестации.

Основная задача – обеспечить оценку уровня сформированности компетенций и индикаторов их достижения, приобретаемых обучающимся в соответствии с этими требованиями.

Контроль знаний обучающихся проводится в форме текущего контроля и промежуточной аттестации.

Текущий контроль успеваемости и промежуточная аттестация проводятся с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся, организации работы обучающихся в ходе учебных занятий и оказания им индивидуальной помощи.

К контролю текущей успеваемости относятся проверка знаний, умений и навыков обучающихся на практических занятиях по результатам выполнения и защиты обучающимися индивидуальных заданий, по результатам выполнения контрольных работ и тестов, по результатам проверки качества конспектов лекций и иных материалов.

В качестве оценочных средств на протяжении семестра используется устные и письменные ответы студентов на индивидуальные вопросы, письменное тестирование по теоретическим разделам курса, реферат. Дополнительным средством оценки знаний и умений студентов является отчет о выполнении практических заданий и его защита.

По итогам курса обучающиеся сдают экзамен. Форма проведения – устный ответ с письменным подкреплением по утвержденным билетам, сформулированным с учетом содержания дисциплины. В билет для экзамена включается два теоретических вопроса и задача. В процессе подготовки к устному ответу студент должен составить в письменном виде план ответа.

1. Перечень компетенций с указанием этапов их формирования

При освоении дисциплины формируются следующие компетенции: ПК-1 (индикаторы ПК-1.1, ПК-1.2, ПК-1.3).

Указанные компетенции формируются в соответствии со следующими этапами:

- формирование и развитие теоретических знаний, предусмотренных указанными компетенциями (лекционные занятия, самостоятельная работа студентов);
- приобретение и развитие практических умений предусмотренных компетенциями (практические занятия, самостоятельная работа студентов);
- закрепление теоретических знаний, умений и практических навыков, предусмотренных компетенциями, в ходе решения конкретных задач на занятиях, выполнения индивидуальных заданий на практических занятиях и их защиты, а так же в процессе сдачи экзамена.

2. Показатели и критерии оценивания компетенций (результатов) на различных этапах их формирования, описание шкал оценивания

Сформированность каждой компетенции в рамках освоения данной дисциплины оценивается по трехуровневой шкале:

- пороговый уровень является обязательным для всех обучающихся по завершении освоения дисциплины;
- продвинутый уровень характеризуется превышением минимальных

характеристик сформированности компетенций по завершении освоения дисциплины;

– эталонный уровень характеризуется максимально возможной выраженностью компетенций и является важным качественным ориентиром для самосовершенствования.

При достаточном качестве освоения более 80% приведенных знаний, умений и навыков преподаватель оценивает освоение данной компетенции в рамках настоящей дисциплины на эталонном уровне, при освоении более 60% приведенных знаний, умений и навыков – на продвинутом, при освоении более 40% приведенных знаний, умений и навыков – на пороговом уровне. При освоении менее 40% приведенных знаний, умений и навыков компетенция в рамках настоящей дисциплины считается неосвоенной.

Уровень сформированности каждой компетенции на различных этапах ее формирования в процессе освоения данной дисциплины оценивается в ходе текущего контроля успеваемости и представлено различными видами оценочных средств.

Оценке сформированности в рамках данной дисциплины подлежат компетенции/индикаторы:

Показатели достижения планируемых результатов обучения и критерии их оценивания на разных уровнях формирования компетенций приведены в таблице 1.

Таблица 1. Показатели достижения индикаторов компетенции

1	2	3	4
Компетенция: код по ФГОС 3++, формулировка	Индикаторы	Этап	Наименование оценочного средства
ПК-1 Владеет навыками использования различных технологий разработки программного обеспечения, включая современные	ПК-1.1 Руководит процессом разработки программного обеспечения ЗНАТЬ: - методы проектирования программного обеспечения и его программную реализации. УМЕТЬ: - применять методы проектирования программного обеспечения и его программную реализацию. ВЛАДЕТЬ: - навыками проектирования программного обеспечения и его программной реализацией. ПК-1.2 Руководит проверкой работоспособности программного обеспечения ЗНАТЬ: - базовые способы проверки работоспособности программного обеспечения, а также наиболее простые способы интеграции программных модулей и компонентов УМЕТЬ:	1	Экзамен.

1	2	3	4
	- проводить проверку работоспособности и рефакторинг кода программного обеспечения ВЛАДЕТЬ: - методами проверки работоспособности кода программного обеспечения, интеграции программных модулей и компонентов разнообразных информационных систем, для большинства платформ и операционных систем ПК-1.3 Организует внедрение и сопровождение разработанного программного обеспечения ЗНАТЬ: - методологию внедрения программного обеспечения УМЕТЬ: - осуществлять разработку, документирование всех настроек, создавать систему поддержки и адекватное обучение пользователей ВЛАДЕТЬ: - всеми этапами сопутствующими внедрению и сопровождению разработанного программного обеспечения		

Преподавателем оценивается содержательная сторона и качество материалов, приведенных в отчетах студента по практическим занятиям. Кроме того, преподавателем учитываются ответы студента на вопросы по соответствующим видам занятий при текущем контроле:

- контрольные опросы;
- задания для практических занятий.

Принимается во внимание **знания** обучающимися:

- методов проектирования программного обеспечения и его программную реализации;

- базовых способов проверки работоспособности программного обеспечения, а также наиболее простых способов интеграции программных модулей и компонентов;
- методологию внедрения программного обеспечения;

наличие **умений**:

- применять методы проектирования программного обеспечения и его программную реализацию;

- проводить проверку работоспособности и рефакторинг кода программного обеспечения;

- осуществлять разработку, документирование всех настроек, создавать систему поддержки и адекватное обучение пользователей;

обладание навыками:

- проектирования программного обеспечения и его программной реализацией;
- проверки работоспособности кода программного обеспечения, интеграции программных модулей и компонентов разнообразных информационных систем, для большинства платформ и операционных систем.

Критерии оценивания уровня сформированности компетенции в процессе выполнения практических работ:

41%-60% правильных ответов соответствует пороговому уровню сформированности компетенции на данном этапе ее формирования;

61%-80% правильных ответов соответствует продвинутому уровню сформированности компетенции на данном этапе ее формирования;

81%-100% правильных ответов соответствует эталонному уровню сформированности компетенции на данном этапе ее формирования.

Сформированность уровня компетенций не ниже порогового является основанием для допуска обучающегося к промежуточной аттестации по данной дисциплине.

Формой промежуточной аттестации по данной дисциплине является экзамен, оцениваемый по принятой в ФГБОУ ВО «РГРТУ» четырехбалльной системе: «неудовлетворительно», «удовлетворительно», «хорошо» и «отлично».

Критерии оценивания промежуточной аттестации представлены в таблице.

Шкала оценивания	Критерии оценивания
«отлично»	студент должен: продемонстрировать глубокое и прочное усвоение знаний материала; исчерпывающе, последовательно, грамотно и логически стройно изложить теоретический материал; правильно формулировать определения; уметь сделать выводы по излагаемому материалу; безупречно ответить не только на вопросы билета, но и на дополнительные вопросы в рамках рабочей программы дисциплины; выполнить все практические задания, предусмотренные программой
«хорошо»	студент должен: продемонстрировать достаточно полное знание материала; продемонстрировать знание основных теоретических понятий; достаточно последовательно, грамотно и логически стройно излагать материал; уметь сделать достаточно обоснованные выводы по излагаемому материалу; ответить на все вопросы билета; выполнить все практические задания, предусмотренные программой.
«удовлетворительно»	студент должен: продемонстрировать общее знание изучаемого материала; знать основную рекомендуемую программой дисциплины учебную литературу; уметь строить ответ в соответствии со структурой излагаемого вопроса; показать общее владение понятийным аппаратом дисциплины; уметь устранить допущенные погрешности в ответе на теоретические вопросы; выполнить все практические задания, предусмотренные программой.
«неудовлетворительно»	ставится в случае: невыполнения практических занятий; незнания значительной части пройденного материала; не владения понятийным аппаратом дисциплины; существенных ошибок при изложении учебного материала; неумения строить ответ в соответствии со структурой излагаемого вопроса; неумения делать выводы по излагаемому материалу. Как правило, оценка «неудовлетворительно» ставится студентам, которые не могут продолжить обучение по образовательной

	программе без дополнительных занятий по соответствующей дисциплине (формирования и развития компетенций, закрепленных за данной дисциплиной). Оценка «неудовлетворительно» выставляется также, если студент после начала экзамена отказался его сдавать или нарушил правила сдачи экзамена (списывал, подсказывал, обманом пытался получить более высокую оценку и т.д.).
--	---

3. Типовые контрольные задания или иные материалы

ФОС по дисциплине содержит следующие оценочные средства, позволяющие оценить знания, умения и уровень приобретенных компетенций при текущем контроле и промежуточной аттестации, разбитые по модулям дисциплины:

- перечень экзаменационных вопросов;
- макет билета к экзамену.

Средства для оценки различных уровней формирования компетенций по категориям «знать», «уметь», «владеть» обеспечивают реализацию основных принципов контроля, таких, как объективность и независимость, практико-ориентированность, междисциплинарность.

С учетом этого, контрольные вопросы (задания, задачи,) входящие в ФОС, для различных категорий и уровней освоения компетенций имеют следующий вид:

Уровень ЗНАТЬ

Дескрипторы	Пример задания из оценочного средства
технологии разработки программного обеспечения (объектно-ориентированная и визуальная)	1. Что называют архитектурой процессора с точки зрения программиста. 2. Дать определение конвекции языка программирования высокого уровня. 3. Дать классификацию команд микропроцессора.

Перечень экзаменационных вопросов

1. Архитектура МП 8088 и 80386. Образование физического адреса.
2. Характеристики регистров.
3. Флаги.
4. Сегментные регистры по умолчанию.
5. Структура одномодульной программы MSDOS. Повторные описания сегментов.
6. Возможные структуры кодового сегмента.
7. Возможные способы начала выполнения и завершения программы MSDOS.
8. Структура программы из нескольких исходных модулей MSDOS.
9. Стандартные директивы описания сегментов.
10. Переменные, метки, символические имена и их атрибуты.
11. Виды предложений языка Ассемблер.
12. Директивы (псевдооператоры): назначение и формы записи.
13. Возможные комбинации сегментов и умолчания.
14. Директивы ASSUME, ORG, END.
15. Структура процедур.
16. Внешние имена.
17. Типы данных и задание начальных значений.
18. Способы описания меток, типы меток.

19. Команды условных переходов.
20. Команды организации циклов.
21. Способы адресации.
22. Организация рекурсивных подпрограмм.
23. Арифметические команды
24. Связывание подпрограмм.
25. Команда CALL. Использование прямой и косвенной адресации.
26. Способы передачи параметров подпрограмм.
27. Способы сохранения и восстановления состояния вызывающей программы.
28. Конвенции языков высокого уровня.
29. Команды сдвига.
30. Команды логических операций.
31. Команды обработки строк и префиксы повторения.
32. Команды пересылки строк.
33. Команды сравнения строк.
34. Команды сканирования строк.
35. Команды загрузки строк.
36. Команды сохранения строк.
37. Листинг программы.
38. Макросредства.
39. Макроопределения (макрофункций и макропроцедур) и макрокоманды.
40. Директива INCLUDE и LOCAL.
41. Рекурсия в макроопределениях.
42. Параметры в макросах.
43. Директивы условного ассемблирования и связанные с ними конструкции.
44. Директивы IFB и IFNB в макроопределениях.
45. Директивы IFIDN и IFDIF в макроопределениях.
46. Операции `;;`, `%`, `&<>`, `!` в макроопределениях.
47. Блоки повторения REPT, IRP/FOR, IRPC/FORC, WHILE.
48. Директива EQU и = в MASM.
49. Директива TEXTEQU в MASM32.
50. Типы макроданных text и number.
51. Именованные макроконстанты MASM32
52. Макроимена, числовые и текстовые макроконстанты.
53. Директивы echo и %echo
54. Способы вывода значений макропеременных и макроконстант с пояснениями
55. Операции в выражениях, вычисляемых препроцессором MASM:
56. Подготовка ассемблерных объектных модулей средствами командной строки для использования в средах разработки консольных приложений на ЯВУ.
57. Добавление ассемблерных модулей в проект консольного приложения С.
58. Добавление ассемблерных модулей в проект консольного приложения PASCAL.
59. Использование ассемблерных вставок в модулях на ЯВУ.
60. Вызов подпрограммы Сиз ассемблерной в VSC++.
61. Передача глобальных данных, определённых в консольной программе VSC++, в ассемблерный модуль.
62. Передача глобальных данных, определённых в ассемблерном модуле в консольный модуль С.
63. Средства отладки в средах разработки консольных приложений на ЯВУ.
Примеры.

Макет билета к экзамену

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования «Рязанский государственный радиотехнический университет
имени В.Ф. Уткина» (РГРТУ)

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1
по дисциплине «МАШИННО-ЗАВИСИМЫЕ ЯЗЫКИ ПРОГРАММИРОВАНИЯ»

1. Архитектура МП 8088. Образование физического адреса.
2. Способы адресации.
3. Операции: % &<> ! в макроопределениях.

Билет рассмотрен и утвержден на заседании кафедры «__» _____20 г.

Перечень лабораторных работ

Лабораторная работа 1.1. Освоение ассемблерного отладчика. Знакомство с шаблоном программы без процедур.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона из процедур, которая должна выводить коды символов в интервале символов, вводимых с клавиатуры.

Лабораторная работа 1.2. Освоение ассемблерного отладчика. Знакомство с шаблоном программы, использующей процедуры.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона без процедур. Программа должна вводить строку символов, находить в ней символ с максимальным кодом и выводить этот символ и все следующие за ним в строке.

Лабораторная работа 1.3. Одномодульные программы, вывод средствами DOS.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона из процедур.

Создать одномодульную программу на основе шаблона без процедур.

Лабораторная работа 1.4. Одномодульные программы, средства ввода-вывода

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать программу, которая должна выводить коды символов в интервале символов, вводимых с клавиатуры.

Лабораторная работа 1.5 Одномодульные программы, использование сегментов.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать программу. Программа должна вводить строку символов, находить в ней символ с максимальным кодом и выводить этот символ и все следующие за ним в строке.

Лабораторная работа 1.6 Многомодульные программы, подключение сегментов, внешние имена.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона из процедур, которая должна выводить коды символов в интервале символов, вводимых с клавиатуры.

Лабораторная работа 1.7. Способы адресации данных.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона из процедур, которая должна выводить коды символов в интервале символов, вводимых с клавиатуры.

Лабораторная работа 1.8. Способы адресации кода по данным в памяти и межпрограммный интерфейс.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона без процедур. Программа должна вводить строку символов, находить в ней символ с максимальным кодом и выводить этот символ и все следующие за ним в строке.

Лабораторная работа 1.9. Команды арифметические, логические, сдвигов, пересылки данных и передачи управления.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона из процедур, которая должна выводить коды символов в интервале символов, вводимых с клавиатуры.

Лабораторная работа 2.1. Средства отладки ассемблерных кодов в среде программирования консольных приложений на языке PASCAL. Вставки ассемблерных кодов в консольное приложение.

Цель работы. Научиться строить одномодульные программы с различного рода средствами ввода и вывода.

Задание

Создать одномодульную программу на основе шаблона без процедур. Программа должна вводить строку символов, находить в ней символ с максимальным кодом и выводить этот символ и все следующие за ним в строке.

Лабораторная работа 2.2. Средства отладки ассемблерных модулей в среде программирования консольных приложений на языке С. Рекурсивные ассемблерные подпрограммы в консольных приложениях.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке С с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Создать консольное приложение на языке С, добавить в него ассемблерный модуль с рекурсивной процедурой, вычисляющей $N!$ и сохраняющей его по заданному адресу.

Лабораторная работа 2.3 Ассемблерные модули для работы с битовыми строками в консольных приложениях на языке PASCAL.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке С с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Сравнить её дизассемблированный код с кодом подобной процедурой, но построенной на языке С.

Лабораторная работа 2.4 Ассемблерные модули для работы с битовыми строками в консольных приложениях на языке С.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке С с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Изменить код ассемблерного модуля так, чтобы минимизировать количество обращений к стеку с соблюдением конвенции С.

Лабораторная работа 2.5 .Команды обработки строк в ассемблерных модулях консольных приложений.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке С с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Создать консольное приложение на языке C, добавить в него ассемблерный модуль с рекурсивной процедурой, вычисляющей $N!$ и сохраняющей его по заданному адресу.

Лабораторная работа 2.6 .Команды обработки строк и однострочные макроопределения.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке C с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Изменить код ассемблерного модуля так, чтобы минимизировать количество обращений к стеку с соблюдением конвенции C.

Лабораторная работа 2.8 .Многострочные макроопределения в ассемблерных модулях.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке C с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Создать консольное приложение на языке C, добавить в него ассемблерный модуль с рекурсивной процедурой, вычисляющей $N!$ и сохраняющей его по заданному адресу.

Лабораторная работа 2.9 .Отладка многострочных макроопределений.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке C с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Сравнить её дизассемблированный код с кодом подобной процедурой, но построенной на языке C.

Лабораторная работа 2.10 Консольное приложение VSC++ с главным ассемблерным модулем.

Цель работы. Знакомство с использованием среды программирования для создания консольных приложений на языке C с ассемблерными модулями. Разработка и отладка рекурсивных ассемблерных подпрограмм для использования в консольных приложениях.

Задание

Изменить код ассемблерного модуля так, чтобы минимизировать количество обращений к стеку с соблюдением конвенции C.