

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Алейник Станислав Николаевич
Должность: Ректор
Дата подписания: 08.04.2021 18:01:19
Уникальный программный ключ:
5258223550ea9fbeb23726a1609b644b37d8986ab6255891f388f917a1751fae

МИНИСТЕРСТВО СЕЛЬСКОГО ХОЗЯЙСТВА РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ АГРАРНЫЙ УНИВЕРСИТЕТ
имени В.Я.ГОРИНА

«УТВЕРЖДАЮ»
Декан инженерного факультета,
С.В. Стребков
«08» 07 2018 г.

РАБОЧАЯ ПРОГРАММА

по дисциплине **«Разработка программных приложений»**

Направление 09.03.03 Прикладная информатика

Направленность (профиль) - Прикладная информатика в АПК

Квалификация – бакалавр

Майский, 2018

Рабочая программа составлена с учетом требований:

- федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.03 «Прикладная информатика», утвержденного приказом Министерства образования и науки РФ от 12 марта 2015 г. № 207;
- порядка организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры, утвержденного приказом Министерства образования и науки РФ от 05.04.2017 г. № 301;
- основной профессиональной образовательной программы ФГБОУ ВО Белгородский ГАУ по направлению подготовки 09.03.03 «Прикладная информатика»

Составитель: канд.техн.наук, доцент Игнатенко В.А.

Рассмотрена на заседании кафедры информатики и информационных технологий от *21.06.* 2018 г., протокол № *13*

и.о. зав. кафедрой



Игнатенко В.А.

Одобрена методической комиссией инженерного факультета от *05.07.* 2018 г., протокол № *9-17/18*

Председатель методической комиссии



Слободюк А.П.

1. ЦЕЛЬ И ЗАДАЧИ ДИСЦИПЛИНЫ

1.1. Цель дисциплины – обучить студентов созданию, отладке и тестированию программных приложений в интегрированной среде разработки.

1.2. Задачи:

- совершенствовать знания объектно-ориентированного и визуального программирования, алгоритмов компьютерной обработки структур данных, а также технологии программирования;
- развить профессиональные компетенции, включая технологию разработки программного обеспечения на языках высокого уровня.

II. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОСНОВНОЙ ОБРАЗОВАТЕЛЬНОЙ ПРОГРАММЫ (ОПОП)

2.1. Цикл (раздел) ОПОП, к которому относится дисциплина

Дисциплина «Разработка программных приложений» относится к дисциплинам вариативной части (Б1.В.ДВ.09.01) основной образовательной программы.

2.2. Логическая взаимосвязь с другими частями ОПОП

| | |
|--|--|
| Наименование предшествующих дисциплин, практик, на которых базируется данная дисциплина (модуль) | 1. Математика |
| | 2. Информатика и программирование |
| | 3. Информационные системы и технологии |
| Требования к предварительной подготовке обучающихся | <p><i>знать:</i></p> <ul style="list-style-type: none"> • основы теории информации; • основные программные конструкции; • основы концепций функционального и объектно-ориентированного программирования. <p><i>уметь:</i></p> <ul style="list-style-type: none"> • реализовывать программные алгоритмы на языке программирования PASCAL, Си. <p><i>владеть:</i></p> <ul style="list-style-type: none"> • Интегрированными средами программирования и отладки (IDE); |

Дисциплина является предшествующей для дисциплин: «Проектный практикум», «Программирование информационных систем».

**III. ОБРАЗОВАТЕЛЬНЫЕ РЕЗУЛЬТАТЫ ОСВОЕНИЯ
ДИСЦИПЛИНЫ, СООТВЕТСТВУЮЩИЕ ФОРМИРУЕМЫМ КОМПЕТЕНЦИЯМ**

| Коды компетенций | Формулировка компетенции | Планируемые результаты обучения по дисциплине |
|------------------|--|--|
| ПК-8 | способностью программировать приложения и создавать программные прототипы решения прикладных задач | Знать: <ul style="list-style-type: none"> • объектно-ориентированную парадигму языков программирования; • свойства и методы элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. |
| | | Уметь: <ul style="list-style-type: none"> • проектировать Windows-интерфейс приложения; • использовать приемы визуального программирования в сочетании с разработкой программного кода; • использовать современные готовые библиотеки классов. |
| | | Владеть: <ul style="list-style-type: none"> • практическими приемами алгоритмизации, разработки, отладки и тестирования программ в различных интегрированных средах на объектно-ориентированных языках. |
| ПК-12 | способностью проводить тестирование компонентов программного обеспечения ИС | Знать: <ul style="list-style-type: none"> • основные этапы реализации проектирования программ. |
| | | Уметь: <ul style="list-style-type: none"> • формализовать поставленную задачу; • тестировать и отлаживать программы в современных интегрированных средах разработки. |
| | | Владеть: <ul style="list-style-type: none"> • навыками алгоритмизации; приемами разработки, отладки и тестирования приложений. |
| ПК-14 | способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач | Знать: <ul style="list-style-type: none"> • основные способы и принципы представления абстрактных объектов данных; • возможности интегрированных сред разработки. |
| | | Уметь: <ul style="list-style-type: none"> • применять полученные знания для разработки прикладного программного обеспечения. |
| | | Владеть: <ul style="list-style-type: none"> • методиками разработки программных приложений для обработки данных. |

IV. ОБЪЕМ, СТРУКТУРА, СОДЕРЖАНИЕ ДИСЦИПЛИНЫ, ВИДЫ УЧЕБНОЙ РАБОТЫ И ФОРМЫ КОНТРОЛЯ ЗНАНИЙ

4.1. Распределение объема учебной работы по формам обучения

| Вид работы | Объем учебной работы, час | |
|---|-----------------------------|----------------|
| | Очная | Заочная |
| Формы обучения (вносятся данные по реализуемым формам) | Очная | Заочная |
| Семестр (курс) изучения дисциплины | 4 семестр 2 курс | 3 курс |
| Общая трудоемкость, всего, час | 144 | 144 |
| <i>зачетные единицы</i> | 4 | 4 |
| Контактная работа обучающихся с преподавателем | 40 | 14 |
| Аудиторные занятия (всего) | 40 | 14 |
| В том числе: | | |
| Лекции | 20 | 6 |
| Лабораторные занятия | 20 | 8 |
| Практические занятия | - | - |
| <i>Иные виды работ в соответствии с учебным планом (учебная практика)</i> | - | - |
| Внеаудиторная работа (всего) | 20 | 6 |
| В том числе: | | |
| Контроль самостоятельной работы (на 1 подгруппу в форме компьютерного тестирования) | - | - |
| Консультации согласно графику кафедры | 20 | 6 |
| <i>Иные виды работ в соответствии с учебным планом (курсовая работа, РГЗ и др.)</i> | - | - |
| Промежуточная аттестация | 4 | 4 |
| В том числе: | | |
| Зачет | 4 | 4 |
| Экзамен (на 1 группу) | - | - |
| Консультация предэкзаменационная (на 1 группу) | - | - |
| Самостоятельная работа обучающихся | 80 | 120 |
| Самостоятельная работа обучающихся (всего) | 80 | 120 |
| в том числе: | | |
| Самостоятельная работа по проработке лекционного материала (60% от объема лекций) | 12 | 4 |
| Самостоятельная работа по подготовке к лабораторно-практическим занятиям (60% от объема аудиторных занятий) | 12 | 5 |
| Работа над темами (вопросами), вынесенными на самостоятельное изучение | 46 | 91 |
| Самостоятельная работа по видам индивидуальных заданий : подготовка реферата (контрольной работы) | 10 | 20 |

Примечание: *осуществляется на аудиторных занятиях

4.2 Общая структура дисциплины и виды учебной работы

| Наименование модулей и разделов дисциплины | Объемы видов учебной работы по формам обучения, час | | | | | | | | | |
|--|---|-----------|----------------------------|---------------------------------|------------------------|------------------------|----------|----------------------------|---------------------------------|------------------------|
| | Очная форма обучения | | | | | Заочная форма обучения | | | | |
| | Всего | Лекции | Лабораторно-практ. занятия | Внеаудиторная работа и пр. атт. | Самостоятельная работа | Всего | Лекции | Лабораторно-практ. занятия | Внеаудиторная работа и пр. атт. | Самостоятельная работа |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Модуль 1. | 65 | 10 | 10 | 10 | 35 | 58 | 2 | 4 | 2 | 50 |
| 1. Введение в язык Си. Структура программы. | 7 | 2 | - | Консультации | 5 | 10 | - | - | Консультации | 10 |
| 2. Объекты языка Си и их типы. Простые объекты. | 11 | 2 | 2 | | 7 | 12 | 2 | - | | 10 |
| 3. Операции. Ввод и вывод информации. Операторы. | 11 | 2 | 2 | | 7 | 12 | - | 2 | | 10 |
| 4. Функции. Прототипы функций. | 11 | 2 | 2 | | 7 | 12 | - | 2 | | 10 |
| 5. Препроцессор. Глобальные и локальные объекты. Математические функции. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| <i>Итоговое занятие по модулю 1</i> | 4 | - | 2 | | 2 | - | - | - | | - |
| Модуль 2. | 65 | 10 | 10 | 10 | 35 | 62 | 4 | 4 | 4 | 50 |
| 1. Указатели. Модели памяти. Динамическое распределение памяти. | 9 | 2 | 2 | Консультации | 5 | 10 | - | - | Консультации | 10 |
| 2. Массивы. Передача массива в функцию. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| 3. Динамическое распределение памяти под массивы | 11 | 2 | 2 | | 7 | 14 | 2 | 2 | | 10 |
| 4. Структуры. Объединения. Битовые поля. | 9 | 2 | - | | 7 | 14 | 2 | 2 | | 10 |
| 5. Функции ввода-вывода. Работа с файлами данных. Работа со строками. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| <i>Итоговое занятие по модулю 2</i> | 4 | - | 2 | | 2 | - | - | - | | - |
| Подготовка реферата в форме презентации (контрольной работы) | 10 | - | - | - | 10 | 20 | - | - | - | 20 |
| Зачёт | 4 | - | - | 4 | - | 4 | - | - | 4 | - |

4.3 Структура и содержание дисциплины по формам обучения

| Наименование модулей и разделов дисциплины | Объемы видов учебной работы по формам обучения, час | | | | | | | | | |
|---|---|-----------|-----------------|------------------|----------------|------------------------|----------|-----------------|------------------|----------------|
| | Очная форма обучения | | | | | Заочная форма обучения | | | | |
| | Всего | Лекции | Лаб.практ. зан. | Внеаудит. работа | Самост. работа | Всего | Лекции | Лаб.практ. зан. | Внеаудит. работа | Самост. работа |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Модуль 1. | 65 | 10 | 10 | 10 | 35 | 58 | 2 | 4 | 2 | 50 |
| 1. Введение в язык Си. Структура программы. | 7 | 2 | - | Консультации | 5 | 10 | - | - | Консультации | 10 |
| 1.1. Введение в концепцию .NET. Языки программирования. Стандартная система типов. Пространства имен. Основные типы приложений. Элементы интерфейса среды разработки приложений. Проекты и задачи. Средства отладки. | 7 | 2 | - | | 5 | 10 | - | - | | 10 |
| 2. Объекты языка Си и их типы. Простые объекты. | 11 | 2 | 2 | | 7 | 12 | 2 | - | | 10 |
| 2.1. Создание проектов Windows.Forms. Использование шаблонов. Основные классы. Свойства и методы классов. Структура классов Application, Component, Control, Form. Использование в пользовательских проектах. | 11 | 2 | 2 | | 7 | 12 | 2 | - | | 10 |
| 3. Операции. Ввод и вывод информации. Операторы. | 11 | 2 | 2 | | 7 | 12 | - | 2 | | 10 |
| 3.1. Пользовательские события. Обработчики событий. Параметры обработчиков. Изменение стандартных способов обработки событий мыши и клавиатуры. Применение событий мыши и клавиатуры для создания пользовательского интерфейса. | 11 | 2 | 2 | | 7 | 12 | - | 2 | | 10 |
| 4. Функции. Прототипы функций. | 11 | 2 | 2 | | 7 | 12 | - | 2 | | 10 |
| 4.1. Проектирование пользовательского интерфейса. Меню и панели инструментов. Разновидности меню и панелей инструментов. Создание панелей инструментов. Использование в приложениях. | 11 | 2 | 2 | | 7 | 12 | - | 2 | | 10 |
| 5. Препроцессор. Глобальные и локальные объекты. Математические функции. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| 5.1. Основные элементы управления. Строки ввода, кнопки, флажки. Специальные элементы (таймер, календарь). Пользовательские диалоговые окна. Приложения с многооконным интерфейсом. Управление поведением окон. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| <i>Итоговое занятие по модулю 1</i> | 4 | - | 2 | 2 | - | - | - | - | | |

| Наименование модулей и разделов дисциплины | Объемы видов учебной работы по формам обучения, час | | | | | | | | | |
|---|---|-----------|-----------------|---------------------|----------------|------------------------|----------|-----------------|---------------------|----------------|
| | Очная форма обучения | | | | | Заочная форма обучения | | | | |
| | Всего | Лекции | Лаб.практ. зан. | Внеаудит. работа | Самост. работа | Всего | Лекции | Лаб.практ. зан. | Внеаудит. работа | Самост. работа |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| Модуль 2. | 65 | 10 | 10 | 10 | 35 | 62 | 4 | 4 | 4 | 50 |
| 1. Указатели. Модели памяти. Динамическое распределение памяти. | 9 | 2 | 2 | Консультации | 5 | 10 | - | - | Консультации | 10 |
| 1.1. Организация ввода-вывода. Работа с текстовыми и бинарными файлами. Классы для работы с файлами FileInfo, Directory, Stream, StreamReader, StreamWriter. Диалоговые компоненты для работы с файлами. | 9 | 2 | 2 | | 5 | 10 | - | - | | 10 |
| 2. Массивы. Передача массива в функцию. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| 2.1. Работа с графикой GDI+. Пространство имен System.Drawing. Формат изображений. Управление текстурой, цветами, шрифтами. Редактирование и вывод изображений. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| 3. Динамическое распределение памяти под массивы | 11 | 2 | 2 | | 7 | 14 | 2 | 2 | | 10 |
| 3.1. Приложения для работы с базой данных. Доступ к данным с помощью технологии ADO.NET. Типы пространства имен System.Data. Элементы управления для создания интерфейса к базе данных. Чтение и запись объектов в различных форматах. Создание команд SQL. | 11 | 2 | 2 | | 7 | 14 | 2 | 2 | | 10 |
| 4. Структуры. Объединения. Битовые поля. | 9 | 2 | - | | 7 | 14 | 2 | 2 | | 10 |
| 4.1. Коммуникационные приложения. Использование протоколов передачи по TCP/IP, соккеты. | 9 | 2 | - | | 7 | 14 | 2 | 2 | | 10 |
| 5. Функции ввода-вывода. Работа с файлами данных. Работа со строками. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| 5.1. Системы контроля версий. GIT, Mercurial и др. Понятие ветви. Операции Commit и Rollback. | 11 | 2 | 2 | | 7 | 10 | - | - | | 10 |
| <i>Итоговое занятие по модулю2</i> | <i>4</i> | - | <i>2</i> | | <i>2</i> | - | - | - | - | |
| <i>Подготовка реферата в форме презентации (контрольной работы)</i> | <i>10</i> | - | - | - | <i>10</i> | <i>20</i> | - | - | - | <i>20</i> |
| Зачёт | 4 | - | - | 4 | - | 4 | - | - | 4 | - |

V. ОЦЕНКА ЗНАНИЙ И ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ ТЕКУЩЕГО КОНТРОЛЯ ЗНАНИЙ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО ДИСЦИПЛИНЕ

5.1. Формы контроля знаний, рейтинговая оценка и формируемые компетенции (дневная форма обучения)

| № п/п | Наименование рейтингов, модулей и блоков | Формируемые компетенции | Объем учебной работы | | | | | Форма контроля знаний | Количество баллов (max) |
|---|---|---------------------------------|----------------------|-----------|-------------------|----------------------------|----------------|-----------------------------------|-------------------------|
| | | | Общая трудоемкость | Лекции | Лабор.-практ.заня | Внеаудиторн. раб. и проме- | Самост. работа | | |
| Всего по дисциплине | | ПК-8 ПК-12 ПК-14 | 144 | 20 | 20 | 24 | 80 | Зачёт | 100 |
| <i>I. Входной рейтинг</i> | | | | | | | | Устный опрос | 5 |
| <i>II. Рубежный рейтинг</i> | | | | | | | | Сумма баллов за модули | 60 |
| Модуль 1. | | ПК-8 ПК-12 ПК-14 | 65 | 10 | 10 | 10 | 35 | | 30 |
| 1. | Введение в язык Си. Структура программы. | | 7 | 2 | - | Консультации | 5 | Устный опрос | |
| 2. | Объекты языка Си и их типы. Простые объекты. | | 11 | 2 | 2 | | 7 | Устный опрос | |
| 3. | Операции. Ввод и вывод информации. Операторы. | | 11 | 2 | 2 | | 7 | Устный опрос | |
| 4. | Функции. Прототипы функций. | | 11 | 2 | 2 | | 7 | Устный опрос | |
| 5. | Препроцессор. Глобальные и локальные объекты. Математические функции. | | 11 | 2 | 2 | | 7 | Устный опрос | |
| Итоговый контроль знаний по темам модуля 1. | | | 4 | - | 2 | | 2 | Тестирование, ситуационные задачи | |
| Модуль 2. | | ПК-8 ПК-12 ПК-14 | 65 | 10 | 10 | 10 | 35 | | 30 |
| 1. | Указатели. Модели памяти. Динамическое распределение памяти. | | 9 | 2 | 2 | Консультации | 5 | Устный опрос | |
| 2. | Массивы. Передача массива в функцию. | | 11 | 2 | 2 | | 7 | Устный опрос | |
| 3. | Динамическое распределение памяти под массивы | | 11 | 2 | 2 | | 7 | Устный опрос | |
| 4. | Структуры. Объединения. Битовые поля. | | 9 | 2 | - | | 7 | Устный опрос | |
| 5. | Функции ввода-вывода. Работа с файлами | | 11 | 2 | 2 | | 7 | Устный опрос | |
| Итоговый контроль знаний по темам модуля 2. | | | 4 | - | 2 | | 2 | Тестирование, ситуационные задачи | |
| <i>III. Творческий рейтинг</i> | | | 10 | - | - | - | 10 | | 5 |
| <i>IV. Выходной рейтинг</i> | | | 4 | - | - | 4 | - | Зачёт | 30 |

5.2. Оценка знаний студента

5.2.1. Основные принципы рейтинговой оценки знаний

Уровень развития компетенций оценивается с помощью рейтинговых баллов.

| Рейтинги | Характеристика рейтингов | Максимум баллов |
|---------------|---|-----------------|
| Входной | Отражает степень подготовленности студента к изучению дисциплины. Определяется по итогам входного контроля знаний на первом практическом занятии. | 5 |
| Рубежный | Отражает работу студента на протяжении всего периода изучения дисциплины. Определяется суммой баллов, которые студент получит по результатам изучения каждого модуля. | 60 |
| Творческий | Результат выполнения студентом индивидуального творческого задания различных уровней сложности, в том числе, участие в различных конференциях и конкурсах на протяжении всего курса изучения дисциплины. | 5 |
| Выходной | Является результатом аттестации на окончательном этапе изучения дисциплины по итогам сдачи экзамена. Отражает уровень освоения информационно-теоретического компонента в целом и основ практической деятельности в частности. | 30 |
| Общий рейтинг | Определяется путём суммирования всех рейтингов | 100 |

Итоговая оценка компетенций студента осуществляется путём автоматического перевода баллов общего рейтинга в стандартные оценки.

Оценка «не зачтено» ставится в том случае, если итоговый рейтинг студента составил менее 60 баллов.

| | |
|------------------------------|--------------------------|
| Не зачтено менее 60 балла | Зачтено 60-100 баллов |
|------------------------------|--------------------------|

5.2.2. Критерии оценки знаний студента на зачете

Оценка на зачете определяется на основании следующих критериев:

- оценка «зачтено» ставится студенту, показавшему систематическое и достаточно глубокое знание учебного материала, умение свободно выполнять ситуационные и тестовые задания, предусмотренные программой, умение логически мыслить и формулировать свою позицию по проблемным вопросам. Зачет может получить студент, который правильно ответил на теоретические вопросы, допустив при этом недочеты не принципиального характера и правильно решившему предложенную на зачете задачу.

- Оценка «не зачтено» ставится студенту, обнаружившему существенные пробелы в знании основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий.

5.3. Фонд оценочных средств. Типовые контрольные задания или иные материалы, необходимые для оценки формируемых компетенций по дисциплине (приложение 2)

VI. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Основная учебная литература

1. Васюткина, И.А. Разработка приложений на С# с использованием СУБД PostgreSQL [Электронный ресурс] / Васюткина И.А., Трошина Г.В., Бычков М.И. - Новосиб.: НГТУ, 2015. - 143 с. Режим доступа: <http://znanium.com/bookread2.php?book=556925>

6.2. Дополнительная литература

1. Технология разработки программного обеспечения: Учеб. пос. [Электронный ресурс] / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с. Режим доступа: <http://znanium.com/bookread2.php?book=389963>

2. Игнатенко, В.А. Методические указания и задания к выполнению лабораторно – практических работ студентов по дисциплине «Разработка программных приложений»/ сост. В.А. Игнатенко - Белгород: Белгородский ГАУ, 2014. – 48с.

Режим доступа: <https://clck.ru/FDrZr>

6.3. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

Самостоятельная работа студентов заключается в инициативном поиске информации о наиболее актуальных проблемах, которые имеют большое практическое значение и являются предметом научных дискуссий в рамках изучаемой дисциплины.

Самостоятельная работа планируется в соответствии с календарными планами рабочей программы по дисциплине и в методическом единстве с тематикой учебных аудиторных занятий.

Самостоятельную работу студента поддерживает электронная информационная среда ВУЗа, доступ к которой <http://do.belgau.edu.ru> (логин, пароль студента)

6.3.1. Методические указания по освоению дисциплины

1. Игнатенко, В. А. Методические указания и задания к выполнению лабораторно-практических и самостоятельных работ студентов по дисциплине "Разработка программных приложений" для студентов экономического факультета направления "Прикладная информатика"

[Электронный ресурс]: методические указания / Белгородский ГАУ; сост. В. А. Игнатенко. - Белгород: Белгородский ГАУ, 2014. - 40 с. Режим доступа: <https://clck.ru/EaRnv>

6.3.2. Видеоматериалы

1. https://www.youtube.com/watch?v=B0Fr8cWWD_w
2. <https://www.youtube.com/watch?v=WfIS-0jqXJ0>

6.3.3 Печатные периодические издания

1. Журнал «Вестник российской сельскохозяйственной науки»
2. Журнал «Достижения науки и техники АПК»
3. Журнал «Экономика, статистика и информатика»

6.4. Ресурсы информационно-телекоммуникационной сети «Интернет», современные профессиональные базы данных, информационные справочные системы.

1. Научно-технический и научно-производственный журнал «Информационные технологии» <http://novtex.ru/IT/arhiv.htm>
2. Профессиональная база данных и информационно справочная система по официальной технической документации для разработчиков под ОС Microsoft Windows <https://msdn.microsoft.com/ru-ru>
3. Профессиональная база данных и информационно справочная система по официальной технической документации для разработчиков под ОС Microsoft Windows <https://technet.microsoft.com/ru-ru>
4. Профессиональная база данных стандартов <http://iso.gost.ru/wps/portal/>
5. Справочно-правовая система Гарант

6.5. Перечень программного обеспечения, информационных технологий.

1. Операционная система Windows;
2. Пакет программ Microsoft Office;
3. SunRay – программа для тестирования;
4. Специализированные инструментальные программные средства и системы: Интегрированная среда разработки CodeBlocks. (бесплатное программное обеспечение <http://www.codeblocks.org/downloads>)

VII. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

Для преподавания дисциплины используются:

1. учебная аудитория лекционного типа, оборудованная мультимедийным оборудованием для демонстрации презентаций;
2. компьютерный класс для проведения лабораторно – практических занятий.
3. помещение для самостоятельной работы обучающихся, оснащенное компьютерной техникой с подключением к сети Интернет и электронной информационно-образовательной среде ВУЗа.
4. Лаборатория прикладной информатики и информационных технологий
5. Лаборатория информатизации АПК

VII. ПРИЛОЖЕНИЯ

Приложение 1

**СВЕДЕНИЯ О ДОПОЛНЕНИИ И ИЗМЕНЕНИИ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ
НА 201 / 201 УЧЕБНЫЙ ГОД**

Разработка программных приложений

дисциплина (модуль)

09.03.03 «Прикладная информатика»

направление подготовки/специальность

| |
|--|
| ДОПОЛНЕНО (с указанием раздела РПД) |
| |
| ИЗМЕНЕНО (с указанием раздела РПД) |
| |
| УДАЛЕНО (с указанием раздела РПД) |
| |

Реквизиты протоколов заседаний кафедр, на которых пересматривалась программа

| | |
|---|---|
| Кафедра информатики и информационных технологий | Кафедра информатики и информационных технологий |
| от _____ № _____ Дата | от _____ № _____ дата |

Методическая комиссия инженерного факультета

« ___ » _____ 201_ года, протокол № _____

Председатель методкомиссии _____ Слободюк А. П.

Декан инженерного факультета

Стребков С.В.

« ___ » _____ 201_ г

Согласовано:

Директор
ООО «Матрица»

«20» июня 2018 г.
Королев И.И.



ФОНД ОЦЕНОЧНЫХ СРЕДСТВ
для проведения промежуточной аттестации обучающихся
по дисциплине Разработка программных приложений
Направление подготовки 09.03.03 Прикладная информатика
Профиль «Прикладная информатика в АПК»

Майский, 2018

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

| Код контролируемой компетенции | Формулировка контролируемой компетенции | Этап (уровень) освоения компетенции | Планируемые результаты обучения | Наименование модулей и (или) разделов дисциплины | Наименование оценочного средства | |
|--------------------------------|---|--|--|--|---|---|
| | | | | | Текущий контроль | Промежуточная аттестация |
| ПК-8 | способность программировать приложения и создавать программные прототипы решения прикладных задач | Первый этап (пороговой уровень) | знать: 1) объектно-ориентированную парадигму языков программирования; 2) свойства и методы элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| | | Модуль 2 | | устный опрос | итоговое тестирование, вопросы к зачёту | |
| | | | | подготовка рефератов | | |
| | | | | тестирование, ситуационные задачи | | |
| | Второй этап (продвинутый уровень) | уметь: 1) проектировать Windows-интерфейс приложения; 2) использовать приемы визуального программирования в сочетании с разработкой программного кода; 3) использовать современные готовые библиотеки классов. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту | |
| | | | | подготовка рефератов | | |
| | | | | тестирование, ситуационные задачи | | |
| Модуль 2 | устный опрос | итоговое тестирование, вопросы к зачёту | | | | |
| | | | подготовка рефератов | | | |
| | | | тестирование, ситуационные задачи | | | |
| Третий этап (высокий уровень) | владеть: 1) практическими приемами алгоритмизации, разработки, отладки и тестирова- | Модуль 1 | устный опрос | итоговое тестирование, вопросы к за- | | |
| | | | подготовка рефератов | | | |

| | | | | | | |
|-----------------------------------|--|--|--|---|---|---|
| | | | ния программ в различных интегрированных средах на объектно-ориентированных языках | | тестирование, ситуационные задачи | чёту |
| | | | | Модуль 2 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| ПК-12 | способностью проводить тестирование компонентов программного обеспечения ИС | Первый этап (пороговой уровень) | знать: 1) основные этапы реализации проектирования программ. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| | | | | Модуль 2 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | подготовка рефератов | | | | |
| | | тестирование, ситуационные задачи | | | | |
| | Второй этап (продвинутый уровень) | уметь: 1) формализовать поставленную задачу; 2) тестировать и отлаживать программы в современных интегрированных средах разработки. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту | |
| | | | | подготовка рефератов | | |
| | | | | тестирование, ситуационные задачи | | |
| | Модуль 2 | | устный опрос | итоговое тестирование, вопросы к зачёту | | |
| подготовка рефератов | | | | | | |
| тестирование, ситуационные задачи | | | | | | |
| Третий этап (высокий уровень) | владеть: 1) навыками алгоритмизации; приёмами разработки, отладки и тестирования приложений. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту | | |
| | | | подготовка рефератов | | | |
| | | | тестирование, ситуационные задачи | | | |
| | | Модуль 2 | устный опрос | итоговое те- | | |

| | | | | | | |
|-------------------------------|---|-----------------------------------|---|---|-----------------------------------|---|
| | | | | | подготовка рефератов | стирование, вопросы к зачёту |
| | | | | | тестирование, ситуационные задачи | |
| ПК-14 | способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач | Первый этап (пороговой уровень) | знать: 1) основные способы и принципы представления абстрактных объектов данных; 2) возможности интегрированных сред разработки; | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| | | | | Модуль 2 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| | | Второй этап (продвинутый уровень) | уметь: 1) применять полученные знания для разработки прикладного программного обеспечения. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| | | | | Модуль 2 | устный опрос | итоговое тестирование, вопросы к зачёту |
| | | | | | подготовка рефератов | |
| | | | | | тестирование, ситуационные задачи | |
| Третий этап (высокий уровень) | владеть: 1) методиками разработки программных приложений для обработки данных. | Модуль 1 | устный опрос | итоговое тестирование, вопросы к зачёту | | |
| | | | подготовка рефератов | | | |
| | | | тестирование, ситуационные задачи | | | |
| | | Модуль 2 | устный опрос | итоговое тестирование, вопросы к зачёту | | |
| | | | подготовка рефератов | | | |
| | | | тестирование, ситуационные задачи | | | |

| | | | | | | |
|--|--|--|--|--|-----------------------------------|------|
| | | | | | тестирование, ситуационные задачи | чёту |
|--|--|--|--|--|-----------------------------------|------|

2. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

| Компетенция | Планируемые результаты обучения (показатели достижения заданного уровня компетенции) | Уровни и критерии оценивания результатов обучения, шкалы оценивания | | | |
|-------------|--|--|--|---|---|
| | | <i>Компетентность не сформирована</i> | <i>Пороговый уровень компетентности</i> | <i>Продвинутый уровень компетентности</i> | <i>Высокий уровень</i> |
| | | <i>не зачтено</i> | <i>зачтено</i> | <i>зачтено</i> | <i>Зачтено</i> |
| ПК-8 | <i>Способность программировать приложения и создавать программные прототипы решения прикладных задач</i> | <i>способность программировать приложения и создавать программные прототипы решения прикладных задач не сформирована</i> | <i>Частично владеет способностью программировать приложения и создавать программные прототипы решения прикладных задач</i> | <i>Владеет способностью программировать приложения и создавать программные прототипы решения прикладных задач</i> | <i>Свободно владеет способностью программировать приложения и создавать программные прототипы решения прикладных задач</i> |
| | Знать: 1) объектно-ориентированную парадигму языков программирования; 2) свойства и методы элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. | Допускает грубые ошибки при воспроизведении объектно-ориентированной парадигмы языков программирования; свойств и методов элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. | Может изложить основы объектно-ориентированной парадигмы языков программирования; свойств и методов элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. | Знает принципы объектно-ориентированной парадигмы языков программирования; свойств и методов элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. | Аргументировано проводит сравнение принципов объектно-ориентированной парадигмы языков программирования; свойств и методов элементов систем разработки приложений, реализующих функциональные возможности объектов WINDOWS. |
| | Уметь: 1) проектировать Windows-интерфейс приложения; 2) использовать приемы визуального программирования в сочетании с разработкой программного кода; 3) использовать современные готовые библиотеки классов. | Не умеет проектировать Windows-интерфейс приложения; использовать приемы визуального программирования в сочетании с разработкой программного кода; использовать современные готовые библиотеки | Частично умеет проектировать Windows-интерфейс приложения; использовать приемы визуального программирования в сочетании с разработкой программного кода; использовать современ- | Способен организовать работу по проектированию Windows-интерфейса приложения; использовать приемы визуального программирования в сочетании с разработкой программного кода; | Способен самостоятельно организовать работу по проектированию Windows-интерфейса приложения; использовать приемы визуального программирования в сочетании с разработ- |

| | | | | | |
|--------------|---|--|--|---|--|
| | | классов | ные готовые библиотеки классов. | использовать современные готовые библиотеки классов | кой программного кода; использовать современные готовые библиотеки классов |
| | Владеть: 1) практическими приемами алгоритмизации, разработки, отладки и тестирования программ в различных интегрированных средах на объектно-ориентированных языках. | Не владеет практическими приемами алгоритмизации, разработки, отладки и тестирования программ в различных интегрированных средах на объектно-ориентированных языках. | Частично владеет практическими приемами алгоритмизации, разработки, отладки и тестирования программ в различных интегрированных средах на объектно-ориентированных языках. | Владеет практическими приемами алгоритмизации, разработки, отладки и тестирования программ в различных интегрированных средах на объектно-ориентированных языках. | Свободно владеет практическими приемами алгоритмизации, разработки, отладки и тестирования программ в различных интегрированных средах на объектно-ориентированных языках. |
| ПК-12 | <i>способность проводить тестирование компонентов программного обеспечения ИС</i> | <i>способность проводить тестирование компонентов программного обеспечения ИС не сформирована</i> | <i>Частично владеет способностью проводить тестирование компонентов программного обеспечения ИС</i> | <i>Владеет способностью проводить тестирование компонентов программного обеспечения ИС</i> | <i>Свободно владеет способностью проводить тестирование компонентов программного обеспечения ИС</i> |
| | Знать: 1) основные этапы реализации проектирования программ. | Допускает грубые ошибки при воспроизведении основных этапов реализации проектирования программ. | Может изложить основные этапы реализации проектирования программ. | Знает основные этапы реализации проектирования программ. | Аргументировано проводит сравнение основных этапов реализации проектирования программ. |
| | Уметь: 1) формализовать поставленную задачу; 2) тестировать и отлаживать программы в современных интегрированных средах разработки. | Не умеет формализовать поставленную задачу; тестировать и отлаживать программы в современных интегрированных средах разработки. | Частично умеет формализовать поставленную задачу; тестировать и отлаживать программы в современных интегрированных средах разработки.. | Способен формализовать поставленную задачу; тестировать и отлаживать программы в современных интегрированных средах разработки.. | Способен формализовать поставленную задачу; тестировать и отлаживать программы в современных интегрированных средах разработки. |
| | Владеть: 1) навыками алгоритмизации; приемами разработки, отлад- | Не владеет навыками алгоритмизации; приемами разработки, отлад- | Частично владеет навыками алгоритмизации; приемами разра- | Владеет навыками алгоритмизации; приемами разработки, от- | Свободно владеет навыками алгоритмизации; приемами раз- |

| | | | | | |
|--------------|---|---|--|---|---|
| | ки и тестирования приложений. | ки и тестирования приложений. | ботки, отладки и тестирования приложений. | ладки и тестирования приложений. | работки, отладки и тестирования приложений. |
| ПК-14 | <i>способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i> | <i>способность осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач не сформирована</i> | <i>Частично владеет способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i> | <i>Владеет способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i> | <i>Свободно владеет способностью осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач</i> |
| | Знать: 1) основные способы и принципы представления абстрактных объектов данных; 2) возможности интегрированных сред разработки. | Допускает грубые ошибки при перечислении основных способов и принципов представления абстрактных объектов данных; возможностей интегрированных сред разработки. | Может изложить основные способы и принципы представления абстрактных объектов данных; возможности интегрированных сред разработки. | Знает основные способы и принципы представления абстрактных объектов данных; возможности интегрированных сред разработки. | Аргументировано проводит выбор и сравнение основных способов и принципов представления абстрактных объектов данных; возможностей интегрированных сред разработки. |
| | Уметь: 1) применять полученные знания для разработки прикладного программного обеспечения. | Не умеет применять полученные знания для разработки прикладного программного обеспечения. | Частично умеет применять полученные знания для разработки прикладного программного обеспечения. | Способен применять полученные знания для разработки прикладного программного обеспечения. | Способен самостоятельно применять полученные знания для разработки прикладного программного обеспечения. |
| | Владеть: 1) методиками разработки программных приложений для обработки данных. | Не владеет методиками разработки программных приложений для обработки данных. | Частично владеет методиками разработки программных приложений для обработки данных. | Владеет методиками разработки программных приложений для обработки данных. | Свободно владеет методиками разработки программных приложений для обработки данных. |

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. Первый этап (пороговой уровень)

ЗНАТЬ (помнить и понимать): студент помнит, понимает и может продемонстрировать широкий спектр фактических, концептуальных, процедурных знаний.

3.1.1. Перечень вопросов для определения входного рейтинга

1. Виды программного обеспечения.
2. Предметная область. Модели предметной области.
3. Преобразование модели предметной области в схему базы данных.
4. Жизненный цикл программного обеспечения. Модели жизненного цикла.
5. Чем регламентируется ЖЦ ПО?
6. Какие группы процессов входят в состав ЖЦ ПО и какие процессы входят в состав каждой группы?
7. Какие процессы, по вашему мнению, наиболее часто используются в реальных процессах и почему?
8. Что понимается под стадией ЖЦ ПО и какие стадии входят в его состав?
9. Каково соотношение между стадиями и процессами ЖЦ ПО?
10. Какие процессы ЖЦ используются для получения достоверных оценок качества ПО?
11. Какие стадии ЖЦ чаще всего присутствуют в модели ЖЦ?
12. Каковы принципиальные особенности каскадной модели?
13. В чем заключаются преимущества и недостатки каскадной модели?
14. Каковы принципиальные особенности спиральной модели?
15. В чем заключаются преимущества и недостатки спиральной модели?

3.1.2. Перечень вопросов к зачету

1. Язык программирования Си. Общая характеристика языка: место в классификации языков, синтаксис и семантика языка, основные объекты языка, операторы, структура программы.
2. Понятие среды программирования. Компоненты среды. Компиляция и компоновка программы.
3. Синтаксические и логические ошибки. Тестирование и отладка программы.
4. Данные в языке Си: константы и переменные. Скалярные типы данных. Модификаторы типов.
5. Данные числовых типов в языке Си: объявление, характеристика, допустимые операции, приведение типов. Пример использования.
6. Операции языка Си. Приоритет операций. Оператор и операция присваивания в языке Си. Множественное присваивание. Выражения.
7. Функции форматного ввода и вывода. Параметры. Управляющая строка. Спецификаторы формата. Управляющие символы.
8. Алгоритмическая конструкция ветвления: полная и неполная форма, блок-схемы. Условная операция. Условный оператор в языке Си: структура оператора, полная и неполная формы, использование сложных условий. Пример на языке Си.
9. Алгоритмическая конструкция выбора: понятие, блок-схема. Оператор выбора в языке Си: структура оператора. Пример программы на Си.
10. Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с предусловием (понятие, использование, блок-схема). Оператор цикла `for`: структура оператора, пример использования.

11. Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с предусловием (понятие, использование, блок-схема). Оператор цикла с предусловием `while` в языке Си: структура оператора, допустимые и недопустимые условия, пример использования.
12. Циклические алгоритмы: понятие, виды (перечислить). Алгоритмическая конструкция цикла с постусловием (понятие, использование, блок-схема). Оператор цикла с постусловием в языке Си: структура оператора, допустимые и недопустимые условия, пример использования.
13. Типовые циклические алгоритмы: максимум/минимум, сумма/произведение, количество.
14. Линейный массив: понятие массива, объявление, инициализация массива, индексация элементов. Формирование и вывод массива.
15. Типовые алгоритмы для работы с линейными массивами.
16. Задача сортировки массива. Алгоритм сортировки линейного массива методом "пузырька".
17. Двумерный массив. Объявление, инициализация двумерного массива, индексация элементов. Формирование и вывод двумерного массива.
18. Типовые алгоритмы для обработки двумерного массива (целиком).
19. Частичная обработка двумерного массива (по строкам или по столбцам). Типовые алгоритмы.
20. Указатели. Понятие указателя, объявление, инициализация. Операции, применимые к указателям. Указатель на указатель.
21. Использование указателей при работе с массивами и матрицами. Получение адресов и значений элементов. Последовательный перебор элементов. Примеры.
22. Функции в языке Си: понятие, объявление, прототипы функций. Вызов функции. Типы возвращаемых значений.
23. Функции в языке Си: параметры формальные и фактические, механизм передачи параметров. Передача параметров «по значению» и «по ссылке». Пример использования.
24. Локальные и глобальные переменные. Области действия и области видимости. Экранирование переменных.
25. Библиотечные функции. Заголовочные файлы. Подключение библиотек. Функции математической библиотеки.
26. Рекурсивное описание функций: база рекурсии, рекурсивный вызов, использование стека. Пример использования.
27. Динамическая память: выделение и освобождение памяти, размещение данных в динамической памяти. Выделение и освобождение памяти при работе с одиночными переменными и с массивами, изменение размера массива.
28. Динамическая память: выделение и освобождение памяти, размещение данных в динамической памяти. Выделение и освобождение памяти при работе с матрицами. Особенности обработки матриц при работе с динамической памятью.
29. Массив символов и строка в языке Си. Ввод и вывод строк. Простейшие алгоритмы сканирования и обработки строки. Пример.
30. Строка в языке Си: библиотечные функции для обработки строк. Примеры использования.

3.2. Второй этап (продвинутый уровень)

УМЕТЬ (применять, анализировать, оценивать, синтезировать): уметь использовать изученный материал в конкретных условиях и в новых ситуациях; осуществлять декомпозицию объекта на отдельные элементы и описывать то, как они соотносятся с целым, выявлять структуру объекта изучения; оценивать значение того или иного материала – научно-технической информации, исследовательских данных и т. д.; комбинировать

элементы так, чтобы получить целое, обладающее новизной

3.2.1. Тестовые задания

1. Что такое функция?
 - a) Некоторая часть программы, содержащая описание переменных и констант основной программы
 - b) Некоторая часть программы, имеющая собственное имя и которая может вызываться из основной программы**
 - c) Некоторая часть программы, содержащая вредоносный код, и блокирует определенные действия системы
 - d) Некоторая часть программы, в которой происходит начальная инициализация всех полей структур, массивов, переменных.

2. Что такое массив?
 - a) Именованный набор переменных имеющих различные типы данных, и располагающихся в одной памяти
 - b) Именованный набор переменных и функций, которые располагаются в одной области памяти
 - c) Именованный набор переменных имеющий один тип данных, и располагающихся в одной области памяти**
 - d) Именованный набор переменных имеющих символьный тип данных, и располагающихся в одной области памяти

3. Как написать следующее выражение на языке C «Переменной a присвоено значение b»?
 - a) `a==b`
 - b) `a=b`**
 - c) `b=a`
 - d) `a:=b`

4. Как написать следующее выражение «Второму элементу массива Myarray присвоено значение пяти»?
 - a) `int [1] Myarray=«пять»`
 - b) `int Myarray [1] = 5`**
 - c) `int Myarray [2] = «пять»`
 - d) `int Myarray [2] = 5`

5. Как написать следующее выражение «Если переменная index больше size то мы инкрементируем переменную count»?
 - a) `if (index>size) { count++; }`**
 - b) `if (index<size) { count--; }`
 - c) `if (index>=size) { ++count; }`
 - d) `if (index<size) { --count; }`

6. Какой диапазон значений имеет тип int для 32-разрядных вычислительных систем:

- a) от 0 до 255
- b) от -32768 до 32767
- c) от 0 до 65535
- d) от 0 до 4 294 967 295**

7. Какой размер в байтах имеет переменная вещественного типа float

- a) 2
- b) 4**
- c) 8
- d) 10

8. Дан массив `int L[3][3] = { { 2, 3, 4 }, { 3, 4, 8 }, { 1, 0, 9 } }`; Чему будет равно значение элемента этого массива `L[1][2]`

- a) 2
- b) 3
- c) 4
- d) 8**

9. Объявление `char *buf`; соответствует

- a) созданию символьной переменной `buf`
- b) созданию строковой переменной `buf`
- c) созданию указателя `buf` на символьное значение**
- d) созданию указателя `buf` на строку

10. Что называется прототипом функции?

- a) описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров
- b) описание функции, включая ее имя, тип возвращаемого значения, типы параметров
- c) имя функции и тип возвращаемого значения**
- d) описание функции, включая ее имя, тип возвращаемого значения, имена и типы параметров, тело функции

11. Как обозначается в языке C (C++) следующий режим работы с потоком - создание нового файла для записи и чтения?

- a) a+
- b) wb
- c) w+**
- d) w+b

12. Какая функция, описанная в заголовочном файле читает строку символов из файла?

- a) `gets()`
- b) `fputs()`
- c) `fgets()`**
- d) `fscanf()`

13. Какой размер массива M будет после выполнения кода:
char M[]="\\nGoodlive" ?

- a) 10
- b) 8
- c) 9
- d) Не определен

14. В каких случаях необходимо использовать оператор return в теле функции?

- a) Всегда
- b) **если необходимо, чтобы функция вернула значение**
- c) если необходимо обеспечить выход из функции в произвольном месте
- d) если указан тип возвращаемого значения, в том числе и void

15. При открытии файла выполняется следующее действие:

- a) физический файл связывается с логическим (файловой переменной)
- b) устанавливается тип файла (текстовый или бинарный)
- c) **устанавливается вид (режим) использования файла**
- d) функцией открытия файла возвращается результат (ошибка)

16. Какое ключевое слово языка C++ используется для описания структурированного типа данных, все элементы которого в памяти начинаются с одного байта?

- a) struct
- b) **union**
- c) enum
- d) template

17. Каким способом можно задать многострочный комментарий в языке C++

- a) **/*комментарии к программе*/**
- b) //комментарии к программе//
- c) //комментарии к программе
- d) {комментарии к программе}

18. Логическое выражение может возвращать результат типа

- a) integer
- b) **boolean**
- c) char
- d) logical

19. Выберите правильный вариант записи на языке C формулы $0 < x < 10$

- a) $x > 0, x \leq 10$
- b) $0 < x \leq 10$
- c) $x > 0 \text{ AND } x \leq 10$
- d) **$(x > 0) \text{ AND } (x < 10)$**

20. Укажите правильный вариант записи условного оператора в языке C

- a) IF x>0 Do y:=sqrt (x)
- b) IF y:=sqrt (x) then x>0
- c) IF x>0 then y:=sqrt (x)
- d) IF (x>0) { y:=sqrt (x)}**

21. Выберите правильный вариант записи на языке C следующего условия: « x принадлежит диапазону [0;10)»

- a) x>=0; x<10
- b) 0<=x<10
- c) (x>0 AND (x<=10))
- d) (x>=0) AND (x<10)**

22. Укажите группу, содержащую последовательность правильно записанных на языке C знаков операций отношений

- a) ~>, <, =, ?
- b) =, <>, ><, >
- c) =, >=, <=, !=**
- d) ~=>, =<, =, <

23. Тело какого цикла всегда будет выполнено хотя бы один раз, независимо от истинности условия:

- a) While
- b) Do While**
- c) For
- d) Нет такого цикла в языке C

24. В результате выполнения кода

```
int i=2;    switch (i)    { case 1: i += 2; case 2: i *= 3; case 6: i /= 2;
default:    ;    }
```

- a) переменная i примет значение 6
- b) переменная i примет значение 3**
- c) переменная i примет значение 2
- d) тело оператора switch не поменяет значение переменной i

25. Укажите директиву препроцессора, которую необходимо подключить для организации форматированного ввода-вывода данных:

- a) #include <iostream.h>
- b) #include <stdio.h>**
- c) #include <math.h>
- d) #include <conio.h>

3.2.2. Темы рефератов

1. История языков программирования.
2. Язык компьютера и человека.

3. Объектно-ориентированное программирование.
4. Непроцедурные системы программирования.
5. Искусственный интеллект и логическое программирование.
6. Языки манипулирования данными в реляционных моделях.
7. Макропрограммирование в среде Microsoft OFFICE.
8. «Визуальное» программирование. VISUAL BASIC, C, PROLOG.
9. Все о DELPHI.
10. Программирование на HTML, JAVA.
11. Издательская система TeX как система программирования.
12. Современные парадигмы программирования. Что дальше?
13. Никлаус Вирт. Структурное программирование. Pascal и Modula.
14. Что мы знаем о Fortran?
15. История языка Бейсик.
16. Язык Ассемблера.
17. Алгоритмический язык Ершова.
18. Все о Logo-мирах.
19. История программирования в лицах.
20. Язык программирования ADA.
21. Язык программирования PL/1.
22. Язык программирования Algol.
23. Язык программирования Си.
24. О фирмах-разработчиках систем программирования.
25. Языки программирования в СУБД.
26. О системах программирования для учебных целей.

3.3. Третий этап (высокий уровень)

ВЛАДЕТЬ навыками по применению теоретических и практических знаний и умений при решении ситуационных задач, практической направленности по дисциплине.

3.3.1. Ситуационные задачи

Задача 1. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Класс> <Фамилия> <Инициалы> <Оценка> <Название предмета>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Для каждого учащегося определить количество оценок по каждому предмету (если по какому-либо предмету учащийся не получил ни одной оценки, то вывести для этого предмета число 0). Сведения о каждом учащемся выводить на отдельной строке, указывая класс, фамилию, инициалы и количество оценок по алгебре, геометрии и информатике. Данные располагать в порядке возрастания номера класса, а для одинаковых классов — в алфавитном порядке фамилий и инициалов.

Задача 2. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Название предмета> <Фамилия> <Инициалы> <Класс> <Оценка>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Вывести сведения об учащихся, имеющих по алгебре среднюю оценку не более 4.0: фамилию, инициалы, номер класса и среднюю оценку по алгебре (выводится с двумя дробными знаками). Учащихся, не имеющих ни одной оценки по алгебре, не учитывать. Сведения о каждом учащемся выводить на отдельной строке и располагать в алфавитном порядке их фамилий и инициалов. Если ни один из учащихся не удовлетворяет указанным условиям, то вывести текст «Требуемые учащиеся не найдены».

Задача 3. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Класс> <Фамилия> <Инициалы> <Название предмета> <Оценка>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Вывести сведения об учащихся, имеющих по информатике среднюю оценку не менее 4.0: номер класса, фамилию, инициалы и среднюю оценку по информатике (выводится с двумя дробными знаками). Сведения о каждом учащемся выводить на отдельной строке и располагать в порядке возрастания классов, а для одинаковых классов — в алфавитном порядке фамилий и инициалов. Если ни один из учащихся не удовлетворяет указанным условиям, то вывести текст «Требуемые учащиеся не найдены».

Задача 4. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается название одного из предметов S и (через пробел) общее число оценок N . Каждая из последующих N строк содержит информацию об одной из оценок в формате

<Фамилия> <Инициалы> <Название предмета> <Оценка> <Класс>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Для каждого класса, присутствующего в наборе исходных данных, определить число учащихся, имеющих по предмету S среднюю оценку не более 3.5. Учащихся, не имеющих ни одной оценки по данному предмету, не учитывать. Сведения о каждом классе выводить на отдельной строке, указывая число найденных учащихся (число может быть равно 0) и номер класса. Данные упорядочивать по возрастанию числа учащихся, а для совпадающих чисел — по убыванию номера класса.

Задача 5. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается название одного из предметов S и (через пробел) общее число оценок N . Каждая из последующих N строк содержит информацию об одной из оценок в формате

<Название предмета> <Фамилия> <Инициалы> <Оценка> <Класс>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается

с заглавной буквы. Для каждого класса, присутствующего в наборе исходных данных, определить число учащихся, имеющих по предмету S среднюю оценку не менее 3.5 и при этом не получивших ни одной двойки по этому предмету. Сведения о каждом классе выводить на отдельной строке, указывая номер класса и число найденных учащихся (число может быть равно 0). Данные упорядочивать по возрастанию номера класса.

Задача 6. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Класс> <Название предмета> <Фамилия> <Инициалы> <Оценка>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Найти всех *двоечников* — учащихся, получивших хотя бы одну двойку по какому-либо предмету. Вывести сведения о каждом из двоечников: номер класса, фамилию, инициалы и полученное число двоек. Сведения о каждом двоечнике выводить на отдельной строке и располагать по убыванию классов, а для одинаковых классов — в алфавитном порядке фамилий и инициалов. Если в наборе исходных данных нет ни одной двойки, то вывести текст «Требуемые учащиеся не найдены».

Задача 7. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Класс> <Оценка> <Фамилия> <Инициалы> <Название предмета>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Найти всех *хорошистов* — учащихся, не получивших ни одной двойки и тройки, но имеющих хотя бы одну четверку по какому-либо предмету. Вывести сведения о каждом хорошисте: полученное число четверок, фамилию, инициалы и номер класса. Сведения о каждом учащемся выводить на отдельной строке и располагать по возрастанию количества четверок, а при их равенстве — в алфавитном порядке фамилий и инициалов. Если в наборе исходных данных нет ни одного учащегося, удовлетворяющего указанным условиям, то вывести текст «Требуемые учащиеся не найдены».

Задача 8. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Класс> <Фамилия> <Инициалы> <Название предмета> <Оценка>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Для каждого класса найти *злостных двоечников* — учащихся, получивших в данном классе максимальное суммарное число двоек по всем предметам (число не должно быть нулевым). Вывести сведения о каждом из злостных двоечников: фамилию, инициалы, номер класса и полученное число двоек. Сведения о каждом двоечнике выводить на отдельной строке и располагать в алфавитном порядке их фамилий и инициа-

лов (сортировку по классам не проводить). Если в наборе исходных данных нет ни одной двойки, то вывести текст «Требуемые учащиеся не найдены».

Задача 9. На вход подаются сведения об оценках учащихся по трем предметам: алгебре, геометрии и информатике. В первой строке указывается общее число оценок N , каждая из последующих N строк содержит информацию об одной из оценок в формате

<Оценка> <Класс> <Фамилия> <Инициалы> <Название предмета>

Количество учащихся не превосходит 100, полных однофамильцев (с совпадающей фамилией и инициалами) среди учащихся нет. Класс описывается целым числом в диапазоне от 7 до 11, оценка — целым числом в диапазоне от 2 до 5. Название предмета указывается с заглавной буквы. Для каждого класса найти *лучших учеников* — учащихся, получивших в данном классе максимальное суммарное число пятерок по всем предметам (число не должно быть нулевым). При поиске лучших учеников (в частности, при определении максимального суммарного числа пятерок) не следует учитывать учащихся, получивших хотя бы одну двойку или тройку. Вывести сведения о каждом из лучших учеников: номер класса, фамилию, инициалы и полученное число пятерок. Сведения о каждом учащемся выводить на отдельной строке и располагать по возрастанию классов, а для одинаковых классов — в алфавитном порядке фамилий и инициалов. Если в наборе исходных данных нет ни одного учащегося, удовлетворяющего указанным условиям, то вывести текст «Требуемые учащиеся не найдены».

Задача 10. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Найти и вывести сумму цифр, входящих в исходный набор. Если набор не содержит цифр, то вывести слово «NO».

Задача 11. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Найти сумму чисел, входящих в исходный набор (число образуют подряд идущие цифры; известно, что сумма чисел не превосходит 20000). Если исходный набор не содержит цифр, то вывести число -1 .

Задача 12. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Попряд идущие в наборе цифры образуют одно число; известно, что все числа имеют не более 4 разрядов (незначащие нули при подсчете количества разрядов не учитываются). Найти общую сумму всех однозначных и двузначных чисел и общее количество всех трехзначных и четырехзначных чисел. Вывести два полученных числа в одной строке в указанном порядке, разделяя их одним пробелом. Если набор не содержит чисел с требуемым количеством разрядов, то в качестве соответствующего числа вывести 0.

Задача 13. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Составить из имеющихся в наборе нечетных цифр минимальное число и вывести строковое представление этого числа. Каждая цифра должна использоваться столько раз, сколько раз она встречается в исходном наборе. Если в наборе нет нечетных цифр, то вывести слово «NO».

Задача 14. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Составить из имеющихся в наборе четных цифр максимальное число и вывести строковое представление этого числа. Каждая цифра должна использоваться столько раз, сколько раз она встречается в исходном наборе (за исключением ситуации, когда отсутствуют все четные цифры, кроме цифры 0; в этом случае надо вывести строку «0»). Если в наборе нет четных цифр, то вывести слово «NO».

Задача 15. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200; в наборе обязательно присутствуют цифры. Составить из имеющихся цифр максимальное число, которое читается одинаково как слева направо, так и справа налево. Все цифры можно не использовать; незначащих нулей в числе не должно быть. Вывести строку, изображающую найденное число.

Задача 16. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Составить из имеющихся в тексте нечетных цифр число максимальной длины, которое читается одинаково как слева направо, так и справа налево. Все цифры можно не использовать. Если требуемое число составить нельзя, то вывести слово «NO»; если можно, то в первой строке вывести слово «YES», а во второй — строку, изображающую найденное число. Если таких чисел несколько, то вывести минимальное из них.

Задача 17. На вход подается набор символов, оканчивающийся точкой. Количество символов не превышает 200. Составить из имеющихся в тексте четных цифр максимальное число, которое читается одинаково как слева направо, так и справа налево. Все цифры можно не использовать; незначащих нулей в числе не должно быть. Если требуемое число составить нельзя, то вывести слово «NO»; если можно, то в первой строке вывести слово «YES», а во второй — строку, изображающую найденное число.

Задача 18. На вход в первой строке подается целое положительное число N , затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Зашифровать исходный текст и вывести его зашифрованный вариант. Шифрование заключается в замене каждой латинской буквы на букву, симметричную шифруемой букве относительно середины английского алфавита (например, «a» должна заменяться на «z», «b» на «у», «x» на «с»). Регистр букв при шифровании остается прежним. Если символ не является латинской буквой, то он не изменяется.

Задача 19. На вход в первой строке подаются два целых положительных числа: K и N , причем K не превосходит 25. Затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Зашифровать исходный текст и вывести его зашифрованный вариант. Шифрование заключается в замене каждой латинской буквы на букву, расположенную в алфавите на K -й позиции после шифруемой буквы (алфавит считается *циклическим*: за буквой «z» опять следует буква «a»). Регистр букв при шифровании остается прежним. Если символ не является латинской буквой, то он не изменяется.

Задача 20. На вход в первой строке подается символ C — латинская буква, во второй строке — целое положительное число N . Затем следует зашифрованный текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Известно, что шифрование выполнялось по тем же правилам, что и в задании ExamTaskC92, причем первый зашифрованный символ текста должен после расшифровки стать символом C . Расшифровать исходный текст и вывести его расшифрованный вариант. Первый зашифрованный символ не обязательно является первым символом текста (поскольку в начале текста может располагаться цифра или знак препинания), однако можно считать, что он всегда присутствует в первой строке текста.

Задача 21. На вход в первой строке подается целое положительное число N , затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Зашифровать исходный текст и вывести его зашифрованный вариант. Шифрование заключается в следующем: каждая латинская буква в слове из L букв заменяется на букву, расположенную в алфавите на L -й позиции после шифруемой буквы (ал-

фавит считается циклическим: за буквой «z» опять следует буква «a»). Например, текст «A gray box.» должен быть заменен на «B kvес ега.». Регистр букв при шифровании остается прежним. Словом считается последовательность латинских букв, не содержащая пробелов и знаков препинания. Если символ не является латинской буквой, то он не изменяется.

Задача 22. На вход в первой строке подаются два целых положительных числа: K и N , причем K не превосходит 10. Затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Зашифровать исходный текст и вывести его зашифрованный вариант. Шифрование заключается в следующем: каждая латинская буква в слове из L букв заменяется на букву, расположенную в алфавите на $(K + L)$ -й позиции после шифруемой буквы (алфавит считается циклическим: за буквой «z» опять следует буква «a»). Например, текст «A gray box.» при $K = 1$ должен быть заменен на «C lwfd fsb.». Регистр букв при шифровании остается прежним. Словом считается последовательность латинских букв, не содержащая пробелов и знаков препинания. Если символ не является латинской буквой, то он не изменяется.

Задача 23. На вход в первой строке подается символ C — латинская буква, во второй строке — целое положительное число N . Затем следует зашифрованный текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Известно, что шифрование выполнялось по тем же правилам, что и в задании ExamTaskC95, причем первый зашифрованный символ текста должен после расшифровки стать символом C . Расшифровать исходный текст и вывести его расшифрованный вариант. Первый зашифрованный символ не обязательно является первым символом текста (поскольку в начале текста может располагаться цифра или знак препинания), однако можно считать, что он всегда присутствует в первой строке текста.

Задача 24. На вход в первой строке подается целое положительное число N , затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Вывести все символы, встретившиеся в исходном тексте, включая пробел и знаки препинания (без повторений). Символы выводить в одной строке (без разделителей) в порядке их первого появления в тексте.

Задача 25. На вход в первой строке подается целое положительное число N , затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Вывести все символы, встретившиеся в исходном тексте, включая пробел и знаки препинания (без повторений). Символы выводить в одной строке (без разделителей) в порядке возрастания их кодов.

Задача 26. На вход в первой строке подается целое положительное число N , затем следует текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Вывести последовательность строчных латинских букв (a, b, c, ..., z) из исходного текста и число их повторений в этом тексте. Вывод производится в алфавитном порядке, каждая буква выводится на новой строке, после буквы указывается пробел и число ее повторений в тексте. Буквы, отсутствующие в тексте, не выводить.

3.4. Представления оценочного средства в фонде

3.4.1. Вопросы для устного опроса (собеседование)

Наименование раздела: Модуль 1

1. Что такое информационная среда программы?
2. Что такое программное средство (ПС)?
3. Что такое ошибка в ПС?
4. Что такое надежность ПС?

5. Что такое технология программирования?
6. Что такое жизненный цикл программного средства (ПС)?
7. Что такое внешнее описание ПС?
8. Что такое сопровождение ПС?
9. Что такое качество ПС?
10. Что такое смежный контроль?

Наименование раздела: Модуль 2

1. Что такое определение требований к программному средству (ПС)?
2. Что такое спецификации качества ПС?
3. Что такое устойчивость (robustness) ПС?
4. Что такое защищенность (defensiveness) ПС?
5. Что такое коммуникабельность (communicativeness) ПС?
6. Что такое функциональная спецификация ПС?
7. Что такое ручная имитация внешнего описания ПС?
8. Что такое программный модуль?
9. Что такое прочность программного модуля?
10. Что такое сцепление программного модуля?

3.4.2. Пример ситуационной задачи (или задачи)

Задание:

На вход в первой строке подается символ C — латинская буква, во второй строке целое положительное число N . Затем следует зашифрованный текст на английском языке, состоящий из N строк. В каждой строке текста содержится не более 80 символов. Известно, что шифрование выполнялось по тем же правилам, что и в задании ExamTaskC92, причем первый зашифрованный символ текста должен после расшифровки стать символом C . Расшифровать исходный текст и вывести его расшифрованный вариант. Первый зашифрованный символ не обязательно является первым символом текста (поскольку в начале текста может располагаться цифра или знак препинания), однако можно считать, что он всегда присутствует в первой строке текста.

3.5. Критериев оценивания контрольных заданий для использования в ФОС дисциплины

3.5.1. Критерии оценивания тестового задания:

Тестовые задания оцениваются по шкале: 1 балл за правильный ответ, 0 баллов за неправильный ответ. Итоговая оценка по тесту формируется путем суммирования набранных баллов и отнесения их к общему количеству вопросов в задании. Помножив полученное значение на 100%, можно привести итоговую оценку к традиционной следующим образом:

| Процент правильных ответов | Оценка |
|----------------------------|---|
| 90 – 100% | От 9 до 10 баллов и/или «отлично» |
| 70 – 89 % | От 6 до 8 баллов и/или «хорошо» |
| 50 – 69 % | От 3 до 5 баллов и/или «удовлетворительно» |
| менее 50 % | От 0 до 2 баллов и/или «неудовлетворительно». |

3.5.2. Критерии оценивания реферата (доклада):

От 4 до 5 баллов и/или «отлично»: глубокое и хорошо аргументированное обоснование темы; четкая формулировка и понимание изучаемой проблемы; широкое и правильное использование относящейся к теме литературы и примененных аналитических методов; содержание исследования и ход защиты указывают на наличие навыков работы студента в данной области; оформление работы хорошее с наличием расширенной библиографии; защита реферата (или выступление с докладом) показала высокий уровень профессиональной подготовленности студента;

От 2 до 3 баллов и/или «хорошо»: аргументированное обоснование темы; четкая формулировка и понимание изучаемой проблемы; использование ограниченного, но достаточного для проведения исследования количества источников; работа основана на среднем по глубине анализе изучаемой проблемы и при этом сделано незначительное число обобщений; содержание исследования и ход защиты (или выступление с докладом) указывают на наличие практических навыков работы студента в данной области; реферат (или доклад) хорошо оформлен с наличием необходимой библиографии; ход защиты реферата (или выступления с докладом) показал достаточную профессиональную подготовку студента;

От 1 до 2 баллов и/или «удовлетворительно»: достаточное обоснование выбранной темы, но отсутствует глубокое понимание рассматриваемой проблемы; в библиографии преобладают ссылки на стандартные литературные источники; труды, необходимые для всестороннего изучения проблемы, использованы в ограниченном объеме; заметна нехватка компетентности студента в данной области знаний; оформление реферата (или доклада) содержит небрежности; защита реферата (или выступление с докладом) показала удовлетворительную профессиональную подготовку студента;

0 баллов и/или «неудовлетворительно»: тема реферата (или доклада) представлена в общем виде; ограниченное число использованных литературных источников; шаблонное изложение материала; суждения по исследуемой проблеме не всегда компетентны; неточности и неверные выводы по рассматриваемой литературе; оформление реферата (или доклада) с элементами заметных отступлений от общих требований; во время защиты (или выступления с докладом) студентом проявлена ограниченная профессиональная эрудиция.

3.5.3. Критерии оценивания на ситуационную задачу:

От 9 до 10 баллов и/или «отлично»: студент глубоко и полно владеет методами решения задачи; решение выполнено оптимальным способом; полученное решение соответствует условиям задачи; решение ситуационной задачи носит самостоятельный характер.

От 6 до 8 баллов и/или «хорошо»: решение студента соответствует указанным выше критериям, но в ход решения имеет отдельные неточности (несущественные ошибки); однако допущенные при решении ошибки исправляются самим студентом после дополнительных вопросов.

От 3 до 5 баллов и/или «удовлетворительно»: студент обнаруживает отсутствие навыков и понимание основных методик решения ситуационной задачи, но решение является неполным, имеет неточности и существенные ошибки; допущенные при решении ошибки не исправляются самим студентом после дополнительных вопросов.

От 0 до 2 баллов и/или «неудовлетворительно»: студент имеет разрозненные, бессистемные знания в области решаемой задачи; не владеет методами и подходами для решения задачи.

3.5.4. Критерии оценивания «Устный опрос»:

Оценка «зачтено» выставляется студенту, если обладает систематизированными знаниями, умениями и навыками по данному разделу дисциплины;

Оценка «не зачтено» выставляется студенту, если он не проявил систематизированных знаний, умений и навыков по данному разделу дисциплины.

3.5.5. Критерий оценивания на зачет

Оценка «зачтено» ставится в том случае, если итоговый рейтинг студента составил 60 и более.

Оценка «не зачтено» ставится в том случае, если итоговый рейтинг студента составил менее 60 баллов.

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

Процедура оценки знаний умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций, производится преподавателем в форме текущего контроля и промежуточной аттестации.

Для повышения эффективности текущего контроля и последующей промежуточной аттестации студентов осуществляется структурирование дисциплины на модули. Каждый модуль учебной дисциплины включает в себя изучение законченного раздела, части дисциплины.

Основными видами текущего контроля знаний, умений и навыков в течение каждого модуля учебной дисциплины являются устный опрос, подготовка рефератов, решение ситуационных задач, тестирование.

Студент должен выполнить все контрольные мероприятия, предусмотренные в модуле учебной дисциплины к указанному сроку, после чего преподаватель проставляет балльные оценки, набранные студентом по результатам текущего контроля модуля учебной дисциплины.

Контрольное мероприятие считается выполненным, если за него студент получил оценку в баллах, не ниже минимальной оценки, установленной программой дисциплины по данному мероприятию.

Промежуточная аттестация обучающихся проводится в форме зачета.

Зачет проводится для оценки уровня усвоения обучающимся учебного материала лекционных курсов и лабораторно-практических занятий, а также самостоятельной работы. Оценка выставляется или по результатам учебной работы студента в течение семестра, или по итогам письменного-устного опроса, или тестирования на последнем занятии. Для дисциплин и видов учебной работы студента, по которым формой итогового отчета является зачет, определена оценка «зачтено», «не зачтено».

Оценка «зачтено» ставится в том случае, если обучающийся:

- владеет знаниями, выделенными в качестве требований к знаниям обучающихся в области изучаемой дисциплины;
- демонстрирует глубину понимания учебного материала с логическим и аргументированным его изложением;
- владеет основным понятийно-категориальным аппаратом по дисциплине;
- демонстрирует практические умения и навыки в области исследовательской деятельности.

Оценка «не зачтено» ставится в том случае, если обучающийся:

- демонстрирует знания по изучаемой дисциплине, но отсутствует глубокое понимание сущности учебного материала;
- допускает ошибки в изложении фактических данных по существу материала, представляется неполный их объем;
- демонстрирует недостаточную системность знаний;
- проявляет слабое знание понятийно-категориального аппарата по дисциплине;
- проявляет непрочность практических умений и навыков в области исследовательской деятельности.

В этом случае студент сдаёт зачёт в форме устных и письменных ответов на любые вопросы в пределах освоенной дисциплины.

Основным методом оценки знаний, умений и навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций является балльно-рейтинговая система, которая регламентируется положением «О балльно-рейтинговой системе оценки качества освоения образовательных программ в ФГБОУ ВО Белгородский ГАУ».

Основными видами поэтапного контроля результатов обучения студентов являются: входной контроль, текущий контроль, рубежный (промежуточный) контроль, творче-

ский контроль, выходной контроль (зачет).

Уровень развития компетенций оценивается с помощью рейтинговых баллов.

| Рейтинги | Характеристика рейтингов | Максимум баллов |
|---------------|---|-----------------|
| Входной | Отражает степень подготовленности студента к изучению дисциплины. Определяется по итогам входного контроля знаний на первом практическом занятии. | 5 |
| Рубежный | Отражает работу студента на протяжении всего периода изучения дисциплины. Определяется суммой баллов, которые студент получит по результатам изучения каждого модуля. | 60 |
| Творческий | Результат выполнения студентом индивидуального творческого задания различных уровней сложности, в том числе, участие в различных конференциях и конкурсах на протяжении всего курса изучения дисциплины. | 5 |
| Выходной | Является результатом аттестации на окончательном этапе изучения дисциплины по итогам сдачи экзамена. Отражает уровень освоения информационно-теоретического компонента в целом и основ практической деятельности в частности. | 30 |
| Общий рейтинг | Определяется путём суммирования всех рейтингов | 100 |

Общий рейтинг по дисциплине складывается из входного, рубежного, выходного (зачета) и творческого рейтинга.

Входной (стартовый) рейтинг – результат входного контроля, проводимого с целью проверки исходного уровня подготовленности студента и оценки его соответствия предъявляемым требованиям для изучения данной дисциплины.

Он проводится на первом занятии при переходе к изучению дисциплины (курса, раздела). Оптимальные формы и методы входного контроля: тестирование, программированный опрос, в т.ч. с применением ПЭВМ и ТСО, решение комплексных и расчетно-графических задач и др.

Рубежный рейтинг – результат рубежного (промежуточного) контроля по каждому модулю дисциплины, проводимого с целью оценки уровня знаний, умений и навыков студента по результатам изучения модуля. Оптимальные формы и методы рубежного контроля: устные собеседования, письменные контрольные опросы, в т.ч. с использованием ПЭВМ и ТСО, результаты выполнения лабораторных и практических заданий. В качестве практических заданий могут выступать крупные части (этапы) курсовой работы или проекта, расчетно-графические задания, микропроекты и т.п.

Выходной рейтинг – результат аттестации на окончательном этапе изучения дисциплины по итогам сдачи зачета, проводимого с целью проверки освоения информационно-теоретического компонента в целом и основ практической деятельности в частности. Оптимальные формы и методы выходного контроля: письменные экзаменационные или контрольные работы, индивидуальные собеседования.

Творческий рейтинг – составная часть общего рейтинга дисциплины, представляет собой результат выполнения студентом индивидуального творческого задания различных уровней сложности.

В рамках рейтинговой системы контроля успеваемости студентов, семестровая составляющая балльной оценки по дисциплине формируется при наборе заданной в программе дисциплины суммы баллов, получаемых студентом при текущем контроле в процессе освоения модулей учебной дисциплины в течение семестра.

Итоговая оценка /зачёта/ компетенций студента осуществляется путём автоматического перевода баллов общего рейтинга в стандартные оценки.

Максимальная сумма рейтинговых баллов по учебной дисциплине составляет 100 баллов.

Оценка «зачтено» ставится в том случае, если итоговый рейтинг студента составил 60 и более.

Оценка «не зачтено» ставится в том случае, если итоговый рейтинг студента составил менее 60 баллов.

| | |
|----------------|---------------|
| Не зачтено | Зачтено |
| менее 60 балла | 60-100 баллов |