



Государственное бюджетное образовательное учреждение высшего образования
Московской области

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ
Проректор по учебной работе
Е.К. Самаров
« 25 » 2021г.



***ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН***

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«Прикладные интернет-технологии»

Направление подготовки: 01.03.02. Прикладная математика и информатика

Профиль: Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавр

Форма обучения: очная




Королев
2021

Автор: Вилисов В.Я. Рабочая программа дисциплины: Прикладные интернет-технологии. – Королев МО: МГОТУ, 2021

Рецензент: д.ф.-м.н. проф. Самаров К.Л.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования по направлению подготовки бакалавров 01.03.02 «Прикладная математика и информатика» и Учебного плана, утвержденного Ученым советом МГОТУ. Протокол № 13 от «22» июня 2021 года.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Бугай И.В. к.т.н., доцент 	<i>Бугай И.В.</i> к.т.н., доцент 	<i>Бугай И.В.</i> к.т.н., доцент 	
Год утверждения (переутверждения)	2021	<i>2022</i>	<i>2023</i>	
Номер и дата протокола заседания кафедры	<i>№10 от</i> <i>28.05.21</i>	<i>№11 от</i> <i>10.06.22</i>	<i>№9 от</i> <i>25.04.23</i>	

Рабочая программа согласована:

Руководитель ОПОП ВО  к.т.н., доц. И.В. Бугай

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2021	<i>2022</i>	<i>2023</i>	
Номер и дата протокола заседания УМС	<i>№7 от</i> <i>15.06.21</i>	<i>№5 от</i> <i>21.06.22</i>	<i>№6 от</i> <i>16.05.23</i>	

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП ВО

Целью изучения дисциплины является:

1. формирование способности к восприятию, обобщению и анализу информации, необходимой для оценки вариантов, обеспечения и поддержки принятия эффективных конструкторских и управленческих решений;
2. освоение необходимого инструментария, применяемого при оценке вариантов, выборе эффективных конструкторских и управленческих решений;
3. формирование готовности применять методы математического анализа и моделирования сложных систем и процессов в профессиональной деятельности.

В процессе обучения студент приобретает и совершенствует следующие компетенции:

общефессиональные компетенции (ОПК):

- Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности (ОПК-3);

профессиональные компетенции (ПК):

- Способность использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ (ПК-3);
- Способность создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий, программирования и компьютерной техники (ПК-5).

Основными **задачами** дисциплины являются:

1. Освоение студентами теоретических методов дисциплины, применяемых при оценке вариантов и выборе эффективных конструкторских и управленческих решений;
2. Получение студентами умений и навыков, применяемых для решения практических задач оценки вариантов и выбора эффективных конструкторских и управленческих решений.

После завершения освоения данной дисциплины студент должен:

Знать:

- основные элементы и особенности скриптовых (сценарных) языков программирования JavaScript, Perl, PHP, Python. СУБД MySQL
- методы, реализованные в языках программирования Ruby, VBScript

Уметь:

- пользоваться всеми необходимыми методами скриптовых (сценарных) языков программирования JavaScript, Perl, PHP, Python. СУБД MySQL при решении профессиональных задач;

- методами, реализованными в языках программирования Ruby, VBScript при решении профессиональных задач;

Владеть:

- методами скриптовых (сценарных) языков программирования JavaScript, Perl, PHP, Python. СУБД MySQL для успешного решения профессиональных задач и содержательной интерпретации полученных результатов
- методами, реализованными в языках программирования Ruby, VBScript для содержательной интерпретации полученных результатов.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Прикладные интернет-технологии» относится к дисциплинам по выбору части, формируемой участниками образовательных отношений, основной профессиональной образовательной программы подготовки бакалавров по направлению 01.03.02. «Прикладная математика и информатика».

Дисциплина базируется на ранее полученных знаниях по таким дисциплинам как «Языки высокого уровня», «Технологии и среды программирования», «Операционные системы, среды и оболочки» и компетенциях: ОПК-2, ОПК-4, ПК-2, ПК-3, ПК-5.

Знания и компетенции, полученные при освоении дисциплины, являются базовыми для выполнения выпускной квалификационной работы бакалавра.

3. Объем дисциплины и виды учебной работы

Общая трудоемкость дисциплины для студентов очной формы составляет 11 зачетных единиц, 396 часов.

Таблица 1

Виды занятий	Всего часов	Семестр пятый	Семестр шестой	Семестр седьмой
ОЧНАЯ ФОРМА ОБУЧЕНИЯ				
Общая трудоемкость	396	108	180	108
Аудиторные занятия	160	48	48	64
Лекции (Л)	64	16	16	32
Практические занятия (ПЗ)	96	32	32	32
Лабораторные работы (ЛР)	-	-	-	-
Самостоятельная работа	236	60	132	44
Курсовые, расчетно-графические работы	+	-	-	+
Расчетно-графические работы	-	-	-	-
Контрольная работа, домашнее задание				
Текущий контроль знаний	Тест	Тест	Тест	Тест
Вид итогового контроля	Экзамен	+	+	+
ЗАОЧНАЯ ФОРМА НЕ ПРЕДУСМОТРЕНА УЧЕБНЫМ ПЛАНОМ				

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, час.	Практические занятия, час	Лабораторные занятия, час	Занятия в интерактивной форме, час	Код компетенций
5 семестр					
Тема 1. Скриптовые (сценарные) языки программирования JavaScript, Perl	16	32	-	5	ОПК-3
6 семестр					
Тема 2. Языки программирования PHP, Python. СУБД MySQL	16	32	-	5	ПК-3
7 семестр					
Тема 3. Языки программирования Ruby, VBScript	32	32	-	10	ПК-5
Итого:	64	96	-	20	

4.2. Содержание тем дисциплины

Тема 1. Скриптовые (сценарные) языки программирования JavaScript, Perl.

- 1.1.** Основы языка. Назначение и применение, понятие объектной модели языка, иерархия классов. Структура программы, типы данных и переменные, операторы языка.
- 1.2.** Программирование свойств окна браузера. Программирование форм: текст в полях ввода, списки и выпадающие меню, кнопки, картинки, обмен данными.
- 1.3.** Программирование графики: графика и таблицы, графика и обработка событий, вертикальные и горизонтальные меню.
- 1.4.** Программирование гипертекстовых переходов.
- 1.5.** Интерфейс Web-приложений. Протокол: общая структура сообщений, методы доступа. Виды интерфейса пользователя. Спецификация CGI, понятие CGI – скрипта. Типы запросов, механизмы приема данных скриптом, механизм генерации отклика скриптом. Библиотека. Режимы буферизации. Коллективные взаимодействия процессов. Управление группами и коммуникаторами.

Тема 2. Языки программирования PHP, Python. СУБД MySQL.

- 2.1.** Основы языка. Назначение и применение, понятие объектной модели языка, иерархия классов. Структура программы, типы данных и переменные, операторы языка.
- 2.2.** Программирование свойств окна браузера. Программирование форм: текст в полях ввода, списки и выпадающие меню, кнопки, картинки, обмен данными.

- 2.3. Программирование графики: графика и таблицы, графика и обработка событий, вертикальные и горизонтальные меню.
- 2.4. Программирование гипертекстовых переходов.
- 2.5. Интерфейс Web-приложений. Протокол: общая структура сообщений, методы доступа. Виды интерфейса пользователя. Спецификация CGI, понятие CGI – скрипта. Типы запросов, механизмы приема данных скриптом, механизм генерации отклика скриптом. Библиотека. Режимы буферизации. Коллективные взаимодействия процессов. Управление группами и коммутаторами.
- 2.6. Структура данных и особенности использования СУБД MySQL в интернет-приложениях.

Тема 3. Языки программирования Ruby, VBScript.

- 3.1. Основы языка. Назначение и применение, понятие объектной модели языка, иерархия классов. Структура программы, типы данных и переменные, операторы языка.
- 3.2. Программирование свойств окна браузера. Программирование форм: текст в полях ввода, списки и выпадающие меню, кнопки, картинки, обмен данными.
- 3.3. Программирование графики: графика и таблицы, графика и обработка событий, вертикальные и горизонтальные меню.
- 3.4. Программирование гипертекстовых переходов.
- 3.5. Интерфейс Web-приложений. Протокол: общая структура сообщений, методы доступа. Виды интерфейса пользователя. Спецификация CGI, понятие CGI – скрипта. Типы запросов, механизмы приема данных скриптом, механизм генерации отклика скриптом. Библиотека. Режимы буферизации. Коллективные взаимодействия процессов. Управление группами и коммутаторами.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

- 1. Методические указания для обучающихся по освоению дисциплины.

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине «Прикладные интернет-технологии» приведен в Приложении 1 к настоящей рабочей программе.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Интернет-технологии: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2015. - 184 с - [электронный ресурс] // <http://znanium.com/catalog.php?bookinfo=488074>.
2. Технология разработки интернет ресурсов: курс лекций: [16+] / авт.-сост. И.А. Журавлёва; Министерство науки и высшего образования Российской Федерации, Северо-Кавказский федеральный университет. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2018. – 171 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=562579>.

Дополнительная литература:

1. Хныкина, А.Г. Информационные технологии: учебное пособие / А.Г. Хныкина, Т.В. Минкина; Министерство образования и науки Российской Федерации, Северо-Кавказский федеральный университет. - Ставрополь: СКФУ, 2017. - 126 с.: схем., ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=494703>
2. Технологии создания интеллектуальных устройств, подключенных к интернет: учебное пособие / А.В. Приемышев, В.Н. Крутов, В.А. Тряель, О.А. Коршакова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2018. — 100 с. — ISBN 978-5-8114-2310-1. — Текст: электронный // Электронно-библиотечная система «Лань»: [сайт]. — URL: <https://e.lanbook.com/book/103911>.

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

1. <http://biblioclub.ru/> - Университетская библиотека ONLINE.
2. <http://www.rucont.ru/> - Электронная библиотека Руконт.
3. <http://www.znanium.com/> - Электронная библиотека издательства Инфра-М.
4. <http://e.lanbook.com/> - Электронная библиотека Лань.
5. <http://www.olap.ru/> - Сайт OLAP-технологий.
6. <http://www.anylogic.ru/> - Сайт системы имитационного моделирования AnyLogic, содержащий методические и учебные материалы по имитационному моделированию в свободном доступе для различных прикладных сфер.
7. <http://www.runthemodel.com/> - Библиотека имитационных моделей, находящихся в свободном доступе, выполненных в среде AnyLogic.
8. <https://ru.atlassian.com/software/jira/> - Система планирования работ проекта ПО.
9. <https://www.jetbrains.com/youtrack/> - Система планирования работ проекта ПО.
10. <https://ru.atlassian.com/software/confluence> - Система хранения проектной документации.
11. <https://git-scm.com/> - Распределенная система контроля версий ПО.
12. <https://ru.wikipedia.org/wiki/Subversion> - Централизованная система контроля версий ПО.

13. <https://jenkins-ci.org/> - Система непрерывной интеграции проекта ПО.

14. <https://www.docker.com/> - Система виртуализации окружения.

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в Приложении 2 к настоящей рабочей программе.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения: *MS Office, Mathcad, AnyLogic, Java, JIRA Software, Confluence, SubVersion, Git, Docker.*

Информационные справочные системы: *Электронные ресурсы образовательной среды Университета*

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Лекционные занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран), доска, комплект маркеров;
- комплект электронных презентаций;
- рабочие места студентов, оснащенные компьютерами с доступом (управляемым с компьютера преподавателя) в Интернет к почтовым серверам и к адресам, приведенным в разделе 8 и к общей сетевой папке группы.

Практические занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран), доска, комплект маркеров;
- комплект электронных презентаций;
- рабочие места студентов, оснащенные компьютерами с доступом (управляемым с компьютера преподавателя) в Интернет к почтовым серверам, к адресам, приведенным в разделе 8 и к общей сетевой папке группы.

***ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН***

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ**

«Прикладные интернет-технологии»

(Приложение 1 к рабочей программе)

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавр

Форма обучения: очная

Королев
2021

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся должен:		
				Знать	уметь	владеть
1.	ОПК-3	Способен применять и модифицировать математические модели для решения задач в области профессиональной деятельности	Тема 1. Скриптовые языки программирования JavaScript, Perl	основные элементы и особенности скриптовых (сценарных) языков программирования JavaScript, Perl, PHP, Python. СУБД MySQL	пользоваться всеми необходимыми методами скриптовых (сценарных) языков программирования JavaScript, Perl, PHP, Python. СУБД MySQL при решении профессиональных задач	методами скриптовых (сценарных) языков программирования JavaScript, Perl, PHP, Python. СУБД MySQL для успешного решения профессиональных задач и содержательной интерпретации полученных результатов
2.	ПК-3	Способность использовать современные методы разработки и реализации конкретных алгоритмов математических моделей на базе языков программирования и пакетов прикладных программ	Тема 2. Языки программирования PHP, Python. СУБД MySQL	методы, реализованные в языках программирования Ruby, VBScript	методами, реализованным и в языках программирования Ruby, VBScript при решении профессиональных задач;	методами, реализованными в языках программирования Ruby, VBScript для содержательной интерпретации полученных результатов
3.	ПК-5	Способность создавать и исследовать новые математические модели в естественных науках, промышленности и бизнесе с учетом возможностей современных информационных технологий, программирования и компьютерной техники.	Тема 3. Языки программирования Ruby, VBScript	методы, реализованные в языках программирования Ruby, VBScript	методами, реализованным и в языках программирования Ruby, VBScript при решении профессиональных задач;	методами, реализованными в языках программирования Ruby, VBScript для содержательной интерпретации полученных результатов

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Инструменты, оценивающие сформированность компетенции	Этапы и показатель оценивания компетенции	Шкала и критерии оценки
ОПК-3, ПК-3,5	Письменное задание	<p>А) полностью сформирована 5 баллов</p> <p>В) частично сформирована 3-4 балла</p> <p>С) сформировано менее 30% 1-2 балла</p> <p>Д) не сформирована 0 балла</p>	<p>Проводится в письменной форме для всех видов нозологий</p> <ol style="list-style-type: none"> 1. Выбор оптимального метода решения задачи (1 балл) 2. Умение применить выбранный метод (1 балл) 3. Логический ход решения правильный, но имеются арифметически в расчетах (1 балл) 4. Решение задачи и получение правильного результата (2 балла) 5. Задача не решена вообще (0 баллов) <p>Максимальная оценка - 5 баллов.</p> <p>Время, отведенное на процедуру – до 40 мин. При необходимости время может быть увеличено.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал</p>
ОПК-3, ПК-3,5	Курсовая работа	<p>А) полностью сформирована -5 баллов</p> <p>Б) частично сформирована 3-4 балла</p> <p>В) не сформирована 2 и менее баллов</p>	<p>Проводится в письменной форме</p> <ol style="list-style-type: none"> 1. Оформление в соответствии с требованиями (1 балл); 2. Соответствует методическим указаниям в части структуры (1 балл); 3. Содержание курсовой работы соответствует заявленной тематике (1 балл); 4. Постановленные цели и задачи достигнуты (1 балл); 5. Качественный и количественный состав использованных источников (1 балл). <p>Максимальная оценка – 5 баллов.</p> <p>Результаты оценочной процедуры представляются обучающимся в срок не позднее 1 недели после проведения процедуры – для текущего контроля. Оценка проставляется в электронный журнал</p>

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

3.1 Примерная тематика письменных заданий, выполняемых с применением программных средств специального и общего назначения на компьютерах:

5 семестр

1. **FizzBuzz.** Напишите программу, которая выводит через `console.log` все числа от 1 до 100, с двумя исключениями. Для чисел, нацело делящихся на 3, она должна выводить 'Fizz', а для чисел, делящихся на 5 (но не на 3) – 'Buzz'. Когда сумеете – исправьте её так, чтобы она выводила «FizzBuzz» для всех чисел, которые делятся и на 3 и на 5.
2. **Считаем бобы.** Символ номер N строки можно получить, добавив к ней `.charAt(N)` ("строка".`charAt(5)`) – схожим образом с получением длины строки при помощи `.length`. Возвращаемое значение будет строковым, состоящим из одного символа (к примеру, "к"). У первого символа строки позиция 0, что означает, что у последнего символа позиция будет `string.length - 1`. Другими словами, у строки из двух символов длина 2, а позиции её символов будут 0 и 1. Напишите функцию `countBs`, которая принимает строку в качестве аргумента, и возвращает количество символов "B", содержащихся в строке. Затем напишите функцию `countChar`, которая работает примерно как `countBs`, только принимает второй параметр — символ, который мы будем искать в строке (вместо того, чтобы просто считать количество символов "B"). Для этого переделайте функцию `countBs`.
3. **Кавычки в тексте** Допустим, вы написали рассказ, и везде для обозначения диалогов использовали одинарные кавычки. Теперь вы хотите заменить кавычки диалогов на двойные, и оставить одинарные в сокращениях слов типа `aren't`. Придумайте шаблон, различающий два этих использования кавычек, и напишите вызов метода `replace`, который производит замену.
4. **Круговые зависимости** Запутанная тема в управлении зависимостями – круговые зависимости, когда модуль A зависит от Б, а Б зависит от А. Многие системы модулей это просто запрещают. Модули `CommonJS` допускают ограниченный вариант: это работает, пока модули не заменяют объект `exports`, существующий по-умолчанию, другим значением, и начинают использовать интерфейсы друг друга только после окончания загрузки. Можете ли вы придумать способ, который позволил бы воплотить систему поддержки таких зависимостей? Посмотрите на определение `require` и подумайте, что нужно сделать этой функции для этого.
5. Дано натуральное число (более четырех знаков):
 - найти произведение цифр числа, меньших 7;
 - верно ли, что в числе нет цифры А?
6. Дано натуральное число (более четырех знаков):

- найти произведение нечетных цифр числа;
 - верно ли, что в числе нет цифр, кратных 3?
7. Найти все натуральные числа из промежутка от 1 до 200, у которых сумма цифр равна S. Найти произведение нечетных делителей числа 54, кратных 3.
 8. Дан одномерный массив целых чисел, состоящий из 9 элементов. Требуется:
 - найти сумму элементов, имеющих нечетные индексы, и количество элементов, значения которых меньше числа b и при этом кратны 5;
 - умножить все элементы, кратные 3, на третий элемент массива, кроме него самого (в случае, если он кратен числу 3).
 9. Дан двумерный массив A размером 4 x 5, заполненный целыми числами. Требуется сформировать одномерный массив B, каждый элемент которого равен количеству отрицательных элементов меньших 5 по модулю, соответствующего столбца массива A.
 10. Вывести различные символы двух строк (т.е. такие, какие есть только в одном из них).
 11. Дана строка с текстом, заканчивающимся точкой. Требуется:
 - заменить во всех словах первую букву заглавной.
 12. Сделать страницу из двух равных горизонтальных фреймов, содержащих какие-либо тексты. Каждые 2 секунды менять друг с другом первые слова из этих текстов, используя методы позиционирования объекта TextRange.
 13. Каждую секунду через любой динамический фильтр меняются цвет фона и названия времен года в центре экрана: "зима", "весна", "лето", "осень".
 - new Array('snow', 'lime', 'pink', 'yellow'); - цвета фона
 - new Array('blue', 'green', 'red', 'orange'); - цвета текста
 14. Разноцветные шарики диаметром 50 пикселей каждую секунду возникают в разных местах экрана. Конец по щелчку мышкой. Используйте цвета из массива new Array("#FF0000", "#00FF00", "#0000FF", "#FFFF00", "#00FFFF")
 15. Создать регулярное выражение, вытаскивающее дату и время. А также комментарий, если он есть. Результат вывести на экран в отсортированном по дате (в убывающем) порядке. Должно получиться только одно регулярное выражение, разбирающее все 3 варианта.
 16. Напишите программу, которая получает результат команды date и вычисляет текущий день недели. Если день недели ≈ рабочий день, выводить get to work, в противном случае выводить go play.
 17. Напишите программу, которая получает все реальные имена пользователей из файла /etc/passwd, а затем трансформирует результат команды who, заменяя регистрационное имя (первая колонка) реальным именем. Попробуйте выполнить эту задачу с использованием команды who как в обратных кавычках, так и открытых как канал. Что легче?
 18. Модифицируйте предыдущую программу так, чтобы ее результат автоматически поступал на принтер. (Если у вас нет доступа к принтеру, то, вероятно, вы можете послать самому себе сообщение электронной почты.)
 19. Предположим, функция mkdir перестала работать. Напишите подпрограмму, которая не использует mkdir, а вызывает /bin/mkdir с

помощью функции `system`. (Убедитесь в том, что она работает с каталогами, в именах которых шесть пробел.)

20. Расширьте программу из предыдущего упражнения так, чтобы в ней устанавливались права доступа (с помощью функции `chmod`).

21. Создать сайт для выбора книги. Книга должна выбираться по любой комбинации из следующих признаков:

- название,
- автор,
- тема.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о книгах должны храниться в базе данных. Описание одной книги должно иметь не менее пяти признаков. В базе данных должно содержаться описание не менее двадцати книг. Вход на сайт должен быть защищён паролем.

22. Создать сайт для выбора товаров со склада. Товар должен выбираться по любой комбинации из следующих признаков:

- название.
- фирма,
- цена.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о товарах должны храниться в базе данных. Описание одного товара должно иметь не менее пяти признаков. В базе данных должно содержаться описание не менее двадцати товаров. Вход на сайт должен быть защищён паролем.

23. Создать сайт для контроля успеваемости учеников в школе. Данные об успеваемости должны выбираться по любой комбинации из следующих признаков:

- ФИО ученика,
- класс,
- предмет.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные об успеваемости учеников должны храниться в базе данных. Описание одного ученика должно иметь не менее пяти признаков. В базе данных должны содержаться данные не менее чем о двадцати учениках. Вход на сайт должен быть защищён паролем.

24. Создать сайт для справок о загрузке преподавателей. Данные о загрузке должны выбираться по любой комбинации из следующих признаков:

- ФИО преподавателя;
- Кафедра;
- предмет

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о загрузке преподавателей должны храниться в базе данных. Описание одного преподавателя должно иметь не менее пяти признаков. В базе данных должны содержаться данные не менее чем о двадцати преподавателях. Вход на сайт должен быть защищён паролем.

25. Создать сайт для справок об отправке поездов. Данные об отправке должны выбираться по любой комбинации из следующих признаков:

- № поезда;
- дата и время;
- Станция назначения.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные об отправке поездов должны храниться в базе данных. Описание одного поезда должно иметь не менее пяти признаков. В базе данных должны содержаться данные не менее чем о двадцати поездах. Вход на сайт должен быть защищён паролем.

26. Создать сайт для справок о рейсах самолётов. Данные о рейсе должны выбираться по любой комбинации из следующих признаков:

- № рейса;
- дата и время;
- аэропорт назначения.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о рейсах самолётов должны храниться в базе данных. Описание одного рейса должно иметь не менее пяти признаков. В базе данных должны содержаться данные не менее чем о двадцати рейсах. Вход на сайт должен быть защищён паролем.

27. Создать сайт для справок о театральных спектаклях. Данные о спектакле должны выбираться по любой комбинации из следующих признаков:

- название спектакля;
- дата и время;
- театр.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о спектаклях должны храниться в базе данных. Описание одного спектакля должно иметь не менее пяти признаков. В базе данных должны содержаться данные не менее чем о двадцати спектаклях. Вход на сайт должен быть защищён паролем.

28. Создать сайт для справок о туристических поездках. Данные о поездке должны выбираться по любой комбинации из следующих признаков:

- место;
- дата и время;
- длительность поездки.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о туристических поездках должны храниться в базе данных. Описание одной туристической поездки должно иметь не менее пяти признаков. В базе данных должны содержаться данные не менее чем о двадцати туристических поездках. Вход на сайт должен быть защищён паролем.

29. Создать сайт для справок о книгах в библиотеке. Данные о книгах должны выбираться по любой комбинации из следующих признаков:

- жанр;
- автор;

- название книги

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о книгах должны храниться в базе данных. Описание одной книги должно иметь не менее пяти полей. В базе данных должны содержаться данные не менее чем о двадцати книгах. Вход на сайт должен быть защищён паролем.

30. Создать сайт для справок о многоквартирном доме. Данные о квартирах должны выбираться по любой комбинации из следующих признаков:

- номер квартиры;
- этаж;
- количество комнат

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о квартирах должны храниться в базе данных. Описание одной квартиры должно иметь не менее пяти полей. В базе данных должны содержаться данные не менее чем о двадцати квартирах. Вход на сайт должен быть защищён паролем.

• Создать сайт для справок о расписании занятий преподавателя. Данные о расписании должны выбираться по любой комбинации из следующих признаков:

- преподаватель;
- дисциплина;
- группы или подгруппа.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о расписании должны храниться в базе данных. Одна запись расписания должна иметь не менее шести полей. В базе данных должны содержаться данные не менее чем о расписаниях не менее трёх преподавателей. Вход на сайт должен быть защищён паролем.

31. Создать сайт для справок о расписании работы врачей в поликлинике. Данные о расписании должны выбираться по любой комбинации из следующих признаков:

- врач;
- часы приёма;
- специализация.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Данные о расписании должны храниться в базе данных. Одна запись расписания должна иметь не менее шести полей. В базе данных должны содержаться данные не менее чем о расписаниях не менее трёх врачей. Вход на сайт должен быть защищён паролем.

32. Создать сайт для справок о счетах клиентов в банке. Данные о счёте должны выбираться по любой комбинации из следующих признаков:

- клиент;
- номер счёта;
- период.

Хотя бы один выбираемый признак должен выбираться пользователем из всплывающего на экране списка. Выводиться должны данные об изменении

счёта за указанный период. Данные о счетах должны храниться в базе данных. В базе данных должны содержаться данные о счетах не менее, чем трёх клиентов. Вход на сайт должен быть защищён паролем.

6 семестр

1. Требуется создать базу с древовидным строением, и вывести все это одним запросом MySQL, в виде дерева, учитывая, что в каждом разделе, может быть сколько угодно подразделов.
2. Нужно разработать программу, которая считала бы сумму цифр числа, введенного пользователем. Например, есть число 123, то программа должна вычислить сумму цифр 1, 2, 3, т. е. 6.
По желанию можете сделать проверку на корректность введения данных пользователем.
3. Нужно разработать программу, которая считала бы количество вхождений какой-нибудь выбранной вами цифры в выбранном вами числе. Например, цифра 5 в числе 442158755745 встречается 4 раза
4. Разработайте программу, которая из чисел 20 ... 45 находила те, которые делятся на 5 и найдите сумму этих чисел. Рекомендую использовать функцию fmod для определения "делится число" или "не делится".
5. Нужно создать массив и заполнить его случайными числами от 1 до 100 (ф-я rand). Далее, вычислить произведение тех элементов, которые больше нуля и у которых парные индексы. После вывести на экран элементы, которые больше нуля и у которых не парный индекс.
6. Разработайте программу, которая определяла количество прошедших часов по введенным пользователем градусам. Введенное число может быть от 0 до 360.
7. Работа светофора запрограммирована таким образом: с начала каждого часа, в течении трех минут горит зеленый сигнал, следующие две минуты горит красный, дальше в течении трех минут - зеленый и т. д. Вам нужно разработать программу, которая по введенному числу определяла какого цвета сейчас горит сигнал.
8. Игральным картам присвоены следующие порядковые номера в зависимости от их достоинства: "валет" - 11, "дама" - 12, "король" - 13, "туз" - 14. Порядковые номера остальных карт соответствуют их названиям ("семерка", "восьмерка" и т. д.). Вам нужно разработать программу, которая выводила достоинство карты по заданному номеру, который будет вводит пользователь.
9. Составить программу, которая бы по заданному числу выводила его название по китайском календаре. Заданное число не может быть меньше 1924.
10. Дано натуральное число n. Вычислить: $1^2 + 2^2 + \dots + n^2$. Вывести на экран квадраты этих чисел, а также сумму квадратов этих чисел.
11. Написать функцию arithmetic, принимающую 3 аргумента: первые 2 - числа, третий - операция, которая должна быть произведена над ними. Если третий аргумент +, сложить их; если —, то вычесть; * — умножить; / — разделить

- (первое на второе). В остальных случаях вернуть строку "Неизвестная операция".
12. Написать функцию `is_year_lear`, принимающую 1 аргумент — год, и возвращающую `True`, если год високосный, и `False` иначе.
 13. Пользователь делает вклад в размере `a` рублей сроком на `years` лет под 10% годовых (каждый год размер его вклада увеличивается на 10%. Эти деньги прибавляются к сумме вклада, и на них в следующем году тоже будут проценты). Написать функцию `bank`, принимающая аргументы `a` и `years`, и возвращающую сумму, которая будет на счету пользователя.
 14. Написать функцию `is_prime`, принимающую 1 аргумент — число от 0 до 1000, и возвращающую `True`, если оно простое, и `False` - иначе.
 15. Написать функцию `date`, принимающую 3 аргумента — день, месяц и год. Вернуть `True`, если такая дата есть в нашем календаре, и `False` иначе.
 16. Написать функцию `XOR_cipher`, принимающая 2 аргумента: строку, которую нужно зашифровать, и ключ шифрования, которая возвращает строку, зашифрованную путем применения функции XOR (^) над символами строки с ключом. Написать также функцию `XOR_uncipher`, которая по зашифрованной строке и ключу восстанавливает исходную строку.
 17. Написать функцию, которой можно передавать аргументы либо списком/кортежем, либо по одному. Функция производит суммирование всех аргументов.
 18. Написать аналог `map`:
 первым аргументом идет либо функция, либо список функций
 вторым аргументом — список аргументов, которые будут переданы функциям
 полагается, что эти функции — функции одного аргумента
`>>> map([add0, add1, add2], [1, 2, 3])`
`[(1, 2, 3), (2, 3, 4), (3, 4, 5)]`
 в данном случае "развернутая" запись будет: `[(add0(1), add0(2), add0(3)), (add1(1), add1(2), add1(3)), (add2(1), add2(2), add2(3))]`
 19. Ввести с клавиатуры список, содержащий 10 элементов и записать в новый список сначала все четные значения элементов, а потом нечетные. На экран вывести исходный и новый список.
 20. Написать класс, который бы по всем внешним признакам был бы словарем, но позволял обращаться к ключам как к атрибутам.
 21. В воображаемой социальной сети есть Пользователи (`id`, имя), Фото (`id`, название, автор) и Комментарии к Фото (`id`, текст, автор, к какому Фото относится). Необходимо добавить возможность для Пользователей ставить лайки другим Пользователям, Фото или Комментариям к Фото. Нужно реализовать такие возможности:
 - пользователь не может поставить 2 лайка одной и той же сущности (например, одному и тому же Фото)
 - пользователь может отозвать лайк
 - необходимо иметь возможность посчитать число полученных сущностью лайков и вывести список Пользователей, поставивших лайки

- в будущем могут появиться новые виды сущностей, которые можно лайкать
22. Есть кинотеатр, в нем идут фильмы. У фильма есть название, длительность (пусть для простоты будет 60, 90 или 120 минут), цена билета (в разное время и дни может быть разная), время начала сеанса (один фильм может быть показан несколько раз в разное время за разную цену). Также, есть информация о купленных билетах (номер билета, на какой сеанс).
Задания:
- составь грамотную нормализованную схему хранения этих данных в БД.
 - Внеси в нее 4-5 фильмов, расписание на один день и несколько проданных билетов.
23. Создать два независимых соединения и показать на примере, что установка кодировки для первого соединения не влияет на кодировку второго соединения.
24. Вывести все строки из результата запроса с помощью функции `mysql_fetch_row()` когда заранее известны индексы полей, которые необходимо выводить.
25. Вывести все строки из результата запроса с помощью функции `mysql_fetch_row()` для всех полей, когда количество полей заранее не известно
26. Вывести в виде HTML-таблицы все строки из результата запроса (с указанием имен полей в шапке таблицы) с помощью функции `mysql_fetch_assoc()`.
27. Вывести в виде HTML-таблицы все строки из результата запроса. В шапке таблицы для каждого поля должны быть указаны имя, тип и звездочка, если поле является ключом.
28. Использовать результат одного запроса при составлении следующего запроса.
29. Выполнить запрос на вставку строки в таблицу с заданными значениями.
30. Выполнить запрос на основании переданных в скрипт данных с проверкой и экранированием входных данных

7 семестр

1. Создать скрипт конвертера любых четырех валют. Выбор валюты осуществить посредством оператора `case`. Валюты и величины брать произвольные.
2. Сделать конвертер градусов по Цельсию в градусы по Кельвина.
3. Реализовать на Ruby с использованием Rails приложение со следующим функционалом:
 - Регистрация / авторизация пользователей.
 - Создание портфеля акций (5-6 акций достаточно) для пользователя: стандартный CRUD.
 - Данные должны скачиваться с Yahoo Finance.
 - Сделать вывод графика "стоимость портфеля от времени" за 2 последних года по выбранным акциям.

4. Дан целочисленный массив. Вывести индексы массива в том порядке, в котором соответствующие им элементы образуют убывающую последовательность.
5. Дан целочисленный массив. Найти количество элементов, расположенных перед первым минимальным.
6. Дан целочисленный массив. Найти индекс последнего экстремального (то есть минимального или максимального) элемента.
7. Дан целочисленный массив. Найти максимальное количество подряд идущих минимальных элементов.
8. Дан целочисленный массив. Проверить, чередуются ли в нем положительные и отрицательные числа.
9. Дано вещественное число R и массив вещественных чисел. Найти два элемента массива, сумма которых наиболее близка к данному числу.
10. Дан целочисленный массив. Возвести в квадрат отрицательные элементы и в третью степень - положительные. Нулевые элементы - не изменять.
11. Дан целочисленный массив. Вывести индексы элементов, которые меньше своего левого соседа, и количество таких чисел.
12. Дан целочисленный массив, содержащий, по крайней мере, два нуля. Вывести сумму чисел из данного массива, расположенных между последними двумя нулями.
13. Создать документ HTML и вставить туда простой сценарий для ответа на событие, порожденное щелчком на кнопке.
14. Создать страницу, на которой производится простое вычисление на основе цен на продукцию, ее количества и комиссионных.
15. Создать документ HTML, содержащий сценарий, который получает данные со страницы, делает с ними вычисления и возвращает результат обратно на страницу.
16. Создать выпадающий список, в котором пользователь может выбрать товар; автоматическое назначение цены при выборе; вычисление размера скидок на основе стоимости покупки
17. Написать программу, которая бы случайным образом загадывала число от 1 до 100, и предлагала Вам его угадать. При неправильном предположении, программа должна выводить сообщение о том, больше загаданное число или меньше. Ведется подсчет попыток. В случае победы выводится поздравление.
18. Составьте программу, которая определяет, принадлежит ли точка $N(x, y, z)$ сфере с центром в точке $Z(a, b, c)$ и радиусом r .
19. Напишите программу, которая определяет по заданному числу месяца и по дню недели первого числа этого месяца день недели для заданного числа. (Пример: первое число - пятница, тогда 6 – вторник, задали число 6).
20. Даны две прямые, заданные уравнениями $y=k_1x+b_1$, $y=k_2x+b_2$, вводятся k_1 , b_1 , k_2 , b_2 , определить являются ли данные прямые перпендикулярными и найти их точку пересечения.
21. Известно, что сторона квадрата равна a , радиус круга r . Выведите на экран название и значение площади большей фигуры.

22. Без использования операторов условного перехода написать программу, находящую максимальное из трех чисел.
23. Напишите программу, которая по заданным переменным a , b и c типа `boolean`, выводит знак операции отношения так, чтобы следующая операция возвращала `true`: $a=(b<\text{операция}>c)$, где $\langle\text{операция}\rangle$ - операция отношения, знак которой нужно вывести на экран. (Пример: $a=true$, $b=true$, $c=true$. Вывод: `&`).
24. Составьте программу для вычисления значения функции в точке X :
- $$y(x) = \begin{cases} x \cdot x + 2, & x \leq -2 \\ x - 1, & x \geq -2 \end{cases}$$
25. Напишите программу, которая по номеру месяца выводит его название.

3.2 Примерная тематика курсовых работ:

1. Математическое и компьютерное моделирование миграционных потоков
2. Основы компьютерного проектирования и моделирования РТС
3. Основы компьютерного проектирования и моделирования радиоэлектронных средств
4. Морфологические методы решения задач классификации объектов по их изображениям
5. Поиск заданного фрагмента на изображении
6. Классификация объектов по их изображениям
7. Прогнозирование случайных процессов
8. Морфологические методы решения задач классификации объектов по их изображениям
9. Поиск заданного фрагмента на изображении
10. Классификация объектов по их изображениям
11. Прогнозирование случайных процессов
12. Морфологические методы решения задач классификации объектов по их изображениям
13. Поиск заданного фрагмента на изображении
14. Классификация объектов по их изображениям
15. Прогнозирование случайных процессов
16. Исследование и имитационное моделирование систем массового обслуживания
17. Исследование и компьютерное моделирование влияния различных факторов на систему или процесс
18. Исследование и компьютерное моделирование для прогнозирования явления, процесса, динамики развития или поведения объекта
19. Разработка и компьютерное моделирование элементов систем автоматизации
20. Компьютерное моделирование и его особенности
21. Модели решения функциональных и вычислительных задач.
22. Системный подход в моделировании систем.
23. Классификация видов моделирования.
24. Математические и информационные модели решения задач.

25. Моделирование информационных процессов. Унифицированный язык моделирования UML.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Программой предусмотрены следующие виды контроля: два текущих контроля знаний в форме тестирования (в каждом семестре), аттестацию в форме экзамена (в конце каждого семестра).

Неделя текущего контроля	Вид оценочного средства	Код компетенций, оценивающий знания, умения, навыки	Содержание оценочного средства	Требования к выполнению	Срок сдачи (неделя семестра)	Критерии оценки по содержанию и качеству с указанием баллов
Согласно графика учебного процесса	Тестирование 1, 2	ОПК-3 ПК-3,5	15 вопросов	Компьютерное тестирование; время, отведенное на процедуру - 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка - 0 Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%. Максимальная оценка – 5 баллов
Согласно графика учебного процесса	Экзамен	ОПК-3 ПК-3,5	3 вопроса	Экзамен проводится в устной форме, путем ответа на вопросы. Время, отведенное на процедуру – 45 минут.	Результаты предоставляются в день проведения экзамена	Критерии оценки: «Отлично»: • знание всех понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание всех методов, изучаемых предметов; • ответ на все вопросы билета и правильное решение задачи. «Хорошо»: • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных методов, изучаемых предметов; • ответ на вопросы билета и решение задачи с незначительными погрешностями. «Удовлетворительно»:

						<ul style="list-style-type: none"> • знание основных понятий предмета; • слабое умение использовать и применять полученные знания на практике; • пассивная работа на практических занятиях; • знание не всех методов, изучаемых предметов; • ответ не на все вопросы билета, решение задачи с ошибками. <p>«Неудовлетворительно:</p> <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; • не отвечает на вопросы билета и не умеет решать задачи.
--	--	--	--	--	--	---

4.1. Типовые вопросы, выносимые на тестирование

5 семестр

1. Какая классификация отражает время реакции системы?

- (?) быстродействующие и интерактивные системы
- (!) реального, разделенного времени, пакетной обработки
- (?) системы жесткого и мягкого реального времени

2. Какого статуса задачи не существует?

- (?) блокированная
- (?) активная
- (!) масштабируемая
- (?) готовая

3. В каком случае реализуется принцип адаптивной приоритетности?

- (?) кооперативной многозадачности
- (?) равнодоступности
- (!) приоритетной многозадачности

4. Что не относится к случаю необходимости синхронизации?

- (?) необходимость упорядочения приоритетов задач
- (?) необходимость упорядоченного доступа нескольких задач к разделяемому ресурсу
- (!) необходимость синхронизации задачи с внешними событиями

4. Какие этапы включает система контроля и управления?

- (?) разработка архитектуры системы автоматизации

(?) создание прикладной системы управления для каждого узла

(?) отладка прикладной программы в режиме эмуляции

(?) формализация постановки задачи

(!) поддержка режима реального времени

5. Что такое объектная модель языка?

(?) Когда в языке присутствуют только объекты, а субъекты отсутствуют.

(!) Объектная модель язык включает объекты, их свойства и отношения.

6. Что такое форма в языке?

(?) Это внешний образ программы.

(!) Это визуальный объект-контейнер, содержащий другие визуальные и невидимые объекты управления программой.

7. Является ли меню визуальным объектом?

(?) Нет.

(!) Да.

8. Какова роль событий, связанных с объектами?

(?) Отражать действия, происходящие с объектом.

(!) Выполнять действия в случае наступления события.

9. Как средство эффективной обработки больших текстовых файлов разрабатывался script-язык:

(!) Perl

(?) RUBY

(?) Python

10. Из перечисленного, script-языками являются:

(!) HTML

(!) Perl

(!) VRML

(?) SQL

(?) PHP

11. На каком этапе развития информационных технологий основным инструментарием стал персональный компьютер с широким спектром стандартных программных продуктов разного назначения:

(?) На первом этапе, в 1960-1970 годах.

(!) На втором этапе, с начала 1980-х годов.

(?) На третьем этапе, с начала 1990-х годов.

(?) На четвертом этапе, с конца 1990-х годов до наших дней.

12. На каком этапе развития информационных технологий стали широко использоваться в различных областях глобальные и локальные компьютерные сети:

(?) На первом этапе, в 1960-1970 годах.

(?) На втором этапе, с начала 1980-х годов.

(!) На третьем этапе, с начала 1990-х годов.

(?) На четвертом этапе, с конца 1990-х годов до наших дней.

13. Как информационные системы и технологии влияют на эффективность работы организаций — субъектов ВЭД:

(?) Снижают надежность передачи коммерческой информации.

- (!) Уменьшают расходы на осуществление коммуникаций.
- (?) Увеличивают время на подготовку и проведение внешнеторговых соглашений.
- (?) Снижают скорость документооборота.

14. Как информационные системы и технологии влияют на эффективность работы организаций — субъектов ВЭД:

- (?) Снижают надежность передачи коммерческой информации.
- (?) Увеличивают расходы на осуществление коммуникаций.
- (!) Уменьшают время на подготовку и проведение внешнеторговых соглашений.
- (?) Снижают скорость документооборота.

15. Как информационные системы и технологии влияют на эффективность работы организаций — субъектов ВЭД:

- (!) Повышают надежность передачи коммерческой информации.
- (?) Увеличивают расходы на осуществление коммуникаций.
- (?) Увеличивают время на подготовку и проведение внешнеторговых соглашений.
- (?) Снижают скорость документооборота.

16. Как информационные системы и технологии влияют на эффективность работы организаций — субъектов ВЭД:

- (?) Снижают надежность передачи коммерческой информации.
- (?) Увеличивают расходы на осуществление коммуникаций.
- (?) Увеличивают время на подготовку и проведение внешнеторговых соглашений.
- (!) Повышают скорость документооборота.

17. Кто является автором (создателем) языка программирования Java?

- (?) Даль и Ньюгорд
- (?) Томпсон и Риччи
- (!) Д. Гослинг
- (?) Б. Страуструп

18. Когда был создан язык программирования Java?

- (?) В 1970-х годах
- (?) В 1980-х годах
- (!) В 1990-х годах
- (?) В 2000-х годах

19. В каком из языков программирования фигурируют понятия апплетов и сервлетов?

- (?) C++
- (!) Java
- (?) Borland Pascal with Objects4. C#

20. Какая из программных конструкций не используется в языке программирования Java?

Варианты ответов:

- (?) ==
- (?) =
- (!) begin end
- (?) { }

21. В каком из языков программирования фигурирует понятие JVM?

- (?) C++

(?) C#

(?) Borland Pascal with Objects

(!) Java

22. Какая из приводимых ниже технологий позволила фирме Microsoft противостоять доминированию Java-технологий при программировании на стороне сервера?

(!) NET

(?) COM

(?) COM+

(?) DCOM

23. JavaScript это

(?)Объектно-ориентированный скриптовый язык программирования.

(?)Скриптовой язык программирования

(?)Прототипно-ориентированный скриптовый язык программирования.

(?)Все выше перечисленное верно

24. JavaScript используемым для разработки

(?)Веб-приложений на стороне сервера

(?)Веб-приложений на стороне клиента

(?)Оба варианта верны

25. var a = 1;

var b = ++a+a;

alert(b);

Результат работы кода - сообщение с текстом

(?)1

(?)2

(?)3

(?)4

(?)Этот код не будет работать

26. var a = 1;

var b = { toString:function(){return '1'} };

alert(a+b);

Результат работы кода - сообщение с текстом

(?)1[object Object]

(?)2

(?)11

(?)Код не работает

27. var a = 1;

var b = a++;

alert(b);

Результат работы кода - сообщение с текстом

(?)1

(?)2

(?)Другое

28. var a = 1;

{

```
var a = 5;
}
alert(a);
```

Результат работы кода - сообщение с текстом

(?)1

(?)5

(?)Другое

29. Какие из этих вариантов не задает массив из элементов "a", "b"?

(?)var a = new Array("a","b")

(?)var a = ["a", "b"]

(?)var a = ("a", "b")

(?)var a = "a,b".split(',')

(?)Все подойдут

(?)Ни один не подойдет

30. var a = true;

var b = false;

alert((a&&b)===!(a&&b));

Результат работы кода - сообщение с текстом

(?)true

(?)false

(?)Другое

(?)Код содержит ошибку

6 семестр

1. Что такое обработчик событий?

(?) Модуль, генерирующий события.

(!) Процедура или функция, вызываемая в случае наступления события, с ней связанного.

2. Что такое библиотека?

(?) Совокупность взаимосвязанных программ.

(!) Совокупность процедур и функций.

3. Какие объекты могут храниться в библиотеках?

(?) Визуальные.

(!) Процедуры, функции.

4. Что такое структура данных в СУБД?

(?) Иерархия подчиненности таблиц данных.

(!) Совокупность таблиц данных с их полями и ссылками между ними.

5. Что такое SQL-запрос?

(?) Запрос на проверку целостности таблицы данных.

(!) Совокупность операторов и условий, обеспечивающих выборку из таблиц необходимых данных.

6. Что такое механизм транзакций в СУБД?

(?) Передача данных.

(!) Возможность отката в сбойных ситуациях.

7. Можно ли средствами SQL-запросов создавать новые таблицы?

(?) Нет.

(!) Да.

8. Каковы функции условного оператора?

(?) Проверить условия завершения программы.

(!) Исполнение необходимых действий при выполнении проверяемых условий.

9. Что позволяет реализовать оператор цикла?

(?) Последовательную логическую проверку условий.

(!) Многократное повторение однотипных операций.

10. Что такое область видимости переменных?

(?) Те программы, в которых могут использоваться переменные.

(!) Те фрагменты кода, в которых переменной можно пользоваться.

11. Фирмой Microsoft разработана СУБД:

• SQL Server

12. Структурированный язык запросов обозначается:

• SQL

13. При технологии объектного связывания данных каждая локальная СУБД выполняет роль:

• SQL-сервера

14. Протокол открытого доступа к базам данных в рамках языка SQL обозначается:

• ODBC

15. Какие главные преимущества хранения информации в базах данных

(!) Подходят все перечисленные пункты

(?) Многообразие использования данных

(?) Ускорение обработки запросов к системе и уменьшение избыточности данных

(?) Простота и удобство внесения изменений в базы данных

16. Что означает – программа или комплекс программ служащих для полнофункциональной работы с данными (СУБД)?

(!) Система управления базами данных

(?) Система управления базой доступа

(?) Система упрощенного базового доступа

(?) Совокупность управляющих баз данных

17. По технологии обработки данных базы данных подразделяют на:

(!) Централизованные и распределенные

(?) Периферийные и централизованные

(?) Внутренние и наружные

(?) Простые и сложные

18. По способу доступа к данным базы данных подразделяют на:

(!) Базы данных с локальным доступом и базы данных с удаленным (сетевым доступом)

(?) Простые и сложные

(?) Быстрые и медленные

(?) Проводные и беспроводные

19. По типу связи между данными базы данных подразделяют на:

(!) Иерархические, сетевые, реляционные, объектно-ориентированные

(?) Компьютерные и персональные

(?) Модульные, модемные и сетевые

(?) Основные и дополнительные

20. С чего всегда начинается создание базы данных?

(!) С разработки структуры ее таблиц

(?) С запуска компьютера и запуска программы просмотрщика баз данных

(?) С создания макета документа

(?) С собеседования и обсуждения проблемы построения базы данных

21. Какой будет результат выполнения следующего кода:

```
<?php
```

```
$sport = 'soccer';
```

```
$statement = 'He plays $sport';
```

```
echo $statement;
```

```
?>
```

(!)He plays \$sport

He plays soccer

22. Как определить константу?

```
variable("FOO", "BAR");
```

```
constant("FOO", "BAR");
```

```
(!)define("FOO", "BAR");
```

```
defineconstant("FOO", "BAR");
```

23 PHP код не может быть встроено в HTML. Это выражение...

Ложь(!)

Правда

24. Является ли использование сокращённых PHP тегов хорошей практикой?

Да

Нет(!)

25. Название переменных в PHP чувствительны к регистру. Это выражение.

Правда

Ложь(!)

26. Какой из типов данных в PHP 5 передаётся по ссылке (по умолчанию)?

string

array

object (!)

class

27. Какая функция в PHP позволяет найти вхождение подстроки?

str_repeat()

strtok()

strpos() (!)

○ substr()

28. Что выведет следующий код в Python 2?

```
print xrange(10,25)
```

```
[10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25]
```

```
[9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24]
```

```
xrange(10,25)
```

```
<xrange from 10 to 25>
```

29. Как называется встроенный в языке Python тип данных неупорядоченной коллекции из нуля или более пар ключ-значение?

dict

set

list

frozenset

30. Какое обычно расширение имеют файлы на языке Python в среде Windows?

*.py

*.pys

*.pyt

*.pi

31. Какое расширение имеют файлы с графическим интерфейсом на языке Python в среде Windows?

*.pit

*.pyt

*.py

*.pyw

32. Какой из типов чисел с плавающей точкой является типом из стандартной библиотеки языка Python?

complex

decimal.Decimal

int

float

33. Какие из приведенных стилей программирования поддерживает язык Python?

a) Процедурный

b) Объектно-ориентированный

c) Функциональный

d) Смешанный

a и b

b и c

все

a, b и c

34. Какая функция используется для вывода строки на печать в языке Python?

fprintf()

echo()

printf()

print()

35. Язык Python:

интерпретируемый
компилируемый

36. Как с помощью SQL удалить из таблицы Persons записи, в которых поле FirstName имеет значение 'Peter'?

- a. DELETE ROW FirstName='Peter' FROM Persons
- (?) DELETE FirstName='Peter' FROM Persons
- (!) DELETE FROM Persons WHERE FirstName = 'Peter'

37. Как с помощью SQL вывести все записи из таблицы Persons, отсортированные по полю FirstName в убывающем порядке?

- a. SELECT * FROM Persons SORT 'FirstName' DESC
- (!) SELECT * FROM Persons ORDER BY FirstName DESC
- c. SELECT * FROM Persons SORT BY 'FirstName' DESC
- d. SELECT * FROM Persons ORDER FirstName DESC

38. Как с помощью SQL вывести все записи из таблицы Persons, в которых в поле FirstName содержится значение Peter?

- a. SELECT [all] FROM Persons WHERE FirstName='Peter'
- (!) SELECT * FROM Persons WHERE FirstName='Peter'
- c. SELECT [all] FROM Persons WHERE FirstName LIKE 'Peter'
- d. SELECT * FROM Persons WHERE FirstName<>'Peter'

39. Что обозначает аббревиатура SQL?

- a. Strong Question Language
- b. Structured Question Language
- (!) Structured Query Language

40. Как с помощью SQL узнать число записей в таблице Persons?

- (!) SELECT COUNT(*) FROM Persons (true)
- b. SELECT COLUMNS() FROM Persons
- c. SELECT COUNT() FROM Persons
- d. SELECT COLUMNS(*) FROM Persons

41. Как с помощью SQL вывести все записи из таблицы Persons, в которых в поле FirstName значение начинается с буквы 'a'?

- a. SELECT * FROM Persons WHERE FirstName LIKE '%a'
- b. SELECT * FROM Persons WHERE FirstName='%a%'
- c. SELECT * FROM Persons WHERE FirstName='a'
- (!) SELECT * FROM Persons WHERE FirstName LIKE 'a%'

42. Какой оператор SQL используется для извлечения данных из базы данных?

- a. GET
- b. EXTRACT
- c. OPEN
- (!) SELECT

43. Как с помощью SQL вывести данные из всех полей таблицы Persons?

- a. SELECT *.Persons
- b. SELECT [all] FROM Persons
- (!) SELECT * FROM Persons
- d. SELECT Persons

44. Какое ключевое слово SQL используется для отсортированного вывода?

- a. SORT BY
- b. SORT
- (!) ORDER BY
- d. ORDER

45. Какой оператор SQL используется для вывода только различающихся значений?

- a. SELECT UNIQUE
- (!) SELECT DISTINCT
- c. SELECT DIFFERENT

7 семестр

1. Что в языке называют конструктором?

- (?) Программа, автоматически конструирующая отдельные элементы кода.
- (!) Оператор, выделяющий для объекта необходимую область памяти.

2. Что в языке называют деструктором?

- (?) Уничтожитель программы.
- (!) Оператор, освобождающий память, ранее выделенную для объекта.

3. Что такое компиляция программного кода?

- (?) Специальная предварительная обработка программного кода для поиска ошибок.
- (!) Перевод программного кода в бинарный код, непосредственно исполняемый процессором.

4. Что такое интерпретация программного кода?

- (?) Вариант его прочтения.
- (!) Исполнение программы без ее компиляции путем непосредственного исполнения каждого оператора.

5. Что называют директивой компилятору?

- (?) Элемент логики программного кода.
- (!) Команды, которые исполняются при компиляции текста программы.

6. Какова роль оператора обработки исключений?

- (?) Срабатывание в исключительных ситуациях.
- (!) Предотвращение аварийной остановки программы.

7. Что такое булева переменная?

- (?) Переменная, принимающая целочисленные значения.
- (!) Переменная, которая может принимать лишь значения True, False.

8. Языки программирования, адаптированные специально для Internet, называются:

- script-языками

9. Для организации виртуальных трехмерных интерфейсов был разработан script-язык:

- VRML

10. Система типов в Ruby:

- слабая, динамическая
- сильная, динамическая

сильная, статическая

слабая, статическая

11. Какой тип данных Ruby является примитивным?

ни один из перечисленных

Integer

nil

Byte

12. Какой синтаксис для многострочных комментариев используется в Ruby?

=begin Это комментарий =end

все перечисленные варианты

/* Это комментарий */

" Это комментарий "

4.2. Типовые вопросы, выносимые на экзамен (5 семестр)

1. Основные понятия и конструкции языка Perl.
2. Основные теги языка гипертекстовой разметки.
3. Средства создания и редактирования HTML страницы.
4. Изменение цвета текста и фона в JavaScript.
5. Задание фонового рисунка в JavaScript.
6. Форматирование текста в JavaScript.
7. Задание параграфов, заголовков текста в Perl.
8. Работа со шрифтами в JavaScript;
9. Вставка картинок, свойства изображений в Perl.
10. Создание ссылок, ссылка в виде картинки в JavaScript.
11. Создание внутренних и внешних ссылок в JavaScript.
12. Ссылка на картинку, электронную почту, музыкальный файл и т.д. в Perl.
13. Создание навигационных карт в Perl.
14. Создание таблицы в JavaScript.
15. Горизонтальное и вертикальное объединение ячеек таблицы в Perl;
16. Границы, заливка и рамки таблицы в JavaScript.
17. Вывод специальных символов в Perl;
18. Рисование линий в JavaScript.
19. Нумерованные и ненумерованные списки в Perl
20. Создание фреймов в JavaScript.
21. Горизонтальное и вертикальное разбиение страницы на фреймы в Perl.
22. Ссылки между страницами фрейма в Perl.
23. Границы, рамки фрейма, полоса прокрутки в JavaScript.
24. Переменные на языке Perl.
25. Операторы цикла в JavaScript.
26. Условный оператор в JavaScript.

27. Область видимости переменных в JavaScript.

4.3. Типовые вопросы, выносимые на экзамен (6 семестр)

1. Основы языка PHP.
2. Назначение, применение, понятие объектной модели языка.
3. Иерархия классов.
4. Структура программы в PHP.
5. Типы данных и переменные в PHP,
6. Операторы языка PHP.
7. Программирование свойств окна браузера в PHP.
8. Программирование форм в PHP: текст в полях ввода, списки и выпадающие меню, кнопки, картинки, обмен данными.
9. Программирование графики в PHP: графика и таблицы, графика и обработка событий, вертикальные и горизонтальные меню.
10. Программирование гипертекстовых переходов в PHP.
11. Интерфейс Web-приложений в PHP.
12. Протокол: общая структура сообщений, методы доступа.
13. Виды интерфейса пользователя в PHP.
14. Спецификация CGI, понятие CGI – скрипта.
15. Типы запросов, механизмы приема данных скриптом, механизм генерации отклика скриптом.
16. Библиотека. Режимы буферизации.
17. Коллективные взаимодействия процессов в PHP. Управление группами и коммутаторами.
18. Структура данных СУБД MySQL.
19. Операторы SQL-запросов.
20. Особенности использования СУБД MySQL в интернет-приложениях
21. Механизм транзакций в СУБД.
22. Отличия диалектов СУБД MySQL.
23. Предикаты СУБД MySQL.
24. Представления результатов запросов СУБД MySQL.
25. Создание новых таблиц данных средствами СУБД MySQL.

4.4. Типовые вопросы, выносимые на экзамен (7 семестр)

1. Основные понятия и конструкции языка HTML;
2. Основные теги языка гипертекстовой разметки;
3. Средства создания и редактирования HTML страницы;
4. Изменение цвета текста и фона;

5. Задание фонового рисунка;
6. Форматирование текста.
7. Задание параграфов, заголовков текста;
8. Работа со шрифтами;
9. Вставка картинок, свойства изображений;
10. Создание ссылок, ссылка в виде картинки;
11. Создание внутренних и внешних ссылок;
12. Ссылка на картинку, электронную почту, музыкальный файл и т.д.;
13. Создание навигационных карт.
14. Создание таблицы;
15. Горизонтальное и вертикальное объединение ячеек таблицы;
16. Границы, заливка и рамки таблицы;
17. Вывод специальных символов;
18. Рисование линий;
19. Нумерованные и ненумерованные списки;
20. Создание фреймов;
21. Горизонтальное и вертикальное разбиение страницы на фреймы;
22. Ссылки между страницами фрейма;
23. Границы, рамки фрейма, полоса прокрутки;
24. Переменные на языке PHP
25. Операторы цикла
26. Условный оператор
27. Область видимости переменных

*ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И ТЕХНОЛОГИЙ
КАФЕДРА МАТЕМАТИКИ И ЕСТЕСТВЕННОНАУЧНЫХ ДИСЦИПЛИН*

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ
«Прикладные интернет-технологии»**

(Приложение 2 к рабочей программе)

Направление подготовки: 01.03.02 Прикладная математика и информатика

Профиль: Искусственный интеллект и управление в ракетно-космических системах

Уровень высшего образования: бакалавр

Форма обучения: очная

Королев
2021

1. Общие положения

Цели дисциплины:

1. формирование способности к восприятию, обобщению и анализу информации, необходимой для оценки вариантов, обеспечения и поддержки принятия эффективных конструкторских и управленческих решений;
2. освоение необходимого инструментария, применяемого при оценке вариантов, выборе эффективных конструкторских и управленческих решений;
3. формирование готовности применять методы математического анализа и моделирования сложных систем и процессов в профессиональной деятельности.

Задачи дисциплины:

1. Освоение студентами теоретических методов дисциплины, применяемых при оценке вариантов и выборе эффективных конструкторских и управленческих решений;
2. Получение студентами умений и навыков, применяемых для решения практических задач оценки вариантов и выбора эффективных конструкторских и управленческих решений.

2. Указания по проведению практических занятий

Практическое занятие 1.

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия: *Построение интернет-приложений средствами JavaScript, Perl.*

Продолжительность занятия – 32/- ч.

Практическое занятие 2.

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия: *Построение интернет-приложений средствами PHP, Python. СУБД MySQL.*

Продолжительность занятия – 32/- ч.

Практическое занятие 3.

Вид практического занятия: *смешанная форма практического занятия.*

Тема и содержание практического занятия: *Построение интернет-приложений средствами Ruby, VBScript.*

Продолжительность занятия – 32/- ч.

3. Указания по проведению лабораторного практикума

Не предусмотрено учебным планом.

4. Указания по проведению самостоятельной работы студентов

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1.	Скриптовые (сценарные) языки программирования JavaScript, Perl	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (построение баз данных).
2.	Языки программирования PHP, Python. СУБД MySQL	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (построение баз данных).
3.	Языки программирования Ruby, VBScript	1. Подготовка к практическим занятиям по материалам лекций и учебной литературы. 2. Выполнение практических заданий 3. Самостоятельное изучение некоторых вопросов дисциплины (построение баз данных).

5. Указания по проведению контрольных работ для студентов факультета заочного обучения

Учебным планом не предусмотрено.

6. Перечень основной и дополнительной учебной литературы

Основная литература:

2. Интернет-технологии: Учебное пособие / С.Р. Гуриков. - М.: Форум: НИЦ ИНФРА-М, 2015. - 184 с - [электронный ресурс] // <http://znanium.com/catalog.php?bookinfo=488074>.

2. Технология разработки интернет ресурсов: курс лекций: [16+] / авт.-сост. И.А. Журавлёва; Министерство науки и высшего образования Российской Федерации, Северо-Кавказский федеральный университет. – Ставрополь: Северо-Кавказский Федеральный университет (СКФУ), 2018. – 171 с.: ил. – Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=562579>.

Дополнительная литература:

2. Хныкина, А.Г. Информационные технологии: учебное пособие / А.Г. Хныкина, Т.В. Минкина; Министерство образования и науки Российской Федерации, Северо-Кавказский федеральный университет. - Ставрополь: СКФУ, 2017. - 126 с.: схем., ил. - Библиогр. в кн.; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=494703>

2. Технологии создания интеллектуальных устройств, подключенных к интернет: учебное пособие / А.В. Приемышев, В.Н. Крутов, В.А. Тряель, О.А. Коршакова. — 2-е изд., стер. — Санкт-Петербург: Лань, 2018. — 100 с. — ISBN 978-5-8114-2310-1. — Текст: электронный // Электронно-библиотечная система «Лань»: [сайт]. — URL: <https://e.lanbook.com/book/103911>.

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Интернет-ресурсы:

1. <http://biblioclub.ru/> - Университетская библиотека ONLINE.
2. <http://www.rucont.ru/> - Электронная библиотека Руконт.
3. <http://www.znaniium.com/> - Электронная библиотека издательства Инфра-М.
4. <http://e.lanbook.com/> - Электронная библиотека Лань.
5. <http://www.olap.ru/> - Сайт OLAP-технологий.
6. <http://www.anylogic.ru/> - Сайт системы имитационного моделирования AnyLogic, содержащий методические и учебные материалы по имитационному моделированию в свободном доступе для различных прикладных сфер.
7. <http://www.runthemodel.com/> - Библиотека имитационных моделей, находящихся в свободном доступе, выполненных в среде AnyLogic.
8. <https://ru.atlassian.com/software/jira/> - Система планирования работ проекта ПО.
9. <https://www.jetbrains.com/youtrack/> - Система планирования работ проекта ПО.
10. <https://ru.atlassian.com/software/confluence> - Система хранения проектной документации.
11. <https://git-scm.com/> - Распределенная система контроля версий ПО.
12. <https://ru.wikipedia.org/wiki/Subversion> - Централизованная система контроля версий ПО.
13. <https://jenkins-ci.org/> - Система непрерывной интеграции проекта ПО.
14. <https://www.docker.com/> - Система виртуализации окружения.

8. Перечень информационных технологий

Перечень программного обеспечения: *MS Office, Mathcad, AnyLogic, Java, JIRA Software, Confluence, SubVersion, Git, Docker.*

Информационные справочные системы: *Электронные ресурсы образовательной среды Университета*