

ПРИЛОЖЕНИЕ

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ИМЕНИ В.Ф. УТКИНА»

КАФЕДРА АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ

ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

по дисциплине

Б1.О.31 «Инструментальные средства информационных систем»

Направление подготовки

09.03.02 Информационные системы и технологии

ОПОП бакалавриата

«Информационные системы и технологии»

Квалификация (степень) выпускника – бакалавр

Формы обучения – очная, заочная

Рязань 2022

Оценочные материалы предназначены для контроля знаний обучающихся по дисциплине «Инструментальные средства информационных систем» и представляют собой фонд оценочных средств, образованный совокупностью учебно-методических материалов (контрольных заданий, описаний лабораторных работ), предназначенных для оценки качества освоения обучающимися данной дисциплины как части основной профессиональной образовательной программы.

Цель – оценить соответствие знаний, умений и уровня приобретенных компетенций обучающихся целям и требованиям основной образовательной программы в ходе проведения учебного процесса.

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

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

Текущий контроль успеваемости проводится с целью определения степени усвоения учебного материала, своевременного выявления и устранения недостатков в подготовке обучающихся и принятия необходимых мер по совершенствованию методики преподавания учебной дисциплины, организации работы обучающихся в ходе учебных занятий и проведения, в случае необходимости, индивидуальных консультаций. К контролю текущей успеваемости относятся проверка знаний, умений и навыков, приобретённых обучающимися на лабораторных работах и практических занятиях.

Промежуточная аттестация студентов по данной дисциплине проводится на основании результатов защиты лабораторных работ. При выполнении лабораторных работ применяется система оценки «зачтено – не зачтено». Количество лабораторных работ по дисциплине определено утвержденным учебным графиком.

По итогам курса студенты сдают в конце семестра обучения экзамен. Форма проведения экзамена – устный ответ, по утвержденным экзаменационным билетам, сформулированным с учетом содержания учебной дисциплины. В экзаменационный билет включается два теоретических вопроса по темам дисциплины.

1 Паспорт фонда оценочных средств по дисциплине

ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.

ОПК-2.1. Понимает состояние и тенденции развития современных информационных технологий и программных средства, в том числе отечественного производства.

Знает: назначение инструментальных средств, виды классификаций инструментальных средств, в том числе отечественного производства, применяемых при решении задач профессиональной деятельности.

Умеет: применять инструментальные средства для проектирования информационных систем.

Владеет: инструментальными средствами, применяемыми при проектировании информационных систем.

ОПК-2.2. Использует при решении задач профессиональной деятельности современные информационные технологии и программные средства, в том числе отечественного производства.

Знает: инструментальные средства автоматизированного проектирования информационных систем.

Умеет: выбирать современные инструментальные средства, в том числе отечественного производства, при решении задач профессиональной деятельности.

Владеет: инструментальными средствами автоматизированного проектирования информационных систем.

ОПК-5. Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем.

ОПК-5.1. Производит установку программного обеспечения для информационных и автоматизированных систем.

Знает: способы установки инструментальных средств обработки регулярных выражений.

Умеет: устанавливать инструментальные средства распараллеливания вычислений.

Владеет: средствами установки программного обеспечения.

ОПК-5.2. Производит установку аппаратного обеспечения для информационных и автоматизированных систем.

Знает: основы установки аппаратного обеспечения.

Умеет: выполнять установку аппаратного обеспечения для информационных систем.

Владеет: навыками установки аппаратного обеспечения информационных систем.

ОПК-7. Способен осуществлять выбор платформ и инструментальных программно-аппаратных средств для реализации информационных систем.

ОПК-7.1. Осуществляет выбор платформ для реализации информационных систем.

Знает: основные платформы, технологии и инструментальные средства для реализации информационных систем.

Умеет: выбирать платформы для реализации информационных систем.

Владеет: высокоуровневыми средствами параллельного программирования для многоядерных процессоров.

ОПК-7.2. Осуществляет выбор инструментальных программно-аппаратных средств для реализации информационных систем.

Знает: критерии выбора инструментальных программно-аппаратных средств для реализации информационных систем.

Умеет: осуществлять выбор платформ и инструментальных средств для реализации информационных систем, применять современные технологии реализации информационных систем.

Владеет: инструментальными средствами для проектирования интеллектуальных информационных систем.

№ п/п	Контролируемые разделы дисциплины	Код контролируемой компетенции	Наименование оценочного средства
1	Инструментальные средства как инструмент проектирования информационных систем	ОПК-2.1-З ОПК-2.1-У ОПК-2.1-В	Контрольные вопросы, экзамен
2	Инструментальные средства автоматизированного проектирования информационных систем	ОПК-2.2-З ОПК-2.2-У ОПК-2.2-В	Контрольные вопросы, экзамен
3	Инструментальные средства обработки регулярных выражений	ОПК-5.1-З ОПК-5.1-У ОПК-5.1-В	Отчет о выполнении лабораторной работы № 1, отчет о выполнении практической работы № 1, контрольные вопросы, экзамен
4	Инструментальные средства распараллеливания вычислений в информационных системах	ОПК-5.2-З ОПК-5.2-У ОПК-5.2-В	Отчет о выполнении лабораторной работы № 2, отчет о выполнении практической работы № 2, контрольные вопросы, экзамен
5	Высокоуровневые средства параллельного программирования для	ОПК-7.1-З ОПК-7.1-У	Отчет о выполнении лабораторной работы № 3,

	многоядерных процессоров: библиотеки Microsoft Parallel FX и язык программирования MC#	ОПК-7.1-B	отчет о выполнении практической работы № 3, контрольные вопросы, экзамен
6	Инструментальные средства проектирования интеллектуальных информационных систем	ОПК-7.2-3 ОПК-7.2-У ОПК-7.2-B	Отчет о выполнении лабораторной работы № 4, отчет о выполнении практической работы № 4, контрольные вопросы, экзамен

Критерии оценивания компетенций по результатам защиты лабораторных работ и сдачи экзамена

1. Уровень усвоения материала, предусмотренного программой.
2. Умение анализировать материал, устанавливать причинно-следственные связи.
3. Качество ответов на вопросы: логичность, убежденность, общая эрудиция.

Критерии приема лабораторных работ:

«зачтено» - студент представил полный отчет о лабораторной работе, ориентируется в представленных в работе результатах, осознано и правильно отвечает на контрольные вопросы;

«не зачтено» - студент не имеет отчета о лабораторной работе, в отчете отсутствуют некоторые пункты задания на выполнение работы, при наличии полного отчета студент не ориентируется в представленных результатах и не отвечает на контрольные вопросы.

Критерии выставления оценок при аттестации результатов обучения по дисциплине в виде экзамена:

- на «отлично» оценивается глубокое раскрытие вопросов, поставленных в экзаменационном задании, понимании е смысла поставленных вопросов, полные ответы на смежные вопросы;

- на «хорошо» оценивается полное раскрытие вопросов, поставленных в экзаменационном задании, понимание смысла поставленных вопросов, но недостаточно полные ответы на смежные вопросы;

- на «удовлетворительно» оценивается неполное раскрытие вопросов экзаменационного задания и затруднения при ответах на смежные вопросы;

- на «неудовлетворительно» оценивается слабое и неполное раскрытие вопросов экзаменационного задания, отсутствие осмысленного представления о существе вопросов, отсутствие ответов на дополнительные вопросы.

2 Примеры контрольных заданий для оценивания компетенций

2.1 Типовые теоретические вопросы для экзамена по дисциплине (3)

ОПК-2.1.

1. Понятие, содержание, назначение инструментальных средств.
2. Виды классификаций инструментальных средств.
3. История и перспективы развития инструментальных средств.

ОПК-2.2.

4. Современные CASE-средства как инструмент многочисленных технологий проектирования информационных систем.

5. Классификация CASE -средств.

6. Характеристики CASE-средств.

7. Организация документооборота и эффективное представление структур и данных документов.

ОПК-5.1.

8. Диалекты регулярных выражений: Perl, PRCE (Perl-Compatible Regular Expressions), .NET, Java, JavaScript, Python, Ruby.
9. Литералы регулярных выражений в исходных текстах.
10. Импортирование библиотеки регулярных выражений.
11. Создание объектов регулярных выражений.
12. Установка параметров регулярных выражений.
13. Проверка возможности совпадения в пределах испытываемой строки.
14. Проверка совпадения со всей испытываемой строкой.
15. Извлечение текста совпадения.
16. Определение позиции и длины совпадения.
17. Извлечение части совпавшего текста.
18. Извлечение списка всех совпадений.
19. Обход всех совпадений в цикле.
20. Проверка полученных совпадений в программном коде.
21. Поиск совпадения внутри другого совпадения.
22. Замена всех совпадений.
23. Замена совпадений с повторным использованием частей совпадений.
24. Разбиение строки.
25. Построчный поиск.

ОПК-5.2.

26. Понятие параллельных вычислений.
27. Требования, приводящие к достижению параллелизма.
28. Режимы выполнения независимых частей программы.
29. Примеры параллельных вычислительных систем (суперкомпьютеры, кластеры, персональные мини-кластеры).
30. Классификация вычислительных систем (по способам взаимодействия потоков выполняемых команд и обрабатываемых данных).
31. Характеристика системных платформ для построения кластеров: Microsoft Compute Cluster Server.
32. Показатели эффективности параллельного алгоритма: ускорение, эффективность, стоимость вычислений.
33. Оценка максимально достижимого параллелизма.
34. Анализ масштабируемости параллельных вычислений.
35. Моделирование параллельных программ.
36. Методика разработки параллельных алгоритмов.

ОПК-7.1.

37. Библиотека Parallel Extensions to the .NET Framework.
38. Конструкция Parallel.For.
39. Планирование исполнения процессов: work sharing – централизованное планирование, work stealing – децентрализованное планирование.
40. Конструкция Parallel.Invoke.
41. Программирование с использованием Task Parallel Library (TPL).
42. Класс System.Threading.Tasks.Future.
43. Координирующие структуры данных.
44. PLINQ (Parallel Language-Integrated Query) – параллельный интегрированный язык запросов.
45. Обработка исключений с использованием библиотеки Parallel FX.
46. Реализация конструкций ContinueWhenAll и ContinueWhenAny.
47. Асинхронное выполнение последовательности задач.
48. Ожидание завершения множества задач.
49. Реализация конструкции ParallelWhileNotEmpty.

- 50. Оценка производительности памяти с помощью теста Random Access.
- 51. Высокоуровневый язык параллельного программирования МС#.
- 52. Async- и movable-методы.
- 53. Каналы и обработчики канальных сообщений.
- 54. Синхронизация в языке МС#.
- 55. Примеры программирования на языке МС#.

ОПК-7.2.

- 56. Стандарты разработки интеллектуальных информационных систем.
- 57. Инструментальные средства разработки интеллектуальных и агентно-ориентированных приложений.

Контрольные вопросы используются на этапах промежуточного контроля (защита лабораторных работ) и заключительного контроля (экзамен) уровня достигнутых компетенций по темам. При проведении текущего и промежуточного контроля по темам используются вопросы тестов, реализованных в рамках системы «Образовательный портал кафедры АСУ» – <http://www.rgrty.ru/>

2.2 Типовые тестовые вопросы (З, У, В)

ОПК-2.1.

- 1. Инструментальные программные средства – это ...
 - 1) программы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ;**
 - 2) аппаратное обеспечение;
 - 3) аппаратно-программное обеспечение;
 - 4) алгоритмы, которые используются в ходе разработки, корректировки или развития других прикладных или системных программ;
 - 5) совокупность аппаратного и программного обеспечения информационных систем.
- 2. Интегрированная среда разработки – это ...
 - 1) система программных средств, используемая программистами для разработки программного обеспечения;**
 - 2) система программных средств, используемая программистами для разработки аппаратного обеспечения;
 - 3) текстовые редакторы;
 - 4) компиляторы;
 - 5) интерпретаторы.
- 3. Компилятор – это ...
 - 1) программа или техническое средство, выполняющее трансляцию программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду;**
 - 2) программа или техническое средство, выполняющее трансляцию программы, составленной на исходном языке низкого уровня, в эквивалентную программу на высокоуровневом языке;
 - 3) устройство разработки алгоритма решения задачи;
 - 4) устройство отладки программы;
 - 5) программное средство отладки программы.
- 4. Интерпретатор – это ...
 - 1) программа или аппаратное устройство, выполняющее пооператорный анализ, обработку и выполнение исходной программы или запроса;**

2) программа или техническое средство, выполняющее трансляцию программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду;

3) устройство разработки алгоритма решения задачи;

4) устройство отладки программы;

5) программное средство отладки программы.

5. Компоновщик – это ...

1) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль;

2) программа или техническое средство, выполняющее трансляцию программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду;

3) программа или аппаратное устройство, выполняющее пооператорный анализ, обработку и выполнение исходной программы или запроса;

4) устройство отладки программы;

5) программное средство отладки программы.

6. Синтаксический анализ – это ...

1) процесс сопоставления линейной последовательности лексем языка с его формальной грамматикой;

2) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль;

3) программа или аппаратное устройство, выполняющее пооператорный анализ, обработку и выполнение исходной программы или запроса;

4) программа или техническое средство, выполняющее трансляцию программы, составленной на исходном языке высокого уровня, в эквивалентную программу на низкоуровневом языке, близком машинному коду;

5) система программных средств, используемая программистами для разработки программного обеспечения.

7. Отладчик – это ...

1) компьютерная программа, предназначенная для поиска ошибок в других программах, ядрах операционных систем, SQL-запросах и других видах кода;

2) процесс сопоставления линейной последовательности лексем языка с его формальной грамматикой;

3) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль;

4) программа или аппаратное устройство, выполняющее пооператорный анализ, обработку и выполнение исходной программы или запроса;

5) система программных средств, используемая программистами для разработки программного обеспечения.

8. Профилирование – это ...

1) сбор характеристик работы программы, таких как время выполнения отдельных фрагментов, число верно предсказанных условных переходов, число кэш промахов;

2) компьютерная программа, предназначенная для поиска ошибок в других программах, ядрах операционных систем, SQL-запросах и других видах кода;

3) процесс сопоставления линейной последовательности лексем языка с его формальной грамматикой;

4) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль;

5) программа или аппаратное устройство, выполняющее пооператорный анализ, обработку и выполнение исходной программы или запроса.

9. Генератор документации – это ...

1) программа или пакет программ, позволяющая получать документацию, предназначенную для программистов и/или для конечных пользователей системы, по особым образом комментированному исходному коду и по исполняемым модулям;

2) сбор характеристик работы программы, таких как время выполнения отдельных фрагментов, число верно предсказанных условных переходов, число кэш промахов ;

3) компьютерная программа, предназначенная для поиска ошибок в других программах, ядрах операционных систем, SQL-запросах и других видах кода;

4) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль;

5) программа, которая используется в ходе разработки, корректировки или развития других прикладных или системных программ.

10. Система управления версиями – это ...

1) программное обеспечение для облегчения работы с изменяющейся информацией;

2) сбор характеристик работы программы, таких как время выполнения отдельных фрагментов, число верно предсказанных условных переходов, число кэш промахов;

3) компьютерная программа, предназначенная для поиска ошибок в других программах, ядрах операционных систем, SQL-запросах и других видах кода;

4) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль;

5) программа, которая принимает на вход один или несколько объектных модулей и собирает по ним исполнимый модуль.

11. Машинные языки – это ...

1) языки программирования, воспринимаемые аппаратной частью компьютера;

2) языки программирования, которые отражают структуру конкретного типа компьютера;

3) языки программирования, не зависящие от архитектуры компьютера, служащие для отражения структуры алгоритма;

4) языки программирования, где имеется возможность описания программы как совокупности процедур, подпрограмм;

5) языки программирования, предназначенные для решения задач определенного класса.

12. Машинно-ориентированные языки – это ...

1) языки программирования, которые отражают структуру конкретного типа компьютера;

2) языки программирования, воспринимаемые аппаратной частью компьютера;

3) языки программирования, не зависящие от архитектуры компьютера, служащие для отражения структуры алгоритма;

4) языки программирования, где имеется возможность описания программы как совокупности процедур, подпрограмм;

5) языки программирования, предназначенные для решения задач определенного класса.

13. Алгоритмические языки – это ...

1) языки программирования, не зависящие от архитектуры компьютера, служащие для отражения структуры алгоритма;

2) языки программирования, которые отражают структуру конкретного типа компьютера;

3) языки программирования, воспринимаемые аппаратной частью компьютера;

4) языки программирования, предназначенные для решения задач определенного класса;

5) ассемблеры.

14. Процедурно-ориентированные языки – это ...

1) языки программирования, где имеется возможность описания программы как совокупности процедур, подпрограмм;

2) языки программирования, не зависящие от архитектуры компьютера, служащие для отражения структуры алгоритма;

3) языки программирования, воспринимаемые аппаратной частью компьютера;

4) языки программирования, предназначенные для решения задач определенного класса;

5) ассемблеры.

15. Проблемно-ориентированные языки – это ...

1) языки программирования, предназначенные для решения задач определенного класса;

2) языки программирования, не зависящие от архитектуры компьютера, служащие для отражения структуры алгоритма;

3) языки программирования, воспринимаемые аппаратной частью компьютера;

4) языки программирования, где имеется возможность описания программы как совокупности процедур, подпрограмм;

5)) ассемблеры.

ОПК-2.2.

16. Базовые программные средства включают ...

1) операционные системы;

2) языки программирования;

3) программные среды;

4) системы управления базами данных;

5) компьютерные игры.

17. Инструментальные средства CASE – это ...

1) специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования информационных систем;

2) текстовые процессоры;

3) электронные таблицы;

4) браузеры;

5) почтовые клиенты.

18. Лексический анализатор ...

1) просматривает литеры исходной программы слева направо и строит символы программы – целые числа, идентификаторы, служебные слова;

2) выполняет компиляцию программы;

3) выполняет трансляцию программы;

4) выполняет отладку программы;

5) ищет вирусы.

19. Инструментальная система поддержки проекта – это ...

1) открытая система, способная поддерживать разработку программного средства на разных языках программирования после соответствующего ее расширения программными инструментами, ориентированными на выбранный язык;

2) программа для просмотра литер исходной программы слева направо и построения символов программы;

3) специальная программа, которая поддерживают одну или несколько методологий анализа и проектирования информационных систем;

4) язык программирования, предназначенный для решения задач определенного класса;

5) язык программирования, в котором имеется возможность описания программы как совокупности процедур, подпрограмм.

20. Программная среда – это ...

1) интегрированная совокупность технических и программных средств, при помощи которых осуществляется разработка программ;

2) открытая система, способная поддерживать разработку программного средства на разных языках программирования после соответствующего ее расширения программными инструментами, ориентированными на выбранный язык;

3) специальные программы, которые поддерживают одну или несколько методологий анализа и проектирования информационных систем;

4) язык программирования, в котором возможность описания программы как совокупности процедур, подпрограмм;

5) язык программирования, не зависящий от архитектуры компьютера, служащий для отражения структуры алгоритма.

ОПК-5.1.

21. Символьный класс `\w` в регулярных выражениях задает соответствие:

1) цифре, букве латинского алфавита или знаку нижнего подчеркивания;

2) цифре, букве русского алфавита или знаку нижнего подчеркивания;

3) цифре, букве знаку нижнего подчеркивания;

4) цифре, букве русского алфавита или пробельному символу;

5) любому символу, кроме цифр, букв и знака нижнего подчеркивания.

22. Символьный класс `\s` в регулярных выражениях задает соответствие:

1) пробельному символу;

2) цифре;

3) букве;

4) букве русского алфавита;

5) букве латинского алфавита.

23. Символ повторения `*` в регулярных выражениях означает соответствие ...

1) предыдущему элементу 0 или более раз;

2) предыдущему элементу один или более раз;

3) предыдущему элементу ноль или один раз;

4) предыдущему элементу один раз;

5) предыдущему элементу два раза.

24. Символ повторения `+` в регулярных выражениях означает соответствие ...

1) предыдущему элементу один или более раз;

2) предыдущему элементу 0 или более раз;

3) предыдущему элементу ноль или один раз;

4) предыдущему элементу один раз;

5) предыдущему элементу два раза.

25. Символ повторения `?` в регулярных выражениях означает соответствие ...

1) предыдущему элементу ноль или один раз;

2) предыдущему элементу один или более раз;

3) предыдущему элементу 0 или более раз;

4) предыдущему элементу один раз;

5) предыдущему элементу два раза.

ОПК-5.2.

26. Класс *Thread* определяет ...

1) ряд методов и свойств, которые позволяют управлять потоком и получать информацию о нем;

2) ряд методов и свойств, которые позволяют управлять подпрограммой и получать информацию о ней;

3) ряд методов и свойств, которые позволяют управлять обработчиком прерываний и получать информацию о нем;

4) ряд методов и свойств, которые позволяют управлять событием и получать информацию о нем;

5) ряд методов и свойств, которые позволяют управлять потоком и получать информацию о нем.

27. Статическое свойство *CurrentContext* класса *Thread* ...

1) позволяет получить контекст, в котором выполняется поток;

2) возвращает ссылку на выполняемый поток;

3) указывает, является ли поток фоновым;

4) содержит имя потока;

5) хранит приоритет потока.

28. Статическое свойство *CurrentThread* класса *Thread* ...

1) возвращает ссылку на выполняемый поток;

2) позволяет получить контекст, в котором выполняется поток;

3) указывает, является ли поток фоновым;

4) содержит имя потока;

5) хранит приоритет потока.

29. Свойство *IsAlive* класса *Thread* ...

1) указывает, работает ли поток в текущий момент;

2) указывает, является ли поток фоновым;

3) возвращает ссылку на выполняемый поток;

4) содержит имя потока;

5) хранит приоритет потока.

30. Свойство *IsBackground* класса *Thread* ...

1) указывает, является ли поток фоновым;

2) указывает, работает ли поток в текущий момент;

3) возвращает ссылку на выполняемый поток;

4) содержит имя потока;

5) хранит приоритет потока.

ОПК-7.1.

31. Конструкция *Parallel.For* является параллельным аналогом ...

1) цикла for;

2) цикла while;

3) цикла do while;

4) цикла repeat until;

5) цикла foreach.

32. Конструкция *Parallel.ForEach* является параллельным аналогом ...

1) цикла foreach;

2) цикла for;

3) цикла while;

4) цикла do while;

5) цикла repeat until.

33. Статический метод *Parallel.Invoke* позволяет ...

1) распараллелить исполнение блоков операторов;

- 2) распараллелить обработку прерываний;
- 3) выполнить цикл for;
- 4) выполнить цикл while;
- 5) выполнить цикл do while.

34. Подход *Work Sharing* – это ...

1) централизованное планирование, при котором планировщик контролирует единый пул задач в системе и назначает задачи процессорам;

- 2) децентрализованное планирование, при котором единый планировщик отсутствует, а процессоры сами выбирают какие задачи им исполняют;
- 3) параллельное выполнение цикла for;
- 4) параллельное выполнение цикла while;
- 5) параллельное выполнение цикла do while.

35. Подход *Work Stealing* – это ...

1) децентрализованное планирование, при котором единый планировщик отсутствует, а процессоры сами выбирают какие задачи им исполняют;

- 2) децентрализованное планирование, при котором единый планировщик отсутствует, а процессоры сами выбирают какие задачи им исполняют;
- 3) параллельное выполнение цикла for;
- 4) параллельное выполнение цикла while;
- 5) параллельное выполнение цикла do while.

ОПК-7.2.

36. Мультиагентная система состоит из ...

1) множества организационных единиц, в котором выделяются: подмножество агентов, манипулирующих подмножеством объектов;

- 2) множества задач, решаемых агентами в рамках выполнения общей задачи;
- 3) среды, т.е. некоторого пространства, в котором существуют агенты и объекты;
- 4) множества отношений между агентами;
- 5) множества формул.

37. Адаптивный планировщик мультиагентной системы ...

1) обрабатывает поток входящих событий;

- 2) обрабатывает поток исходящих событий;
- 3) вызывает подпрограммы;
- 4) вызывает методы;
- 5) обрабатывает прерывания.

38. Конструктор сцены мультиагентной системы ...

1) позволяет редактировать начальную конфигурацию сети и определить все параметры ресурсов компании;

- 2) позволяет редактировать конечную конфигурацию сети и определить все параметры ресурсов компании;
- 3) обрабатывает поток входящих событий;
- 4) обрабатывает поток исходящих событий;
- 5) вызывает подпрограммы.

39. Редактор онтологии мультиагентной системы позволяет ...

1) ввести и изменить общую онтологию компании, описывающую модель знаний предметной области;

- 2) редактировать начальную конфигурацию сети и определить все параметры ресурсов компании;
- 3) обрабатывать поток входящих событий;
- 4) обрабатывать поток исходящих событий;

5) вызывать подпрограммы.

40. Эволюционный дизайн мультиагентной системы – это ...

1) модуль, вырабатывающий предложения по улучшению конфигурации сети в части увеличения или уменьшения определенного числа ресурсов, изменению географии ресурсов;

2) модуль ввода и изменения общей онтологии компании, описывающей модель знаний предметной области;

3) модуль редактирования начальной конфигурации сети и определения всех параметров ресурсов компании;

4) модуль обработки потока входящих событий;

5) модуль вызова подпрограмм.

2.3. Контрольная работа (для заочной формы обучения) (З, У, В)

Пример задания на контрольную работу

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
имени В.Ф. Уткина

Кафедра автоматизированных систем управления

ЗАДАНИЕ НА КОНТРОЛЬНУЮ РАБОТУ

по дисциплине

«Инструментальные средства информационных систем»

Студент(ка) _____ группы _____

Тема: Инструментальные средства обработки регулярных выражений

Вопросы.

1. Понятие регулярных выражений.
2. Класс Regex.
3. Метод IsMatch().
4. Метод Match().
5. Метод Matches().
6. Метод Replace().
7. Метод Split().
8. Классы символов регулярных выражений.
9. Символы повторения.
10. Символы привязки.
11. Символы выбора.
12. Служебные символы.
13. Перечисление RegexOptions.
14. Сопоставление с одним символом из нескольких.
15. Сопоставление в начале и/или в конце строки.
16. Сопоставление с целыми словами.
17. Сопоставление с одной из нескольких альтернатив.
18. Группы и сохранение части совпадения.
19. Повторный поиск соответствия с ранее совпавшим текстом.
20. Сохранение и именованные части совпадения.

Задание принял к исполнению _____ ” _____ 20__ г.

Консультант

Челебаев С.В.

3 Формы контроля

3.1 Формы текущего контроля

Текущий контроль по дисциплине проводится в виде тестовых опросов (в том числе с использованием дистанционных средств контроля на сайте кафедры www.rgrty.ru) по отдельным темам дисциплины, проверки заданий, выполняемых самостоятельно при подготовке к практическим занятиям.

3.2 Формы промежуточного контроля

Форма промежуточного контроля по дисциплине – защита лабораторных работ.

Защита контрольной работы (для заочной формы обучения).

3.3 Формы заключительного контроля

Форма заключительного контроля по дисциплине – экзамен.

3.4 Критерий допуска к экзамену

К экзамену допускаются студенты, защитившие ко дню проведения экзамена по расписанию экзаменационной сессии все лабораторные работы.

Студенты, не защитившие ко дню проведения экзамена по расписанию экзаменационной сессии хотя бы одну лабораторную работу, на экзамене получают неудовлетворительную оценку. Решение о повторном экзамене и сроках проведения экзамена принимает деканат после ликвидации студентом имеющейся задолженности по лабораторным работам.

Составил

доцент кафедры АСУ

к.т.н., доцент

Челебаев С.В.

Заведующий кафедрой АСУ

к.т.н., доцент

Холопов С.И.