



Федеральное агентство по рыболовству
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Калининградский государственный технический университет»
(ФГБОУ ВО «КГТУ»)

УТВЕРЖДАЮ
Начальник УРОПСИ

Фонд оценочных средств
(приложение к рабочей программе модуля)
«ТЕХНОЛОГИИ И МЕТОДЫ ПРОГРАММИРОВАНИЯ»

основной профессиональной образовательной программы специалитета
по специальности

**10.05.03 ИНФОРМАЦИОННАЯ БЕЗОПАСНОСТЬ АВТОМАТИЗИРОВАННЫХ
СИСТЕМ**

Специализация

«БЕЗОПАСНОСТЬ ОТКРЫТЫХ ИНФОРМАЦИОННЫХ СИСТЕМ»

ИНСТИТУТ
РАЗРАБОТЧИК

цифровых технологий
кафедра информационной безопасности

1. РЕЗУЛЬТАТЫ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Таблица 1 – Планируемые результаты обучения по дисциплине, соотнесенные с установленными индикаторами достижения компетенций

Код и наименование компетенции	Индикаторы достижения компетенции	Дисциплина	Результаты обучения (владения, умения и знания), соотнесенные с компетенциями/индикаторами достижения компетенции
ОПК-7: Способен создавать программы на языках общего назначения, применять методы и инструментальные средства программирования для решения профессиональных задач, осуществлять обоснованный выбор инструментария программирования и способов организации программ.	ОПК-7.2: Использует современные технологии и методы программирования, структуру и архитектуру программного обеспечения с использованием современных методологий и средств автоматизации проектирования программного обеспечения, показатели качества программного обеспечения, возможности базового языка программирования для решения типовых профессиональных задач.	Технологии и методы программирования	<p><u>Знать:</u> современные технологии и методы программирования, структуру и архитектуру программного обеспечения.</p> <p><u>Уметь:</u> применять методы и инструментальные средства программирования для решения профессиональных задач.</p> <p><u>Владеть:</u> основными средствами и методами разработки алгоритмов; основными приемами структурного программирования на языке высокого уровня.</p>

2. ПЕРЕЧЕНЬ ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПОЭТАПНОГО ФОРМИРОВАНИЯ РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДИСЦИПЛИНЫ (ТЕКУЩИЙ КОНТРОЛЬ) И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

2.1 Для оценки результатов освоения дисциплины используются:

- оценочные средства текущего контроля успеваемости;
- оценочные средства для промежуточной аттестации по дисциплине.

2.2 К оценочным средствам текущего контроля успеваемости относятся:

- тестовые задания;
- задания и контрольные вопросы по лабораторным работам;

2.3 К оценочным средствам для промежуточной аттестации по дисциплине, проводимой в форме зачета и экзамена, относятся:

- задания по курсовой работе;

- промежуточная аттестация в форме зачета проходит по результатам прохождения всех видов текущего контроля успеваемости;

- экзаменационные вопросы.

3. ОЦЕНОЧНЫЕ СРЕДСТВА ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ

3.1 Типовые варианты тестовых заданий приведены в Приложении № 1, ключи ответов – в Приложении № 4.

Таблица 3. Шкала оценок уровня освоения дисциплины по тесту

Оценка			
Неудовлетворительный	Пороговый	Углубленный	Продвинутый
«2» (неудовлетворительно)	«3» (удовлетворительно)	«4» (хорошо)	«5» (отлично)
Менее 50% правильных ответов.	50-70% правильных ответов.	71-90% правильных ответов.	91-100% правильных ответов.

3.2. Образцы заданий для лабораторных работ приведены в Приложении № 2, В Приложении № 5 даны типовые задания для защиты лабораторных работ.

4. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

4.1 Промежуточная аттестация по дисциплине проводится в форме зачета, экзамена. Типовые варианты вопросов к экзамену приведены в Приложении № 3

К зачету, экзамену допускаются студенты, положительно аттестованные по результатам текущего контроля.

Дисциплина рассчитана на два семестра (3 и 4 семестр), в 3 семестре проводится зачет, в 4 семестре – экзамен.

Таблица 2. Шкала оценок уровня освоения дисциплины по экзамену.

Оценка			
Неудовлетворительный	Пороговый	Углубленный	Продвинутый
неудовлетворительно	удовлетворительно	хорошо	отлично
Не знает значительной части программного материала, допускает существенные	Усвоил только основной материал, но не знает отдельных деталей,	Твердо знает программный материал, грамотно и по существу излагает его, не	Глубоко и прочно усвоил весь программный материал, исчерпывающе, последовательно, грамотно и

ошибки, с большими затруднениями выполняет практические задания, задачи.	допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий.	допускает существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий.	логически стройно его изложил, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок.
--	--	--	--

4.3. Примерные темы курсовой работы приведены в Приложении № 4.

Таблица 4. Шкала оценок курсовой работы

Оценка			
Неудовлетворительный	Пороговый	Углубленный	Продвинутый
«2» (неудовлетворительно)	«3» (удовлетворительно)	«4» (хорошо)	«5» (отлично)
Работа носит реферативный характер, студент допускает существенные ошибки при защите, с большими затруднениями отвечает на вопросы, оформление работы не соответствует правилам.	Работа носит реферативный характер, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении при защите, оформление работы имеет незначительные отклонения от правил.	В работе углублены теоретические и практические знания, материал излагается грамотно и по существу, не допускается существенных неточностей в ответе на вопрос, оформление работы соответствует правилам.	В процессе выполнения работы приобретены навыки самостоятельного планирования и выполнения научно-исследовательской работы; получен опыт сбора и обработки исходного материала, анализа научно-технической литературы, материал излагается грамотно, оформление работы соответствует правилам.

5. СВЕДЕНИЯ О ФОНДЕ ОЦЕНОЧНЫХ СРЕДСТВ И ЕГО СОГЛАСОВАНИИ

Фонд оценочных средств для аттестации по дисциплине «Технологии и методы программирования» представляет собой компонент основной профессиональной образовательной программы специалитета по специальности 10.05.03 Информационная безопасность автоматизированных систем (специализация «Безопасность открытых информационных систем»).

Фонд оценочных средств рассмотрен и одобрен на заседании кафедры информационной безопасности 20.04.2022 г. (протокол № 7).

Заведующий кафедрой



Н.Я. Великите

Приложение № 1

ТИПОВЫЕ ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ

(семестр 3)

ВАРИАНТ 1	
1.	<p>Определение изображения в качестве гиперссылки:</p> <ol style="list-style-type: none"> 1. <code> IMG SRC="image.gif"></code> 2. <code></code> 3. <code> <IMG="image.gif"></code>
2.	<p>Выравнивание текста в ячейке таблицы происходит с помощью атрибута:</p> <ol style="list-style-type: none"> 1. HALIGN 2. VALIGN 3. ALIGN
3.	<p>Между границей страницы и изображением задает горизонтальное расстояние атрибут тега <code></code>:</p> <ol style="list-style-type: none"> 1. BORDER 2. HSPACE 3. VSPACE
4.	<p>Тегу <code><AREA></code> принадлежит атрибут:</p> <ol style="list-style-type: none"> 1. SRC 2. SHAPE 3. CIRCLE
5.	<p>Переключатель создается в фрагменте кода:</p> <ol style="list-style-type: none"> 1. <code><input type="checkbox" name="a1" value="1"><input type="checkbox" name="a1" value="2"><input type="text" name="a1" value="2"></code> 2. <code><input type="radiobutton" name="a1" value="1"><input type="radiobutton" name="a1" value="2"></code> 3. <code><input type="radio" name="a1" value="1"><input type="radio" name="a1" value="2"></code>
6.	<p>Текст выровнен по центру ячеек в таблице:</p> <ol style="list-style-type: none"> 1. <code><table align=""center"" width=""300""></code> 2. <code><table align="center"></code> 3. <code><table align=""left""></code>
7.	<p>Правильное описание элемента TR:</p> <ol style="list-style-type: none"> 1. <code><TR> <TD>ячейка1</TD></TR></code> 2. <code><TD> <TR>ячейка1ячейка2<TD></code> 3. <code><TR> <TD>ячейка1</code>
8.	<p>Форма будет отправлена на обработку в случае:</p> <ol style="list-style-type: none"> 1. <code><form method="post" action="http://www.alexfine.ru/shop/"></code> 2. <code><form method="default" action="http://www.alexfine.ru/shop/sp.pl"></code> 3. <code><form method="get" action="http://www.alexfine.ru/"></code>
9.	<p>Создавать прокручивающиеся списки в формах можно с помощью тега:</p> <ol style="list-style-type: none"> 1. TEXTAREA 2. SELECT 3. INPUT
10.	<p>Правильное разбиение монитора на строки:</p> <ol style="list-style-type: none"> 1. <code><FRAMESET ROWS=""50%, 50%, 25%""></code> 2. <code><FRAMESET ROW=""25%, 50%, 25%""></code> 3. <code><FRAMESET ROWS=""25%, 50%, 25%""></code>

11.	Недостаток бесплатного хостинга: 1. коммерческая реклама от поставщика услуги 2. отсутствие вариантов размещения 3. доменное имя
12.	Услуга размещения сайта на сервере, постоянно находящемся в сети Интернет: 1. хостинг 2. адаптация 3. моделинг
13.	Проектированием структуры web-сайта занимается: 1. web-программист 2. провайдер 3. web-дизайнер
14.	Тег – это: 1. гиперссылка 2. команда, заключенная в угловые скобки 3. указатель ссылки
15.	HTML – это: 1. язык разметки гипертекста 2. страница Internet Explorer 3. браузер
16.	При наполнении страниц сайта информационными материалами не следует: 1. ставить точку в названиях страниц, если они состоят из одного предложения 2. применять краткие названия пунктов 3. избегать слишком длинных текстов
17.	Тэг может быть: 1. главным 2. основным 3. закрывающим
18.	Для создания Web-страниц используется эта программа: 1. Блокнот 2. ScanDisk 3. QBasic
19.	Для просмотра Web-страниц в Интернете используются программы: 1. Internet Explorer или NetScape Navigator 2. MicroSoft Word или Word Pad 3. HTMLPad или Front Page
20.	Локальные ссылки на Web-странице могут обеспечить переход 1. на любую web-страницу любого сервера Интернет 2. только на web-страницы данного сервера 3. только в пределах данной web-страницы
ВАРИАНТ 2	
1.	Ошибочное определение гиперссылки. 1. alexfine 2. alexfine 3. alexfine
2.	Атрибут CELLSPACING у элемента разметки TABLE определяет: 1. расстояние от содержания до границы ячейки 2. интервал между ячейками таблицы 3. ширину ячейки

3.	Нумерованные списки создаются с помощью тега: 1. 2. 3. <DT>
4.	Заголовок документа HTML определяет тэг: 1. HTML 2. BODY 3. HEAD
5.	Правильное описание синтаксиса тега SCRIPT: 1. <script type="тип_языка_программирования">текст программы 2. <script name="язык_программирования">текст программы 3. <script type="тип_документа">текст программы
6.	Тело документа HTML определяет тег: 1. META 2. BODY 3. HTML
7.	Файл изображения и путь к нему указывает атрибут тега : 1. src 2. alt 3. align
8.	Данные формы будут переданы обработчику в случае: 1. <form method="get" action="http://www.alexfine.ru/"> 2. <form method="post" action="http://www.alexfine.ru/help/first.pl"> 3. <form method="try" action="http://www.alexfine.ru/help/script.php?param=test">
9.	Для отправки формы можно применять методы: 1. TRY, PUT 2. POST, GET 3. HEAD, MAILTO
10.	Бегущую строку можно организовать с помощью тега: 1. <MARQUEE> 2. <DIRECTION> 3. <BEHAVIOR>
11.	Схему страницы, на которой представлены элементы, имеющиеся на страницах сайта, называют: 1. матрица 2. шаблон 3. фундамент
12.	Перед размещением сайта в сети Интернет следует провести его тестирование, чтобы убедиться в том, что он правильно отображается разными: 1. сайтами 2. страницами 3. браузерами
13.	При наполнении страниц сайта информационными материалами не следует: 1. ставить точку в названиях страниц, если они состоят из одного предложения 2. избегать слишком длинных текстов 3. применять краткие названия пунктов
14.	Услугу по размещению сайта на сервере, постоянно находящемся в сети Интернет называют: 1. сервис

	2. домен 3. хостинг
15.	Сайт можно создать, воспользовавшись: 1. языком программирования С 2. языком разметки гипертекста HTML 3. электронными таблицами
16.	Тэг может быть: 1. автономным 2. непарным 3. вспомогательным
17.	Для выравнивания текста в документе используется атрибут: 1. src 2. valign 3. align
18.	Гипертекст – это: 1. текст очень большого размера 2. структурированный текст, где возможны переходы по выделенным меткам 3. текст, в котором используется шрифт большого размера
19.	Сайт можно создать: 1. сохранив документ в формате HTML 2. с помощью электронных таблиц 3. воспользовавшись языком программирования Паскаль
20.	Гиперссылки задаются тегом: 1. ... 2. <H1>...</H1> 3. <A>...
ВАРИАНТ 3	
1.	Ширина промежутков в таблице между ячейками составит 20 пикселей: 1. <table cellspacing="20"> 2. <table gridspacing="20"> 3. <table cellpadding="20">
2.	Задать цвет фона страницы позволяет атрибут тэга BODY: 1. background 2. bgcolor 3. set
3.	Атрибут выравнивания align имеет более высокий приоритет случае: 1. <TH align="left"> 2. <COL align="left"> 3. <TABLE align="left">
4.	Гипертекстовую ссылку из документа 1.html на другой документ задает: 1. ссылка 2. ссылка 3. ссылка
5.	Для определения положения изображения относительно окружающего текста используется атрибут ALIGN: 1. left 2. check 3. base
6.	Тег BASE правильно организован в случае:

	<ol style="list-style-type: none"> 1. <base href=""><a>http://www.alexfine.ru/intro.html" target=new> 2. <base a="" href="alexfine.ru/intro.html"> 3. <base href=""><a>http://www.alexfine.ru/intro.html" target=new >
7.	<p>Позволяет изменять цвет активных гиперссылок атрибут тега BODY:</p> <ol style="list-style-type: none"> 1. COLOR 2. VLINK 3. ALINK
8.	<p>HTML – это:</p> <ol style="list-style-type: none"> 1. язык программирования 2. структурная разметка 3. гипертекстовая разметка
9.	<p>Разделить HTML-страницу на кадры можно с помощью конструкции:</p> <ol style="list-style-type: none"> 1. <FRAME><FRAMESET>...</FRAME> 2. <FRAMESET><FRAME>...</FRAMESET> 3. <FRAME><FRAMESET>...</FRAMESET>
10.	<p>Правильное разбиение монитора на столбцы:</p> <ol style="list-style-type: none"> 1. <FRAMESET COLS="50%, 25%, 25%"> 2. <FRAMESET COL="25%, 50%, 25%"> 3. <FRAMESET COLS="25%, 50%, 50%">
11.	<p>Организация переходы с одной страницы на другую – это:</p> <ol style="list-style-type: none"> 1. выход 2. реверс 3. навигация
12.	<p>Чтобы отличать теги от текста, их заключают в:</p> <ol style="list-style-type: none"> 1. круглые скобки 2. угловые скобки 3. фигурные скобки
13.	<p>Сайт можно создать, воспользовавшись языком:</p> <ol style="list-style-type: none"> 1. программирования Си 2. программирования Паскаль 3. разметки гипертекста HTML
14.	<p>При наполнении страниц сайта информационными материалами не следует:</p> <ol style="list-style-type: none"> 1. использовать пестрый фон 2. применять краткие названия пунктов 3. использовать одинаковые приемы форматирования абзацев
15.	<p>Тэг может быть:</p> <ol style="list-style-type: none"> 1. четким 2. авторитетным 3. парным
16.	<p>Они используются для разбивки окна браузера на несколько областей, каждая из которых представляет собой отдельный HTML-документ:</p> <ol style="list-style-type: none"> 1. фреймы 2. гиперссылки 3. контейнеры
17.	<p>Для создания Web-страниц используется эта программа:</p> <ol style="list-style-type: none"> 1. MS FrontPage 2. Defrag 3. Turbo Pascal

18.	Гиперссылки на Web-странице могут обеспечить переход 1. только на web-страницы данного сервера 2. на любую web-страницу любого сервера Интернет 3. только в пределах данной web-страницы
19.	Локальные ссылки задаются тегом: 1. ... 2. <A>... 3. <H1>...</H1>
20.	При программировании фреймов в HTML-коде необходимо: 1. обязательное присутствие тега <BODY> 2. обязательное присутствие и тега <BODY>, и тега <FRAMESET> 3. присутствие тега < FRAMESET> вместо тега < BODY >

ТИПОВЫЕ ВАРИАНТЫ ТЕСТОВЫХ ЗАДАНИЙ (семестр 4)

ВАРИАНТ 1	
1.	Главный критерий качества программного изделия: 1. надежность 2. универсальность 3. защищенность 4. эффективность
2.	Синтаксис языка программирования: 1. правильно выбранные идентификаторы 2. соблюдение отступов при кодировании программ 3. совокупность правил записи, которым должна удовлетворять любая программа. 4. соблюдение регистра при записи операторов языка программирования
3.	Из указанных вариантов вычисления Y выполнится быстрее: 1. $X = 2*Y + (A-1)/P + 2*T$ 2. $X = 2*(Y + T) + (A-1)/P$ 3. $X = 2*Y + 2*T + A/P - 1/P$ 4. $X = 2*Y + (A-1)*1/P + 2*T$
4.	Ошибки выполнения в программе обнаруживаются: 1. на этапе компиляции 2. только на этапе выполнения 3. и во время компиляции, и во время выполнения 4. на этапе выполнения или тестирования
5.	Последовательность создания программных модулей: 1. исходный модуль, объектный модуль, загрузочный модуль 2. исходный модуль, загрузочный модуль, объектный модуль 3. загрузочный модуль, исходный модуль, объектный модуль 4. загрузочный модуль, объектный модуль, исходный модуль
6.	Отладка и тестирование программы находятся в соотношении: 1. это одно и то же 2. тестирование устанавливает факт наличия ошибки, отладка объясняет ее причину 3. отладка устанавливает факт наличия ошибки, тестирование объясняет ее причину

	4. отладка – расстановка в программе отладочной печати, тестирование – эксплуатация программы на производстве
7.	<p>Диапазон типа int:</p> <ol style="list-style-type: none"> 1. от 0 до 255 2. от -32767 до 32768 3. от -32768 до 32767 4. от 0 до 65535
8.	<p>Оператор сравнения в языке C++:</p> <ol style="list-style-type: none"> 1. = 2. <> 3. !=
9.	<p>Если функция вычисляет произведение двух чисел, и исходные данные вводятся с клавиатуры, то в программе целесообразно:</p> <ol style="list-style-type: none"> 1. ввести проверку, что исходные данные являются числами 2. не вводить проверки, все возможные ошибки выявит компилятор 3. проверить исходные данные на равенство нулю
10.	<p>В интегрированную среду программирования может входить:</p> <ol style="list-style-type: none"> 1. наладчик 2. доводчик 3. отладчик
11.	<p>Из наборов перечисляемых значений правильно записан:</p> <ol style="list-style-type: none"> 1. enum { a, b = 3, c = 4, 3 }; 2. enum { a, b, 3, 4 }; 3. enum { a, b = 3, c, d };+
12.	<p>Если в арифметическом выражении участвуют целый и вещественный операнды, то:</p> <ol style="list-style-type: none"> 1. произойдет ошибка компиляции 2. целый тип приводится к вещественному 3. вещественный тип приводится к целому
13.	<p>Истинное высказывание: переменная.....</p> <ol style="list-style-type: none"> 1. инициализируется, потом объявляется 2. изменяется, потом объявляется 3. объявляется, потом изменяется
14.	<p>Переменная типа signed char может принимать значения:</p> <ol style="list-style-type: none"> 1. только из алфавита языка C++ 2. только символов английского алфавита, цифр и символа подчеркивания 3. от -128 до 127
15.	<p>Числовое значение выражения $e/2*a-abs(e)*1e^0$ при $e = 4, a = 2$ равно:</p> <ol style="list-style-type: none"> 1. 3 2. 0 3. 1
16.	<p>Значение выражения $(a \ \&\& \ ! \ b \ \ c)$, где a, b и c – величины типа bool, имеющие значения false, true и true соответственно равно:</p> <ol style="list-style-type: none"> 1. false 2. yes 3. true
17.	<p>В языке C++ справа налево выполняется операция:</p> <ol style="list-style-type: none"> 1. = 2. ->

	3. []
18.	<p>Оператор выбора в языке C ++:</p> <ol style="list-style-type: none"> 1. CASE 2. choice 3. switch ... case 4. default
19.	<p>Разница между фактическими и формальными параметрами функций:</p> <ol style="list-style-type: none"> 1. формальные параметры могут использоваться только вне тела функции, а фактические — используются как вне функции, так и внутри нее 2. нет разницы, это одни и те же параметры 3. и формальные, и фактические параметры используются вне тела функции 4. формальные параметры определены в теле функции, а фактические – значение, с которыми функция вызывается.
20.	<p>Если значение переменной number не лежит между 3 и 6, то верная запись данного утверждения на языке C ++:</p> <ol style="list-style-type: none"> 1. number > 3 && number < 6 2. !(number < 3 && number < 6) 3. !(number < 6 number > 3) 4. number < 3 number > 6
21.	<p>Правильный вариант объявления константной переменной в C++, где type – тип данных; variable – имя переменной; value – константные значения:</p> <ol style="list-style-type: none"> 1. const type variable = value; 2. const variable = value; 3. type const variable = value;
22.	<p>Если в программе на языке C ++ есть два объявления переменных int qwerty; int QWERTY; то:</p> <ol style="list-style-type: none"> 1. такие имена переменных недопустимы 2. объявления правильные 3. такие объявления недопустимы, потому что мы пытаемся создать две переменные с одинаковыми идентификаторами 4. переменные описываются не по такому принципу
23.	<p>В рамках хорошего стиля программирования необходимы:</p> <ol style="list-style-type: none"> 1. мнемоника и выравнивание текста при кодировании программы 2. достаточно следовать правилам составления идентификаторов DOS 3. необходимо избегать комментариев
24.	<p>Если дан массив с именем A из целых чисел (1 2 3 4 5 6 7 8 9 10), то результат работы программы int S=0; for (int i=0; i<10: i++) S+=A[i]; равен:</p> <ol style="list-style-type: none"> 1. 46 2. 45 3. 56 4. 55
25.	<p>Группа операторов, которые решают логически самостоятельную часть задачи, называется:</p> <ol style="list-style-type: none"> 1. головной программой 2. подпрограммой 3. глобальными данными

26.	Графическое изображение логической структуры алгоритма называется: 1. блок-схемой 2. псевдокодом 3. словесным описанием алгоритма
27.	Исполняемыми называются операторы, которые: 1. содержатся в теле цикла 2. содержатся в теле условного оператора 3. задают действия над данными
28.	Линейным является алгоритм, в котором: 1. действия выполняются последовательно один раз и без проверки каких-либо условий 2. действия выполняются последовательно несколько раз 3. предусматриваются варианты действий в зависимости от изменения определенных условий
29.	Запоминающее устройство, из которого элементы извлекаются в порядке, обратном их помещению в запоминающее устройство, называется: 1. регистр 2. стек 3. оперативная память
30.	Массовость алгоритма предполагает: 1. точность указаний, исключающую их произвольное толкование 2. расчленение процесса на отдельные элементарные операции 3. решение алгоритмом целой группы родственных задач, отличающихся исходными данными
ВАРИАНТ 2	
1.	В рамках хорошего стиля программирования необходимы: 1. мнемоника и выравнивание текста при кодировании программы 2. достаточно следовать правилам составления идентификаторов DOS 3. необходимо избегать комментариев 4. необязательно следовать правилам составления идентификаторов DOS
2.	Оптимизация программы – это: 1. упрощение кода программы с целью экономии ОП 2. уменьшение количества решаемых задач в программе 3. построение эквивалентной программы, обладающей лучшими характеристиками времени работы и/или объема занимаемой ОП 4. строгое следование правилам составления идентификаторов DOS
3.	Для скорости выполнения вложенных циклов предпочтительнее реализация: 1. внешний цикл должен иметь меньше итераций по отношению к внутреннему 2. внутренний цикл должен иметь меньше итераций по отношению к внешнему 3. количество внутренних и внешних циклов должно быть одинаково
4.	Ошибки в алгоритме программы обнаруживаются: 1. на этапе компиляции 2. на этапе тестирования или эксплуатации 3. только на этапе тестирования 4. и во время компиляции, и во время выполнения
5.	Исходным называется модуль, который: 1. содержит последовательность машинных команд 2. является результатом работы компилятора 3. запускается для выполнения программы

	4. содержит текст программы на языке программирования
6.	В языке C++ не определен цикл: <ol style="list-style-type: none"> 1. for 2. while 3. repeat...until 4. do... while
7.	Тип int занимает в ОП: <ol style="list-style-type: none"> 1. 2 или 4 байта 2. 1 байт 3. 4 байта 4. 255 байт
8.	Оператор присваивания в языке C++: <ol style="list-style-type: none"> 1. = 2. <> 3. !=
9.	Оператор namespace предназначен для: <ol style="list-style-type: none"> 1. использования классов, переменных и функций из других модулей программы без использования заголовочных файлов 2. заключения в группу объявлений классов, переменных и функций в отдельный контекст со своим именем 3. заключения в группу объявлений классов, переменных и функций для использования только в текущем модуле
10.	В интегрированную среду программирования может входить текстовый: <ol style="list-style-type: none"> 1. редактор 2. директор 3. модератор
11.	При использовании следующих выражений #include <...> и #include «...»: <ol style="list-style-type: none"> 1. различие заключается в методе поиска препроцессором включаемого файла 2. в различии использования заголовочных и исходных файлов 3. нет различий
12.	Потеря точности произойдет в выражении: <ol style="list-style-type: none"> 1. short i = 0x3; float x = 2.7, v; v = i + x; 2. ж float M = 235.2; double Z = 3; Z *= M; 3. int i; float x = 2.134, y = 3.14; i = x/y;
13.	Переменная типа signed char может принимать значения: <ol style="list-style-type: none"> 1. только символов английского алфавита, цифр и символа подчеркивания 2. из первой половины кодовой таблицы 3. только из алфавита языка C++
14.	В переменной типа short можно хранить число: <ol style="list-style-type: none"> 1. -213 2. 213 + 3. 1213
15.	Правильное утверждение: целой..... <ol style="list-style-type: none"> 1. переменной можно присвоить вещественную константу 2. константе можно присвоить целую переменную 3. константе можно присвоить вещественную переменную
16.	Не содержат синтаксических ошибок выражение: <ol style="list-style-type: none"> 1. sin(abs(0.6(e*3))) 2. ((cos(3*a+1.*abs(x))))

	3. $a \cdot \exp(t) \cdot (2t)$
17.	Не содержит синтаксических ошибок выражение: 1. $(-0.18) \cdot a / (r - 0.2 \cdot t)$ 2. $(-0.18) \cdot a \cdot (r - 0.2 \cdot t)$ 3. $-0.18 \cdot a / r - 0.2 \cdot t$
18.	Правильно применяется операция в целочисленных типах (язык C++) в выражении. 1. <code>int a = 2, b = 0x24, c; c = a + b % a;</code> 2. <code>short x = 0, y = 2; for (y != x) { ... };</code> 3. <code>int c <> 2;</code> 4. <code>const char mychar = 2; long s = 3; mychar / = s;</code>
19.	Объект cout определен в файле заголовков: 1. <code>stream.h</code> 2. <code>sysutils.hpp</code> 3. <code>cout.h</code> 4. <code>iostream.h</code>
20.	Строки в C++ представляются как массивы элементов типа char, заканчивающиеся символом: 1. '0' 2. '\0' 3. » 4. '.'
21.	В C++ быстрее будет вычисляться выражение: 1. <code>X = X + Y</code> 2. <code>X += Y</code> 3. <code>X = Y + X</code> 4. все три выражения будут выполнены с одинаковой скоростью
22.	Двумерный массив в C++ объявлен верно в фрагменте: 1. <code>int anarray [20] [20];</code> 2. <code>int array [20, 20];</code> 3. <code>array anarray [20] [20];</code>
23.	Оптимизация программы – это: 1. упрощение кода программы с целью экономии ОП 2. уменьшение количества решаемых задач в программе 3. построение эквивалентной программы, обладающей лучшими характеристиками времени работы и/или объема занимаемой ОП
24.	Если дан массив с именем A из целых чисел (1 2 3 4 5 6 7 8 9 10), то результат работы программы <code>int S=10; for (int i=0; i<10: i++) S+=A[i];</code> равен: 1. 46 2. 45 3. 55 4. 65
25.	Величина, не изменяющаяся в процессе выполнения приложения, называется: 1. константой 2. переменной 3. символом
26.	Единица текста программы, которая при компиляции воспринимается как единое целое и по смыслу не может быть разделена на более мелкие элементы, называется: 1. синтаксической единицей 2. лексемой

	3. словом
27.	Конструкция «цикл пока» выполняется, если: 1. условие истинно 2. условие ложно 3. цикл выполнялся столько раз, сколько указано в его заголовке
28.	Разветвляющимся называется алгоритм, в котором: 1. действия выполняются последовательно один раз и без проверки каких-либо условий 2. действия выполняются последовательно несколько раз 3. предусматриваются варианты действий в зависимости от изменения определенных условий
29.	Языки программирования делятся на языки высокого и.... 1. низкого уровня 2. физического уровня 3. среднего уровня
30.	Дискретность алгоритма предполагает: 1. точность указаний, исключающую их произвольное толкование 2. расчленение процесса на отдельные элементарные операции 3. решение алгоритмом целой группы родственных задач, отличающихся исходными данными
ВАРИАНТ 3	
1.	Алфавит языка программирования: 1. все команды языка 2. набор символов, включающий буквы, символы и спец. знаки 3. компилятор языка 4. латинский и русский алфавит
2.	Из следующих одинаковых операции выполняются быстрее: 1. $3*I$ 2. $I + I + I$ 3. $2*I+I$ 4. $I+2*I$
3.	Синтаксические ошибки в программе обнаруживаются: 1. на этапе компиляции 2. на этапе выполнения 3. и во время компиляции, и во время выполнения 4. на этапе тестирования
4.	Если программа выдает неверные результаты, необходимо: 1. расставить отладочную печать 2. сразу же внести изменения в текст программы 3. сразу же переделать алгоритм
5.	На языке C++ переменные объявляются: 1. только в начале программы 2. только в конце программы 3. в специально подключаемом файле 4. в любом месте программы, но обязательно перед использованием переменной
6.	К языку C++ не относится тип данных: 1. float 2. real 3. short

	4. char
7.	<p>Оператор сравнения в языке C++:</p> <ol style="list-style-type: none"> 1. = 2. <> 3. ==
8.	<p>В языке C++ отсутствует оператор:</p> <ol style="list-style-type: none"> 1. = 2. <> 3. != 4. ==
9.	<p>Правильный вариант выделения динамической памяти под переменную X типа float:</p> <ol style="list-style-type: none"> 1. float *ptr = new float; X = *ptr; 2. float & ptr = new float; X = & ptr; 3. float * ptr = &X; X = new float;
10.	<p>В интегрированную среду программирования может входить:</p> <ol style="list-style-type: none"> 1. регулятор 2. доминатор 3. компилятор
11.	<p>Результат вычисления выражения: int d=5; bool b = true, c; c = (!b (d>3)) будет равен:</p> <ol style="list-style-type: none"> 1. ошибке компилятора 2. false 3. true
12.	<p>Если после выражения стоит точка с запятой, то:</p> <ol style="list-style-type: none"> 1. выражение вычисляется, а его значение запоминается в специальной переменной, которую можно использовать в следующем операторе 2. это оператор-выражение, действие которого заключается в вычислении выражения 3. выражение вычисляется только если первой стоит операция присваивания
13.	<p>Переменная типа signed char может принимать значения:</p> <ol style="list-style-type: none"> 1. только символов английского алфавита, цифр и символа подчеркивания 2. из первой половины кодовой таблицы 3. только из алфавита языка C++
14.	<p>В переменной типа unsigned char можно хранить число:</p> <ol style="list-style-type: none"> 1. -13 2. 1213 3. 13
15.	<p>Правильное утверждение: целой.....</p> <ol style="list-style-type: none"> 1. константе можно присвоить целую переменную 2. переменной можно присвоить целую константу 3. константе можно присвоить вещественную переменную
16.	<p>Не содержат синтаксических ошибок выражение:</p> <ol style="list-style-type: none"> 1. a*exp(t)\(2t) 2. sin(abs(0.6(e*3)) 3. 0XCC00*.34E-4/_do/k-2
17.	<p>Значение выражения sqrt(4)+142/20*2 равно:</p> <ol style="list-style-type: none"> 1. 5 2. 16 3. 9

18.	В сложных выражениях последовательность выполнения операций определяется: 1. только приоритетом операций 2. только скобками 3. скобками, приоритетом операций, а при одинаковом приоритете ассоциативностью операций 4. только ассоциативностью операций.
19.	Оператор continue в языке C++ выполняется следующим образом: 1. пропускает остаток тела цикла и переходит к следующей итерации 2. пропускает цикл и переходит к следующему оператору в теле программы 3. определяет условие продолжения цикла 4. продолжает выполнение текущей итерации цикла
20.	Операция ++ в языке C++: 1. уменьшает значение операнда на единицу 2. уменьшает значение операнда на два 3. увеличивает значение операнда на два 4. увеличивает значение операнда на единицу
21.	Неправильно применяются операции с действительными типами в выражении: 1. float a = 2, b = 0x24, c; c = a + b% a; 2. float x = 0, y = 2; и f (b >= x) {...}; 3. long double c = 2; 4. double mychar = 2; float s = 3; mychar /= s;
22.	Одномерный массив в C++ объявлен верно в фрагменте: 1. int array [20, 20]; 2. array anarray [20]; 3. int anarray [20];
23.	Жизненный цикл программного продукта – это интервал времени от момента: 1. возникновения объективной необходимости в программе до момента изъятия ее из эксплуатации 2. написания программы до момента ввода ее в эксплуатацию 3. ввода программы в эксплуатацию до возникновения в ней ошибки в процессе работы
24.	Если дан массив с именем A из целых чисел (1 2 3 4 5 6 7 8 9 10), то результат работы программы for (int i=0; i<10: i++) S+=A[i]; равен: 1. 46 2. 45 3. 56 4. непредсказуемому значению
25.	Величина, принимающая в процессе выполнения приложения различные значения, называется: 1. константой 2. переменной 3. символом
26.	Если некоторая часть программы выполняется многократно и после проверки некоторого условия осуществляется выход из нее, то такую часть называют: 1. оператором выбора 2. циклом 3. условным оператором

27.	Имеющая единое имя совокупность однотипных элементов – это: 1. подпрограмма 2. строка 3. массив
28.	Последовательность действий, которая выполняется многократно, называется телом: 1. цикла 2. оператора выбора 3. подпрограммы
29.	Языки, основанные на построении объектов, объединяющих данные и операции над ними, называют: 1. процедурными 2. объектно ориентированными 3. структурными
30.	Определенность алгоритма предполагает: 1. точность указаний, исключающую их произвольное толкование 2. расчленение процесса на отдельные элементарные операции 3. решение алгоритмом целой группы родственных задач, отличающихся исходными данными

ТЕМЫ И ОБРАЗЦЫ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ (семестр 3)

Тема **ФОРМЫ**

Создать файлы по предложенным примерам:

№1. Форма с полями для ввода текста

```
< Html ><HEAD><title>Пример ввода текста</title></head>
<body text="#000000" bgcolor="#D6FDD5">
<form>
<center><table BORDER=0 WIDTH="96%" >
<tr><td><div align=right>Имя:</div></td>
<td><input type="text" maxlength=10 name="firstname" size=10></td></tr>
<tr><td><div align=right>Фамилия:</div></td>
<td><input type="text" maxlength=20 name="lastname" size=20></td></tr>
</table></center>
</form>
</body></html>
```

№2. Использование кнопки для очистки поля ввода текста

```
<html><head><title>Пример очистки формы для элемента ввода тек-
ста</title></head>
<body text="#000000" bgcolor="#D6FDD5">
<form>
<center><table BORDER=0 WIDTH="96%" >
<tr><td><div align=right>Страна :</div></td>
<td><input type="text" maxlength=20 name="firstname" size=20></td>
<td><div align=right>Город :</div></td>
<td><input type="text" maxlength=20 name="lastname" size=20></td></tr>
</table>
<INPUT type="reset" value="Очистить форму"></center>
</form></body></html>
```

№3. Чекбоксы

```
<html><head><title> Пример чекбоксов</title></head>
<body text="#000000" bgcolor="#D6FDD5">
<form>
<center><table WIDTH="96%" BORDER="0" >
<tr><td><div align=right>Какие книги Вам нужны:</div>
</td><td><input type="checkbox" name="ch1" value="ich1"> Книга 1
<br><input type="checkbox" name="ch2" value="ich2"> Книга 2
<br><input type="checkbox" name="ch3" value="ich3"> Книга 3</td>
</tr></table></center>
</form></body></html>
```

№4. Меню выбора из выпадающего списка

```
<html><head><title>Пример выбора из списка</title></head>
<body text="#000000" bgcolor="#D6FDD5">
```

```
<center><form><b>Шахматы</b>
<br>Какими фигурами предпочитаете играть?&nbsp;
<select size=1 name="color"><option>белыми</option><option>черными</option></select>
</form></center></body></html>
```

№5. Радиокнопки

```
<html><head><title>Примеры радиокнопок</title></head>
<body text="#000000" bgcolor="#D6FDD5">
<form>
<H2>какой язык вы используете?
<UL>
<input type="RADIO" name="LANG" value="Русский" checked>Русский<br>
<input type="RADIO" name="LANG" value="Английский">Английский<br>
<input type="RADIO" name="LANG" value="Немецкий">Немецкий</tr>
</UL>
</H2>
</form></body></html>
```

Тема ТАБЛИЦЫ CSS

1. Создайте файл *style.css*

Подключите его к своим страницам (см. способ 3 задания стилей).

2. Задайте по умолчанию следующие параметры для всех страниц (переопределив тег *<body>*):

- цвет фона;
- размер шрифта;
- цвет шрифта;
- семейство шрифта (например, *Arial*).

3. В комментариях (в файле *style.css*) поясните эти параметры.

Тема ЯЗЫК JAVASCRIPT

1. **Вычисление площади треугольника.** Написать сценарий, определяющий площадь прямоугольного треугольника по заданным катетам.

2. **Вычисление площади квадрата.** Написать сценарий, определяющий площадь квадрата по заданной стороне.

3. **Нахождение максимального значения.** Для трех заданных значений *a*, *b*, *c* написать сценарий, определяющий максимальное значение.

4. Вводится последовательность из пяти чисел. Напишите сценарий, в котором определяется число максимальных элементов.

5. **Простое вертикальное меню.** Написать сценарий, реализующий вертикальное графическое меню. При наведении курсора мыши на пункт меню меняется цветовая палитра, соответствующая выделенному пункту меню.

6. Разработайте анкету, определяющую пол, возраст, семейное положение и т.п., человека.

7. Написать сценарий, который позволяет выбрать для таблицы и составляющих ее ячеек либо цвет фона, либо фоновое изображение, либо и то и другое.

ТЕМЫ И ОБРАЗЦЫ ЗАДАНИЙ ДЛЯ ЛАБОРАТОРНЫХ ЗАНЯТИЙ (семестр 4)

Тема *Линейные программы*

1. Программа, переводящей температуру по Фаренгейту в температуру по Цельсию:

```
#include <iostream.h>
int main()
{
float frg,cls;
cout<<"Введите температуру по Фаренгейту "; cin>>frg;
cls=5/9*(frg-32);
cout<<"Температура по Цельсию ";<<cls;
return 0;
}
```

2. Напишите программу для расчета по двум формулам (результат вычисления по первой формуле должен совпадать со второй).

Вариант 1

$$z_1 = \cos \alpha + \sin \alpha + \cos 3\alpha + \sin 3\alpha$$

$$z_2 = 2\sqrt{2} \cos \alpha \cdot \sin \left(\frac{\pi}{4} + 2\alpha \right)$$

Вариант 2

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2\sin^2 2\alpha}$$

$$z_2 = 2\sin \alpha$$

Вариант 3

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos \alpha}$$

$$z_2 = \operatorname{tg} 3\alpha$$

Вариант 4

$$z_1 = \frac{\sqrt{2b + 2\sqrt{b^2 - 4}}}{\sqrt{b^2 - 4} + b + 2}$$

$$z_2 = \frac{1}{\sqrt{b + 2}}$$

Тема *Разветвляющиеся программы*

Написать программу, которая по введенному значению аргумента вычисляет значение функции

Перед написанием программы следует составить алгоритм ее решения в словесном виде или в виде блок-схемы. Опишем алгоритм нашей задачи в словесной форме:

1. ввести значение аргумента x ;
2. определить, к какому интервалу оно принадлежит;
3. вычислить значение функции по соответствующей формуле;
4. вывести значение y .

$$y = \begin{cases} 0, & x < -2 \\ -x - 2, & -2 \leq x < -1 \\ x, & -1 \leq x < 1 \\ -x + 2, & 1 \leq x < 2 \\ 0, & x \geq 2 \end{cases}$$

Запрограммировать решение квадратных уравнений $ax^2+bx+c=0$, когда заранее неизвестно их количество. Выход из цикла сделать при условии, когда коэффициент $a=0$ (тогда квадратное уравнение вырождается в линейное). Алгоритм решения этой задачи следующий:

1. ввести с клавиатуры коэффициент a ;
2. проверим его равенство нулю;
3. если коэффициент a равен нулю, – выход из цикла;
4. если коэффициент a отличен от нуля, ввести с клавиатуры коэффициенты b, c ;
5. вычислить дискриминант;
6. если дискриминант меньше нуля – нет решений, если дискриминант равен нулю – уравнение имеет одно решение, если дискриминант больше нуля – уравнение имеет два решения;
7. вывести корни или соответствующее сообщение.

Написать программу печати таблицы значений функции $y=x^2$ для аргумента, изменяющегося в заданных пределах с заданным шагом. Исходными данными являются: начальное значение аргумента x_n , конечное значение аргумента x_k , шаг изменения аргумента dx . Все величины вещественные. Программа должна выводить таблицу, состоящую из двух столбцов – значений аргумента и соответствующих им значений функции.

Алгоритм написания программы:

1. ввести исходные данные;
2. вычислить значение функции по формуле;
3. вывести строку таблицы;
4. перейти к следующему значению аргумента;
5. если оно не превышает конечное значение, повторить шаги 2-6, иначе закончить выполнение;

Шаги 2-5 повторяются многократно, поэтому для их выполнения надо организовать цикл.

Написать программу печати таблицы значений функции.

$$y = \begin{cases} t, & x < 0 \\ tx, & 0 \leq x < 10 \\ 2t, & x \geq 10 \end{cases}$$

для аргумента, изменяющегося в заданных пределах с заданным шагом. Если $t > 0$, должны выводиться целые значения функции. Исходными данными являются: начальное значение аргумента x_n , конечное значение аргумента x_k , шаг изменения аргумента dx и параметр t . Все величины вещественные. Программа должна выводить таблицу, состоящую из двух столбцов – значений аргумента и соответствующих им значений функции.

Алгоритм написания программы:

1. Ввести исходные данные.
2. Определить, к какому интервалу принадлежит значение аргумента.
3. Вычислить значение функции по соответствующей формуле.
4. Если $t > 0$, преобразовать значение y в целое.
5. Вывести строку таблицы.
6. Перейти к следующему значению аргумента.
7. Если оно не превышает конечное значение, повторить шаги 2-6, иначе закончить выполнение.

Шаги 2-6 повторяются многократно, поэтому для их выполнения надо организовать цикл.

Вычислить и вывести на экран значение функции y .

Вариант 1.

$$y = \begin{cases} \prod_{k=2}^{N+2} \sin \sqrt{k^2 - 1}, & N \geq 5 \\ \sum_{i=1}^N \cos i, & N < 5 \end{cases}$$

Вариант 2.

$$y = \begin{cases} \prod_{k=2}^{N+2} \sin \sqrt{k^2 - 1} \times \sum_{j=1}^{N-1} (j+1), & N \leq 10 \\ 0, & N > 10 \end{cases}$$

Вариант 3.

$$y = \begin{cases} \prod_{k=1}^{N+2} \cos \sqrt{k} \times \sin N, & N \leq 10 \\ \sum_{l=2}^{N+10} l^3, & N > 10 \end{cases}$$

Вариант 4.

$$y = \begin{cases} \prod_{k=1}^{N+2} \cos \sqrt{k} \times \sum_{k=1}^N (k+1), & N \leq 10 \\ \sum_{l=2}^{N+10} \sqrt{l}, & N > 10 \end{cases}$$

Тема работы: Массивы.

Пример №8.

Написать программу, которая для целочисленного массива из 100 элементов определяет, сколько положительных элементов располагается между его минимальным и максимальным элементами.

Запишем алгоритм задачи. Порядок расположения элементов в массиве неизвестен, и сначала может следовать как минимальный, так и максимальный элемент, кроме того, они могут совпадать.

1. определить, где в массиве расположены минимальный и максимальный элементы (задать начальные значения для индексов этих элементов, потом просмотреть массив, поочередно сравнивая каждый его элемент с найденным ранее максимумом и минимумом);
2. определить границы просмотра массива для поиска положительных элементов между минимумом и максимумом;
3. просмотреть массив, увеличивая счетчик положительных элементов (если таковые встретятся).

При написании программы рекомендуется использовать отладочную печать. Для большей наглядности здесь элементы массива задаются путем инициализации.

Вариант 2.

В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму положительных элементов массива, произведение элементов массива с четными номерами. Преобразовать массив таким образом, чтобы сначала располагались все положительные элементы, а потом все отрицательные.

Вариант 3.

В одномерном массиве, состоящем из n целых элементов, вычислить произведение элементов массива с четными номерами, сумму элементов массива, расположенных между первым и последним нулевыми элементами. Упорядочить элементы массива по убыванию.

Вариант 4.

В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму элементов массива с нечетными номерами, сумму элементов массива, расположенных между первым и последним отрицательными элементами. Сжать массив, удалив из него все элементы, модуль которых не превышает 1, освободившиеся в конце массива элементы заполнить нулями.

ВОПРОСЫ К ЭКЗАМЕНУ

1. Хороший стиль программирования.
2. Оптимизация программ.
3. Создание и инициализация одномерных массивов.
4. Двумерные массивы. Объявление, типы, методы работы.
5. Методы сортировки массивов.
6. Поиск минимального и максимального элементов.
7. Файлы. Чтение и запись информации. Пример.
8. Создание цикла при чтении информации из файла. Создание цикла при записи информации в файл. Пример.
9. Файлы. Функции позиционирования. Пример.
10. Динамические массивы. Объявление, методы работы.
11. Выделение и освобождение динамической памяти. Подпрограммы работы с динамической памятью.
12. Понятие указателя. Базовый тип указателя. Пример.
13. Присваивание значений с помощью указателя. Пример.
14. Арифметические операции над указателями. Пример.
15. Указатели и массивы. Имя массива как указатель.
16. Соглашение о нулевых указателях.
17. Массивы указателей. Пример.
18. Функции. Передача аргумента по значению и по ссылке.
19. Функции. Использование указателя для обеспечения вызова по ссылке.
20. Функции. Ссылочные параметры. Возврат ссылок
21. Функции. Аргументы по умолчанию.
22. Понятие перегрузки функций. Тип и количество параметров при перегрузке.
23. Перегрузка функций и неоднозначность.
24. Векторы. Объявление и заполнение векторов.
25. Векторы. Цикл, основанный на диапазоне.
26. Векторы. Итераторы.
27. Свойства объектно-ориентированного программирования (ООП).
28. Понятие объекта в ООП.

ПРИМЕРЫ К ЭКЗАМЕНУ

1. В одномерном массиве, состоящем из n целочисленных элементов, вычислить:
 - минимальный элемент массива и его индекс;
 - максимальный элемент массива и его индекс.

Указание: рассмотреть случай однородного массива.

2. Написать программу печати таблицы значений функции

$$y = \begin{cases} t, & x < 0 \\ tx, & 0 \leq x < 10 \\ 2t, & x \geq 10 \end{cases}$$

для аргумента, изменяющегося в заданных пределах с заданным шагом (с экрана ввести x начальное, x конечное, шаг dx и t).

3. Доказать тождество выражений

$$z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2\sin^2 2\alpha}$$

$$z_2 = 2\sin \alpha$$

4. Используя операторы *if*, *for* составить программу для расчета функции:

$$y = \begin{cases} \prod_{k=2}^{N+2} (k-1) + \sum_{i=1}^N i^2, & 0 < N < 5 \\ \sum_{m=1}^N \sqrt{m}, & N > 5 \end{cases}$$

5. Написать программу, которая для целочисленного массива из n элементов определяет сумму его элементов, расположенных правее последнего отрицательного числа.

Указание: рассмотреть случай отсутствия в массиве отрицательных чисел.

6. Получить с экрана 2 строки, вычислить длину каждой строки, объединить их. Полученную строку вывести на экран.

7. Объявить строку, присвоить ей значение и вычислить длину. Строку и ее длину вывести на экран.

ПРИМЕРНЫЕ ТЕМЫ КУРСОВЫХ РАБОТ

1. Пользовательские типы данных
2. Указатели, ссылки и динамические структуры данных.
3. Модульное программирование.
4. Объекты и классы. Наследование классов.
5. Полиморфизм и виртуальные методы.
6. Обработка исключительных ситуаций. Обработка сообщений.
7. Графический интерфейс пользователя.
8. Назначение и состав интегрированных средств разработки.
9. Разработка калькулятора.
10. Алгоритм поиска элемента в массиве.
11. Создание сайта учебной группы на языке HTML.
12. Формы и программирование CGI.
13. Мониторинг локальных вычислительных сетей.
14. Транспьютеры. Архитектура и концепция.
15. Методы сортировок массивов.
16. CASE-технологии. Общая методология.
17. Технология внедрения CASE-средств.
18. Сравнительный анализ методов сортировки.
19. Методы тестирования и отладки программ.
20. Сравнительная характеристика использования фреймов и таблиц в создании сайтов.
21. Аналитическое программирование Дейкстры.
22. Языки функционального программирования.
23. Параллельные алгоритмы. Моделирование параллельных программ.
24. Защита данных. Ограничение доступа к данным SQL SERVER.
25. Система безопасности SQL. Режимы аутентификации.
26. Оценка эффективности программы. Составляющие эффективности.
27. Оценка сложности алгоритма. Правила определения сложности.
28. Создание сайтов. Использование средств Java Script и CSS.
29. Динамические структуры. Стеки, очереди.
30. Динамические структуры. Графы, деревья.

Студент может предложить собственную тему.

Приложение № 5

ЗАДАНИЯ ДЛЯ ЗАЩИТЫ ЛАБОРАТОРНЫХ РАБОТ

Задание 1

Создать главную страницу, на которой разместить название учебного заведения, номер группы, название специальности. Вставить графический рисунок. Задать фон и цвет текста. В заголовке HTML-документа указать номер группы и свою фамилию.

Создать в своей папке страницу расписания занятий своей группы. В заголовке HTML-документа указать номер группы и свою фамилию. Задать фон. Разные цвета и начертания шрифтов в заголовке и списках. Например, фон голубой, заголовок синий, дни недели – красные, дисциплины – синие.

Связать созданные документы с помощью ссылок

Ответить на контрольные вопросы.

Задание 2

Создать главную страницу, состоящую из 3 фреймов. В заголовке HTML-документа указать номер группы и свою фамилию.

Создать страницу с таблицей «Преподаватели».

Создать страницу «Новости».


Создать страницу «О нас».

На странице «Расписание» вынести список дней недели вверх и организовать ссылки на расписания конкретных дней недели (ссылки внутри документа).

Организовать переход на страницы «Расписание», «Преподаватели», «Новости», «О нас» с главной страницы.

Образец страницы:

Первая страница:

	<p>Институт "Калининградская Школа №2" Группа №__ "Автоматизированные системы обработки информации и управления"</p>
<ul style="list-style-type: none">○ О нас○ Расписание○ Преподаватели○ Новости <p>Примечание: это гиперссылки на отдельные страницы</p>	<p>Примечание: эта часть окна обозревателя должна быть предназначена для отображения отдельных страниц,, запускаемых из ссылок левого фрейма</p>

Задание 3

Создать форму с текстовыми полями, выпадающим меню, радиокнопками, чекбоксами. Интерфейс взять у преподавателя.