

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
Ивангородский гуманитарно-технический институт (филиал)
федерального государственного автономного образовательного учреждения высшего
образования
"Санкт-Петербургский государственный университет аэрокосмического
приборостроения"

Кафедра № 2

УТВЕРЖДАЮ
Ответственный за образовательную
программу

старший преподаватель

(должность, уч. степень, звание)

А.А. Сорокин

(инициалы, фамилия)

(подпись)

«19» июня 2025 г

Лист согласования рабочей программы дисциплины

Программу составил (а)

доц., к.т.н., доц.
(должность, уч. степень, звание)


19.06.2025
(подпись, дата)

Л.Н. Бариков
(инициалы, фамилия)

Программа одобрена на заседании кафедры № 2

«19» июня 2025 г, протокол № 10

И.о. зав. кафедрой № 2

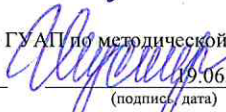
д.ф.-м.н.
(уч. степень, звание)


19.06.2025
(подпись, дата)

Ю.В. Рождественский
(инициалы, фамилия)

Заместитель директора ИФ ГУАП по методической работе

(должность, уч. степень, звание)


19.06.2025
(подпись, дата)

Н.В. Шустер
(инициалы, фамилия)

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

«Алгоритмизация и программирование»
(Наименование дисциплины)

| | |
|---|--|
| Код направления подготовки/ специальности | 15.03.04 |
| Наименование направления подготовки/ специальности | Автоматизация технологических процессов и производств |
| Наименование направленности | Автоматизация технологических процессов и производств |
| Форма обучения | очная |
| Год приема | |

Аннотация

Дисциплина «Алгоритмизация и программирование» входит в образовательную программу высшего образования – программу бакалавриата по направлению подготовки/ специальности 15.03.04 «Автоматизация технологических процессов и производств» направленности «Автоматизация технологических процессов и производств. (ИФ)». Дисциплина реализуется кафедрой «№2».

Дисциплина нацелена на формирование у выпускника следующих компетенций:

ОПК-2 «Применять основные методы, способы и средства получения, хранения, переработки информации»

ОПК-4 «Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности»

ОПК-14 «Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения»

Содержание дисциплины охватывает круг вопросов, связанных с программированием и основами алгоритмизации на языке программирования С и С++.

Преподавание дисциплины предусматривает следующие формы организации учебного процесса: лекции, лабораторные работы, самостоятельная работа студента.

Программой дисциплины предусмотрены следующие виды контроля: текущий контроль успеваемости, промежуточная аттестация в форме экзамена.

Общая трудоемкость освоения дисциплины составляет 4 зачетных единицы, 144 часа.

Язык обучения по дисциплине «русский»

1. Перечень планируемых результатов обучения по дисциплине

1.1. Цели преподавания дисциплины

Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО). Основная цель преподавания дисциплины «Алгоритмизация и программирование» заключается в изучении общих принципов прикладного программирования, изучение применения типовых структур алгоритмов для решения задач, знакомство с объектно-ориентированным и функциональным программированием, а также ознакомление студентов с возможностями применения современных вычислительных средств при практическом решении инженерных задач на основе последних достижений в области программирования и автоматизации инженерных расчетов. Цель дисциплины состоит в получении студентами необходимых теоретических и практических навыков в области программирования на языке С и С++.

1.2. Дисциплина входит в состав обязательной части образовательной программы высшего образования (далее – ОП ВО).

1.3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОП ВО.

В результате изучения дисциплины обучающийся должен обладать следующими компетенциями или их частями. Компетенции и индикаторы их достижения приведены в таблице 1.

Таблица 1 – Перечень компетенций и индикаторов их достижения

| Категория (группа) компетенции | Код и наименование компетенции | Код и наименование индикатора достижения компетенции |
|----------------------------------|---|---|
| Общепрофессиональные компетенции | ОПК-2 Применять основные методы, способы и средства получения, хранения, переработки информации | ОПК-2.3.1 знать основные методы, способы и средства получения, хранения, переработки информации в рамках профессиональной деятельности ОПК-2.У.1 уметь применять методы, способы и средства получения, хранения, переработки информации в рамках профессиональной деятельности ОПК-2.В.1 владеть навыками работы с информацией в рамках профессиональной деятельности |
| Общепрофессиональные компетенции | ОПК-4 Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности | ОПК-4.3.1 знать принципы работы в программных продуктах, обеспечивающих сферу профессиональной деятельности ОПК-4.У.1 уметь использовать программные средства для решения задач профессиональной деятельности |
| Общепрофессиональные компетенции | ОПК-14 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для | ОПК-14.3.1 знать принципы разработки компьютерных программ и алгоритмов, применяемых в рамках профессиональной деятельности, в том числе с применением искусственного интеллекта |

| | | |
|--|--------------------------|---|
| | практического применения | ОПК-14.У.1 уметь применять разработанные алгоритмы решения, в том числе интеллектуальные, практических задач в профессиональной деятельности ОПК-14.В.1 владеть навыками решения профессиональных задач с применением разработанных компьютерных программ и алгоритмов |
|--|--------------------------|---|

2. Место дисциплины в структуре ОП

Дисциплина может базироваться на знаниях, ранее приобретенных обучающимися при изучении следующих дисциплин:

- Информатика

Знания, полученные при изучении материала данной дисциплины, имеют как самостоятельное значение, так и используются при изучении других дисциплин:

- Программирование микроконтроллеров

3. Объем и трудоемкость дисциплины

Данные об общем объеме дисциплины, трудоемкости отдельных видов учебной работы по дисциплине (и распределение этой трудоемкости по семестрам) представлены в таблице 2.

Таблица 2 – Объем и трудоемкость дисциплины

| Вид учебной работы | Всего | Трудоемкость по семестрам |
|---|--------|---------------------------|
| | | №2 |
| 1 | 2 | 3 |
| Общая трудоемкость дисциплины, ЗЕ/ (час) | 4/ 144 | 4/ 144 |
| Из них часов практической подготовки | | |
| Аудиторные занятия, всего час. | 51 | 51 |
| в том числе: | | |
| лекции (Л), (час) | 17 | 17 |
| практические/семинарские занятия (ПЗ), (час) | | |
| лабораторные работы (ЛР), (час) | 34 | 34 |
| курсовой проект (работа) (КП, КР), (час) | | |
| экзамен, (час) | 27 | 27 |
| Самостоятельная работа, всего (час) | 66 | 66 |
| Вид промежуточной аттестации: зачет, дифф. зачет, экзамен (Зачет, Дифф. зач, Экз.**) | Экз. | Экз. |

Примечание: ** кандидатский экзамен

4. Содержание дисциплины

4.1. Распределение трудоемкости дисциплины по разделам и видам занятий.

Разделы, темы дисциплины и их трудоемкость приведены в таблице 3.

Таблица 3 – Разделы, темы дисциплины, их трудоемкость

| Разделы, темы дисциплины | Лекции (час) | ПЗ (СЗ) | ЛР (час) | КП (час) | СРС (час) |
|--------------------------|--------------|---------|----------|----------|-----------|
|--------------------------|--------------|---------|----------|----------|-----------|

| Семестр 2 | | | | | |
|---|---|--|---|--|---|
| Раздел 1. Структура персональной ЭВМ Тема 1.1. Размещение данных и программ в памяти ЭВМ Тема 1.2. Программные модули. Ошибки Тема 1.3. Функциональная и модульная декомпозиция. Тема 1.4. Файловая система хранения информации Тема 1.5. Операционная система. | 1 | | 3 | | 5 |
| Раздел 2. Основы алгоритмизации Тема 2.1. Основные свойства алгоритма Тема 2.2. Общие принципы разработки алгоритмов. Примеры алгоритмизации задач | 1 | | 2 | | 8 |
| Раздел 3. Языки программирования и основные понятия алгоритмического языка Тема 3.1. Алгоритм, язык программирования, программа Тема 3.2 Компиляторы и интерпретаторы Тема 3.3 Уровни языков программирования Тема 3.4 Состав и описание алгоритмического языка | 2 | | 2 | | 8 |
| Раздел 4. Введение в язык программирования Си Тема 4.1. Алфавит языка Си. Элементарные конструкции (лексемы) языка Си Тема 4.2. Концепция типа данных. Типы данных. Структура программы Тема 4.3. Операции и выражения. Алгоритм и операторы | 2 | | 5 | | 8 |
| Раздел 5. Операторы простой последовательности действий Тема 5.1. Функции форматного ввода-вывода данных. Функции ввода-вывода символов. Ввод-вывод данных в языке C++. Основные библиотечные функции Тема 5.2. Примеры задач на использование операторов простой последовательности и библиотечных функций | 3 | | 5 | | 8 |
| Раздел 6. Условные конструкции: операторы ветвления Тема 6.1. Условный оператор. Примеры задач на использование условного оператора Тема 6.2. Оператор множественного выбора (переключатель). Пример задачи на использование оператора множественного выбора | 2 | | 4 | | 8 |
| Раздел 7. Условные конструкции: операторы циклов Тема 7.1. Оператор цикла с параметром (счетчиком). Пример задачи на использование оператора цикла с параметром Тема 7.2. Итерационные циклы. Примеры задач на использование итерационных циклов (с предусловием и с постусловием) | 2 | | 3 | | 8 |

| | | | | | |
|---|----|---|----|---|----|
| Раздел 8. Указатели и массивы данных Тема 8.1. Указатели Тема 8.2. Массив как статическая структура данных и адресная арифметика Тема 8.3. Линейный поиск и сортировка в массивах данных Тема 8.4. Рекомендации при работе со статическим массивом данных | 3 | | 8 | | 8 |
| Раздел 9. Строки Тема 9.1. Символьные и строковые литералы и переменные Тема 9.2. Операции со строками | 1 | | 2 | | 5 |
| Итого в семестре: | 17 | | 34 | | 66 |
| Итого | 17 | 0 | 34 | 0 | 66 |

Практическая подготовка заключается в непосредственном выполнении обучающимися определенных трудовых функций, связанных с будущей профессиональной деятельностью.

4.2. Содержание разделов и тем лекционных занятий.

Содержание разделов и тем лекционных занятий приведено в таблице 4.

Таблица 4 – Содержание разделов и тем лекционного цикла

| Номер раздела | Название и содержание разделов и тем лекционных занятий |
|---------------|---|
| 1 | <p style="text-align: center;">Структура персональной ЭВМ</p> <p>1.1. Размещение данных и программ в памяти ЭВМ: центральный процессор, оперативное запоминающее устройство, постоянное запоминающее устройство, дисковод.</p> <p>1.2. Размещение данных и программ в памяти ЭВМ: программа, команда, ячейка, программа в машинных кодах</p> <p>1.3. Программные модули. Ошибки.</p> <p>Транслятор, компилятор, интерпретатор, объектный модуль.</p> <p>Синтаксические ошибки, логические ошибки.</p> <p>1.4. Файловая система хранения информации: функциональная декомпозиция, модельная декомпозиция, файловая система, каталог, подкаталог, маршрут файла.</p> <p>1.5. Операционная система: примеры операционных систем.</p> |
| 2 | <p style="text-align: center;">Основы алгоритмизации</p> <p>2.1. Основные свойства алгоритма: свойства алгоритмов, графическое описание алгоритмов, словесное описание алгоритмов, основные символы схемы алгоритмов.</p> <p>2.2. Общие принципы разработки алгоритмов. Примеры алгоритмизации задач. Алгоритмические структуры: альтернативный, циклический, итерационный цикл, цикл с параметром. Словесный способ, графический способ, псевдокод. Структурная схема, блок-схема. Виды графических символов для построения блок-схемы алгоритма.</p> |
| 3 | Языки программирования и основные понятия алгоритмического языка |

| | |
|---|--|
| | <p>3.1. Алгоритм, язык программирования, программа: транслятор, тестирование, отладка, синтаксис языка.</p> <p>3.2. Компиляторы и интерпретаторы: основы.</p> <p>3.3. Уровни языков программирования: языки программирование низкого уровня, языки программирования высокого уровня</p> <p>3.4. Состав и описание алгоритмического языка: алгоритмический язык, описание языка (выражения, операторы, символы), синтаксические определения.</p> |
| 4 | <p>Введение в язык программирования Си</p> <p>4.1.Алфавит языка Си. Элементарные конструкции (лексемы) языка Си: идентификатор, служебные слова, константы, знаки операций, комментарии, разделители.</p> <p>4.2. Концепция типа данных. Типы данных. Структура программы. Базовые, целочисленные, вещественные типы данных, представление символьных данных, тип void.</p> <p>4.3. Операции и выражения. Алгоритм и операторы Знаки операций, операции присваивания, арифметические операции, операции отношения и логические операции, условная операция, операция явного преобразования типа.</p> |
| 5 | <p>Операторы простой последовательности действий</p> <p>5.1. Функции форматного ввода-вывода данных. Функции ввода-вывода символов. Ввод-вывод данных в языке C++. Основные библиотечные функции Ввод-вывод данных, функция форматного ввода данных, спецификаторы преобразования, символы преобразования, символы, не являющиеся разделителями, функция форматного ввода данных. Пример организации форматного ввода-вывода данных.</p> <p>5.2. Примеры задач на использование операторов простой последовательности и библиотечных функций: реализация программы.</p> |
| 6 | <p>Условные конструкции: операторы ветвления</p> <p>6.1. Условный оператор. Примеры задач на использование условного оператора Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си.</p> <p>6.2. Оператор множественного выбора (переключатель). Пример задачи на использование оператора множественного выбора Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си.</p> |
| 7 | <p>Условные конструкции: операторы циклов</p> <p>7.1. Оператор цикла с параметром (счетчиком). Пример задачи на использование оператора цикла с параметром Постановка задачи. Математическая модель и описательный алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си.</p> <p>7.2. Итерационные циклы. Примеры задач на использование итерационных циклов (с предусловием и с постусловием) Постановка задачи. Математическая модель и описательный</p> |

| | |
|----------|--|
| | алгоритм решения. Блок-схема алгоритма Текст программы – реализация алгоритма на языке Си. |
| 8 | Указатели и массивы данных 8.1. Указатели 8.2. Массив как статическая структура данных и адресная арифметика 8.3. Линейный поиск и сортировка в массивах данных 8.4. Рекомендации при работе со статическим массивом данных. Примеры программ |
| 9 | Строки 9.1. Символьные и строковые литералы и переменные 9.2. Операции со строками |

4.3. Практические (семинарские) занятия

Темы практических занятий и их трудоемкость приведены в таблице 5.

Таблица 5 – Практические занятия и их трудоемкость

| № п/п | Темы практических занятий | Формы практических занятий | Трудоемкость, (час) | Из них практической подготовки, (час) | № раздела дисциплины |
|---------------------------------|---------------------------|----------------------------|---------------------|---------------------------------------|----------------------|
| Учебным планом не предусмотрено | | | | | |

4.4. Лабораторные занятия

Темы лабораторных занятий и их трудоемкость приведены в таблице 6.

Таблица 6 – Лабораторные занятия и их трудоемкость

| № п/п | Наименование лабораторных работ | Трудоемкость, (час) | Из них практической подготовки, (час) | № раздела дисциплины |
|-----------|---|---------------------|---------------------------------------|----------------------|
| Семестр 2 | | | | |
| 1 | Алгоритмы линейной структуры и их программирование на Си и Си++ | 4 | | 1-3 |
| 2 | Операторы повторения на языке Си | 4 | | 1-3 |
| 3 | Указатели на языке Си | 4 | | 1-4 |
| 4 | Условный оператор на Си++ | 4 | | 4-5 |
| 5 | Оператор выбора на Си++ | 4 | | 4-5 |
| 6 | Стандартные алгоритмы работы с одномерными массивами на Си++ | 4 | | 6-8 |
| 7 | Двумерный массив на Си++ | 4 | | 6-8 |
| 8 | Операции с матрицами на Си++ | 2 | | 6-8 |
| 9 | Строки в С++ | 4 | | 9 |
| Всего | | 34 | | |

4.5. Курсовое проектирование/ выполнение курсовой работы

Учебным планом не предусмотрено

4.6. Самостоятельная работа обучающихся

Виды самостоятельной работы и ее трудоемкость приведены в таблице 7.

Таблица 7 – Виды самостоятельной работы и ее трудоемкость

| Вид самостоятельной работы | Всего, час | Семестр 2, час |
|---|------------|----------------|
| 1 | 2 | 3 |
| Изучение теоретического материала дисциплины (ТО) | 20 | 20 |
| Подготовка к текущему контролю успеваемости (ТКУ) | 16 | 16 |
| Домашнее задание (ДЗ) | 10 | 10 |
| Подготовка к промежуточной аттестации (ПА) | 20 | 20 |
| Всего: | 66 | 66 |

5. Перечень учебно-методического обеспечения

для самостоятельной работы обучающихся по дисциплине (модулю)

Учебно-методические материалы для самостоятельной работы обучающихся указаны в п.п. 7-11.

6. Перечень печатных и электронных учебных изданий

Перечень печатных и электронных учебных изданий приведен в таблице 8.

Таблица 8– Перечень печатных и электронных учебных изданий

| Шифр/ URL адрес | Библиографическая ссылка | Количество экземпляров в библиотеке (кроме электронных экземпляров) |
|---|---|---|
| URL: https://znanium.ru/catalog/document?id=361010 | Голицына, О. Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-570-7. | - |
| URL: https://znanium.ru/catalog/document?id=398591&pid=1054007 | Колдаев, В. Д. Структуры и алгоритмы обработки данных : учебное пособие / В.Д. Колдаев. — Москва : РИОР : ИНФРА-М, 2021. — 296 с. — (Высшее образование: Бакалавриат). — www.dx.doi.org/10.12737/2833 . - ISBN 978-5-369-01264-2. | - |

7. Перечень электронных образовательных ресурсов

информационно-телекоммуникационной сети «Интернет»

Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины приведен в таблице 9.

Таблица 9 – Перечень электронных образовательных ресурсов информационно-телекоммуникационной сети «Интернет»

| URL адрес | Наименование |
|------------------------|--|
| https://www.intuit.ru/ | Национальный Открытый Университет "ИНТУИТ" |
| https://elibrary.ru/ | eLIBRARY.RU - Научная электронная библиотека |
| http://lib.guap.ru/ | Библиотека ГУАП |
| https://znanium.com/ | Электронно-библиотечная система Znanium |
| https://urait.ru/ | Образовательная платформа Юрайт |

8. Перечень информационных технологий

8.1. Перечень программного обеспечения, используемого при осуществлении образовательного процесса по дисциплине.

Перечень используемого программного обеспечения представлен в таблице 10.

Таблица 10– Перечень программного обеспечения

| № п/п | Наименование |
|-------|--|
| 1. | Microsoft Office Professional Plus |
| 2. | Microsoft Windows 10 Professional |
| 3. | Microsoft Visio |
| 4. | Firefox |
| 5. | Acrobat Reader DC |
| 6. | Консультант Плюс |
| 7. | 7-Zip |
| 8. | Visual Studio Community 2022 |
| 9. | Oracle VirtualBox GNU General Public License |
| 10. | Arduino IDE |
| 11. | Gnu/Linux (Ubuntu) |
| 12. | OpenOffice |
| 13. | LibreOffice |

8.2. Перечень информационно-справочных систем, используемых при осуществлении образовательного процесса по дисциплине

Перечень используемых информационно-справочных систем представлен в таблице 11.

Таблица 11– Перечень информационно-справочных систем

| № п/п | Наименование |
|-------|------------------|
| | Не предусмотрено |

9. Материально-техническая база

Состав материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине, представлен в таблице 12.

Таблица 12 – Состав материально-технической базы

| № п/п | Наименование составной части материально-технической базы | Номер аудитории (при необходимости) |
|-------|---|-------------------------------------|
|-------|---|-------------------------------------|

| | | |
|---|--|-----|
| 1 | <p>Лаборатория прикладной математики и информационных технологий для занятий лекционного типа, занятий практического типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы № 206</p> <p>Основное оборудование: стол преподавателя – 1 шт. столы ученические – 17 шт. стулья – 35 шт. доска маркерная – 1 шт. тематические стенды – 11 шт. Системный блок UNIVERSAL i3 D2 – 8 шт Монитор ACER V173Dob – 8 шт Проектор BENQ MW550 – 1 шт. Экран для проектора Cactus Wallscreen CS-PSW-183x244 4:3 Экран SCREEN MEDIA APOLLO T200x200 – 1 шт. Мышь Genius PS/2 – 8шт Сетевой фильтр Defender ES – 1 шт. Коммутатор 16 port – 1 шт. Клавиатура Logitech USB – 8 шт Ноутбук Acer Aspire E1-570G-53334G50Mnii.NX.MJ4ER.001 – 1 шт.</p> | 206 |
| 2 | <p>Помещения для организации самостоятельной работы № 111</p> <p>Библиотека, читальный зал: Мебель; WiFi с выходом в вычислительную сеть ИФ ГУАП и Интернет, обеспечивающий доступ в электронную информационно-образовательную среду организации и к подписным ресурсам: Электронно-библиотечные системы «ZNANIUM», «Юрайт», «Лань»; Оборудованные места для самостоятельной работы, зонированные офисными перегородками – бшт. Системный блок UNIVERSAL i3 D2 -8 шт Монитор ACER V173Dob - 8 шт Клавиатура 8 - шт Мышь Genius PS/2 - 8 шт МФУ Kyocera m2035dn - 2 шт Коммутатор 8 port -2 шт</p> | 111 |

10. Оценочные средства для проведения промежуточной аттестации

10.1. Состав оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведен в таблице 13.

Таблица 13 – Состав оценочных средств для проведения промежуточной аттестации

| Вид промежуточной аттестации | Перечень оценочных средств |
|------------------------------|---------------------------------------|
| Экзамен | Список вопросов к экзамену; Тесты. |

10.2. В качестве критериев оценки уровня сформированности (освоения) компетенций обучающимися применяется 5-балльная шкала оценки сформированности компетенций, которая приведена в таблице 14. В течение семестра может использоваться 100-балльная шкала модульно-рейтинговой системы Университета, правила использования которой, установлены соответствующим локальным нормативным актом ГУАП.

Таблица 14 –Критерии оценки уровня сформированности компетенций

| Оценка компетенции | Характеристика сформированных компетенций |
|---------------------------------------|---|
| 5-балльная шкала | |
| «отлично» «зачтено» | <ul style="list-style-type: none"> – обучающийся глубоко и всесторонне усвоил программный материал; – уверенно, логично, последовательно и грамотно его излагает; – опираясь на знания основной и дополнительной литературы, тесно привязывает усвоенные научные положения с практической деятельностью направления; – умело обосновывает и аргументирует выдвигаемые им идеи; – делает выводы и обобщения; – свободно владеет системой специализированных понятий. |
| «хорошо» «зачтено» | <ul style="list-style-type: none"> – обучающийся твердо усвоил программный материал, грамотно и по существу излагает его, опираясь на знания основной литературы; – не допускает существенных неточностей; – увязывает усвоенные знания с практической деятельностью направления; – аргументирует научные положения; – делает выводы и обобщения; – владеет системой специализированных понятий. |
| «удовлетворительно» «зачтено» | <ul style="list-style-type: none"> – обучающийся усвоил только основной программный материал, по существу излагает его, опираясь на знания только основной литературы; – допускает несущественные ошибки и неточности; – испытывает затруднения в практическом применении знаний направления; – слабо аргументирует научные положения; – затрудняется в формулировании выводов и обобщений; – частично владеет системой специализированных понятий. |
| «неудовлетворительно» «не зачтено» | <ul style="list-style-type: none"> – обучающийся не усвоил значительной части программного материала; – допускает существенные ошибки и неточности при рассмотрении проблем в конкретном направлении; – испытывает трудности в практическом применении знаний; – не может аргументировать научные положения; – не формулирует выводов и обобщений. |

10.3. Типовые контрольные задания или иные материалы.

Вопросы (задачи) для экзамена представлены в таблице 15.

Таблица 15 – Вопросы (задачи) для экзамена

| № п/п | Перечень вопросов (задач) для экзамена | Код индикатора |
|-------|---|----------------|
| 1 | Структура персональной ЭВМ | ОПК-2.3.1 |
| 2 | Размещение данных и программ в памяти ЭВМ | ОПК-2.У.1 |
| 3 | Программные модули | ОПК-2.В.1 |
| 4 | Ошибки | ОПК-4.3.1 |
| 5 | Функциональная и модульная декомпозиции | ОПК-4.У.1 |
| 6 | Файловая система хранения информации | ОПК-14.3.1 |
| 7 | Операционная система | ОПК-14.У.1 |

| | | |
|----|---|------------|
| 8 | Основные свойства алгоритма | ОПК-14.В.1 |
| 9 | Общие принципы разработки алгоритмов | ОПК-2.3.1 |
| 10 | Примеры алгоритмизации задач | ОПК-2.У.1 |
| 11 | Алгоритм, язык программирования, программа | ОПК-2.В.1 |
| 12 | Компиляторы и интерпретаторы | ОПК-4.3.1 |
| 13 | Уровни языков программирования | ОПК-4.У.1 |
| 14 | Состав и описание алгоритмического языка | ОПК-14.3.1 |
| 15 | Алфавит языка Си. Элементарные конструкции (лексемы) языка Си | ОПК-14.У.1 |
| 16 | Концепция типа данных | ОПК-14.В.1 |
| 17 | Типы данных | ОПК-2.3.1 |
| 18 | Структура программы | ОПК-2.У.1 |
| 19 | Операции и выражения | ОПК-2.В.1 |
| 20 | Алгоритм и операторы | ОПК-4.3.1 |
| 21 | Функции форматного ввода-вывода данных | ОПК-4.У.1 |
| 22 | Функции ввода-вывода символов | ОПК-14.3.1 |
| 23 | Ввод-вывод данных в языке C++ | ОПК-14.У.1 |
| 24 | Основные библиотечные функции | ОПК-14.В.1 |
| 25 | Пример задачи на использование операторов простой последовательности | ОПК-2.3.1 |
| 26 | Пример задачи на использование операторов библиотечных функций | ОПК-2.3.1 |
| 27 | Условный оператор | ОПК-2.3.1 |
| 28 | Примеры задач на использование условного оператора | ОПК-2.У.1 |
| 29 | Оператор множественного выбора (переключатель) | ОПК-2.3.1 |
| 30 | Пример задачи на использование оператора множественного выбора | ОПК-2.У.1 |
| 31 | Оператор цикла с параметром (счетчиком) | ОПК-2.В.1 |
| 32 | Пример задачи на использование оператора цикла с параметром | ОПК-4.3.1 |
| 33 | Итерационные циклы | ОПК-4.У.1 |
| 34 | Примеры задач на использование итерационных циклов (с предусловием) | ОПК-14.3.1 |
| 35 | Примеры задач на использование итерационных циклов (с постусловием) | ОПК-14.У.1 |
| 36 | Разработать алгоритм суммирования n введенных чисел, вычисления их среднего арифметического значения и вывода полученных значений. | ОПК-14.В.1 |
| 37 | Разработать алгоритм суммирования положительных из n введенных чисел и вывода полученного значения суммы | ОПК-2.3.1 |
| 38 | Разработать алгоритм вычисления произведения только отрицательных из n введенных чисел с вычетом из него значения первого введенного числа и вывода полученного значения. | ОПК-2.У.1 |
| 39 | Вычислить и вывести на экран значение функции $2 \cdot x^3 + x + 1$, если $x \leq 0$; $y = -2 \cdot x^2 + 3$, если $0 < x < 10$; 0 , если $x \geq 10$. | ОПК-2.В.1 |
| 40 | Ввести координаты точки x, y , присвоить $z=1$, если точка принадлежит окружности с введенным радиусом R и центром в точке с введенными координатами a, b , и присвоить $z=0$ – в противном случае; вывести значение z . | ОПК-4.3.1 |
| 41 | Вывести на экран сообщение, в какой четверти координатной плоскости находится точка с координатами x, y , если $x \cdot y \neq 0$. | ОПК-4.У.1 |
| 42 | Вывести минимальное и максимальное значение из трех введенных чисел x_1, x_2, x_3 . | ОПК-2.3.1 |

| | | |
|----|--|------------|
| 43 | Определить, является ли введенное число n совершенным, т.е. равным сумме всех своих делителей, не превосходящих само число; вывести соответствующее сообщение. | ОПК-2.У.1 |
| 44 | Вычислить и вывести для введенного натурального числа n сумму $S=1+22+33+\dots+nn$, не используя стандартную функцию возведения в степень. | ОПК-2.В.1 |
| 45 | Сумма цифр трехзначного числа кратна 7, само число также делится на 7. Найти и вывести все такие числа. | ОПК-4.3.1 |
| 46 | Определить и вывести все числа, кратные введенным числам a и b , меньшие $a \cdot b$. | ОПК-4.У.1 |
| 47 | В массиве $A=(a_1, a_2, \dots, a_n)$ все положительные элементы, стоящие перед минимальным положительным элементом, переслать в «хвост» массива. | ОПК-14.3.1 |
| 48 | Дан двумерный массив целых чисел из 4 столбцов и 3 строк. Найти максимальный элемент в массиве и заменить его нулем. | ОПК-14.У.1 |
| 49 | Дана строка. Удалить слова со второго по четвертое. | ОПК-14.В.1 |
| 50 | Написать программу, которая бы по введенному месяцу выдает все приходящиеся на этот месяц праздничные дни. | ОПК-2.3.1 |

Вопросы (задачи) для зачета / дифф. зачета представлены в таблице 16.

Таблица 16 – Вопросы (задачи) для зачета / дифф. зачета

| № п/п | Перечень вопросов (задач) для зачета / дифф. зачета | Код индикатора |
|-------|---|----------------|
| | Учебным планом не предусмотрено | |

Перечень тем для курсового проектирования/выполнения курсовой работы представлены в таблице 17.

Таблица 17 – Перечень тем для курсового проектирования/выполнения курсовой работы

| № п/п | Примерный перечень тем для курсового проектирования/выполнения курсовой работы |
|-------|--|
| | Учебным планом не предусмотрено |

Вопросы для проведения промежуточной аттестации в виде тестирования представлены в таблице 18.

Таблица 18 – Примерный перечень вопросов для тестов

| № п/п | Примерный перечень вопросов для тестов | Код индикатора |
|-------|---|----------------|
| 1 | Укажите, в каком году впервые был стандартизован язык C++. Необходимо выбрать один правильный ответ. 1) 1978 2) 1983 3) 1989 4) 1999 5) 2003 6) 2013 7) 2011 8) 1998 9) 2002 10) 2015 11) 2008 12) 2001 13) 1995 | ОПК-2 |

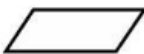

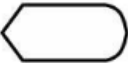





| | | |
|---|---|-------|
| | 14) 1980 | |
| 2 | <p>В языке программирования C++ дан массив <code>int array[5] = { 3, 10, 7, 9, 2}</code>. Выберите ответ, где можно обратиться к числу 7.</p> <p>1) <code>array[7]</code>; 2) <code>array[2]</code>; 3) <code>array[3]</code>;</p> | ОПК-2 |
| 3 | <p>Дан следующий код на языке программирования C++:</p> <pre>#include <iostream> int main() { for (int i=3; i<8; i++) { std::cout return 0; } }</pre> <p>Выберете правильный результат кода.</p> <p>1) 3,4,5,6,7, 2) 3,4,5,6,7 3) 3 4 5 6 7 4) 01234567 5) 34567</p> | ОПК-2 |
| 4 | <p>Дан следующий код на языке программирования C++:</p> <pre>#include <iostream> int main() { int x=1; switch(x) { case 1:std::cout << "Один"; case 0:std::cout << "Нуль"; case 3:std::cout << "Всем привет!"; } return 0; }</pre> <p>Напишите результат программы</p> | ОПК-2 |
| 5 | <p>Выберите из списка объявления, которые стоит помещать в заголовочные файлы в языке программирования C++</p> <p>1) <code>void foo()</code> 2) <code>void bar() { foo(); }</code> 3) <code>int a;</code> 4) <code>void foo();</code> 5) <code>extern int a;</code></p> | ОПК-2 |
| 6 | <p>Дан следующий код на языке программирования C++.</p> <pre>#include <iostream> int main() { std::cout << "Доброе утро" << "\n"; std::cout << "Сегодня есть пара?"; return 0; }</pre> <p>Укажите правильный результат данного кода.</p> <p>1) Доброе утро Сегодня есть пара? 2) Доброе утро/nСегодня есть пара? 3) Доброе утро 4) Сегодня есть пара? 5) Доброе утроСегодня есть пара?</p> | ОПК-2 |
| 7 | <p>Дан код программы на языке программирования C++:</p> <pre>#include <iostream> int main() {</pre> | ОПК-2 |

| | | |
|----|--|-------|
| | <pre>int k, num=30; k = (num>5 ? (num <=10 ? 100 : 200): 500); std::cout<<num; return 0; }</pre> <p>Выберете, что появится на экране (в консоли) в результате выполнения данного кода.</p> <ol style="list-style-type: none"> 1) 100 2) 10 3) 200 4) 500 5) 30 6) 5 7) 0 | |
| 8 | <p>Укажите, сколько параметров можно передать в деструктор (язык программирования C++)</p> <ol style="list-style-type: none"> 1) Не более 3 2) Не более 10 3) Не более 15 4) Максимум 1 5) Нельзя передавать параметры в деструктор | ОПК-2 |
| 9 | <p>Дан код программы на языке программирования C++:</p> <pre>#include <iostream> #include <string> void print(int v) {std::cout << "int:" << v << std::endl;} void print(bool v) {std::cout << "bool:" << v << std::endl;} void print(std::string v) {std::cout << "std::string:" << v << std::endl;} int main() {print(1); print(true); print("Hello world"); return 0; }</pre> <p>Напишите ответ, что появится на экране (в консоли) в результате выполнения данного кода.</p> | ОПК-2 |
| 10 | <p>Задача на соответствие- основные конструкции, использующиеся для построения блок-схем.</p> <p style="text-align: center;"> 1 2 3 </p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="border: 1px solid black; border-radius: 15px; padding: 5px 15px; margin: 5px;">НАЧАЛО</div> <div style="border: 1px solid black; border-radius: 15px; padding: 5px 15px; margin: 5px;">КОНЕЦ</div> <div style="border: 1px solid black; padding: 5px 15px; margin: 5px;">ДЕЙСТВИЕ</div> <div style="border: 1px solid black; width: 40px; height: 20px; margin: 5px; display: flex; justify-content: space-between; align-items: center;"> <div style="width: 10px; height: 100%;"></div> <div style="width: 10px; height: 100%;"></div> </div> </div> <ol style="list-style-type: none"> а) Блок, характеризующий начало/конец алгоритма (для подпрограмм – вызов/возврат); б) Блок - процесс, предназначенный для описания отдельных действий; в) Блок - predetermined процесс, предназначенный для обращения к вспомогательным алгоритмам (подпрограммам) | ОПК-2 |
| 11 | <p>Задача на соответствие. Фамилия- что изобрел.</p> <ol style="list-style-type: none"> 1)Линус Торвальдс 2)Тим Бернерс-Ли 3)Джеймс Гослинг <ol style="list-style-type: none"> а) Linux б) создание HTTP, URL, HTML в) Создатель языка Java | ОПК-2 |
| 12 | <p>Задача на соответствие. Фамилия- что изобрел.</p> | ОПК-2 |

| | | |
|----|--|-------|
| | <p>1)Брендан Эйх 2)Бьёрн Страуструп 3)Марк Цукерберг а) JavaScript б) Создатель языка C++ в) Facebook</p> | |
| 13 | <p>Задача на соответствие.Функция – команда на C++ 1)Синус 2)Косинус 3)Возведение числа в степень 4)Корень квадратный а) $\sin(x)$ б) $\cos(x)$ в) $\text{pow}(x, y)$ г) $\text{sqrt}(x)$</p> | ОПК-2 |
| 14 | <p>Задача. Разработайте алгоритм, который проверяет введенное пользователем значение величины целого типа A, и если оно отрицательное, то увеличивает введенное значение на 2, иначе уменьшает его на 4, а затем выводит на экран полученное значение A. Решение. Проектирование 1 Постановка задачи. Входные данные: величина A целого типа, цел A; Выходные данные: величина A целого типа, цел A; Связь между входными и выходными данными: если $A < 0$ то $A := A + 2$ иначе $A := A - 4$ Напишите алгоритм по действиям</p> | ОПК-2 |
| 15 | <p>Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S. Решение. Проектирование 1 Постановка задачи. Входные данные: величина n целого типа, цел n; Выходные данные: величина S целого типа, цел S; Связь между входными и выходными данными: для i от 1 до n шаг 1 нц $S := S + i$ Кц</p> <p>Поставьте в правильном порядке выполняемые действия Выполняемые действия а) Начало алгоритма б) Описание используемых переменных (целые n, S, i) в) Ввод данных (ввод значения переменной n, тип целый) г) Присваивание значения (переменной S присваивается значение 0) д) Анализ данных (значение переменной i от 1 до n) е) Присваивание значения (переменной S присваивается значение $S + i$) ж) Вывод данных (вывод вычисленного значения переменной величины S, тип целый) з) Конец алгоритма</p> | ОПК-2 |
| 16 | <p>У вас есть проект на языке программирования C++, в котором есть три файла с различными определениями: utility.cpp, lexer.cpp и parser.cpp. Каждому файлу соответствует заголовочный файл с необходимыми объявлениями: utility.hpp, lexer.hpp и parser.hpp. Каждый из файлов</p> | ОПК-2 |

| | | |
|----|---|-------|
| | <p>определений непосредственно (т. е. с помощью директивы include) подключает соответствующий ему заголовочный файл с объявлениями (т. е. utility.cpp подключает utility.hpp, а lexer.cpp подключает lexer.hpp и аналогично для parser.cpp и parser.hpp). Кроме того, файл parser.hpp непосредственно подключает заголовки lexer.hpp и utility.hpp, а файл lexer.hpp подключает заголовок utility.hpp. В этой задаче вам нужно выбрать все верные утверждения из списка. Перед решением задачи полезно нарисовать дерево включений.</p> <ol style="list-style-type: none"> 1) Файл utility.hpp включается только в utility.cpp 2) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp один раз 3) Если в заголовке utility.hpp отсутствует защита от повторного включения, то он будет включен в parser.cpp дважды 4) Файл parser.cpp включает заголовочный файл utility.hpp 5) Файл utility.cpp подключает только один заголовочный файл из перечисленных | |
| 17 | <p>Отметьте все верные утверждения для языка программирования C++.</p> <ol style="list-style-type: none"> 1) Скомпилировать программу на C++ для некоторой архитектуры X можно только на компьютере с архитектурой X. 2) Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 3) Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 4) Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5) Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка. 6) Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка. | ОПК-2 |
| 18 | <p>Укажите существующие модификаторы доступа в языке программирования C++</p> <ol style="list-style-type: none"> 1) protected 2) included 3) private | ОПК-2 |
| 19 | <p>Напишите ответ, где процессор будет искать файл стандартной библиотеки ввода/вывода: #include (язык программирования C, C++)</p> | ОПК-2 |
| 20 | <p>Напишите ответ, чем является конструктор класса в языке программирования C++.</p> <p>Ответ: Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями</p> | ОПК-2 |
| 21 | <p>Укажите, как правильно объявить переменную на языке программирования C++.</p> <ol style="list-style-type: none"> 1) string text="text"; 2) char text[]='text'; 3) char text[]="text"; | ОПК-4 |
| 22 | <p>В языке программирования C++ дан массив <code>int array[5] = { 3, 10, 7, 9, 2}</code>. Напишите ответ, где можно обратиться к числу 7.</p> | ОПК-4 |
| 23 | <p>Дан следующий код на языке программирования C++:</p> <pre>#include <iostream> int main() { for (int i=3; i<8; i++)</pre> | ОПК-4 |

| | | |
|----|--|-------|
| | <pre>{ std::cout return 0; }</pre> <p>Выберете правильный результат кода.</p> <ol style="list-style-type: none"> 1) 3,4,5,6,7, 2) 3,4,5,6,7 3) 3 4 5 6 7 4) 01234567 5) 34567 | |
| 24 | <p>Дан следующий код на языке программирования C++:</p> <pre>#include <iostream> int main() { int x=1; switch(x) { case 1:std::cout << "Один"; case 0:std::cout << "Нуль"; case 3:std::cout << "Всем привет!"; } return 0; }</pre> <p>Напишите результат данного кода.</p> | ОПК-4 |
| 25 | <p>Выберите из списка объявления, которые стоит помещать в заголовочные файлы в языке программирования C++</p> <ol style="list-style-type: none"> 1) void foo() 2) void bar() { foo(); } 3) int a; 4) void foo(); 5) extern int a; | ОПК-4 |
| 26 | <p>Дан следующий код на языке программирования C++.</p> <pre>#include <iostream> int main() { std::cout << "Доброе утро" << "\n"; std::cout <<"Сегодня есть пара?"; return 0; }</pre> <p>Укажите правильный результат данного кода.</p> <ol style="list-style-type: none"> 1) Доброе утро Сегодня есть пара? 2) Доброе утро/nСегодня есть пара? 3) Доброе утро 4) Сегодня есть пара? 5) Доброе утроСегодня есть пара? | ОПК-4 |
| 27 | <p>Дан код программы на языке программирования C++:</p> <pre>#include <iostream> int main() { int k, num=30; k = (num>5 ? (num <=10 ? 100 : 200): 500); std::cout<<num; return 0; }</pre> <p>Напишите ответ, что появится на экране (в консоли) в результате выполнения данного кода.</p> | ОПК-4 |
| 28 | <p>Задача на соответствие- основные конструкции, использующиеся для построения блок-схем</p> | ОПК-4 |

| | | |
|----|---|-------|
| | <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">1 </div> <div style="text-align: center;">2 </div> <div style="text-align: center;">3 </div> <div style="text-align: center;">4 </div> </div> <p>а) Блок - ввода/вывода с неопределенного носителя; б) Блок - ввод с клавиатуры; в) Блок - вывод на монитор; г) Блок - вывод на печатающее устройство.</p> | |
| 29 | <p>Дан код программы на языке программирования C++:</p> <pre>#include <iostream> #include <string> void print(int v) {std::cout << "int:" << v << std::endl;} void print(bool v) {std::cout << "bool:" << v << std::endl;} void print(std::string v) {std::cout << "std::string:" << v << std::endl;} int main() {print(1); print(true); print("Hello world"); return 0; }</pre> <p>Напишите ответ, что появится на экране (в консоли) в результате выполнения данного кода.</p> | ОПК-4 |
| 30 | <p>Задача на соответствие- основные конструкции, используемые для построения блок-схем.</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">1 </div> <div style="text-align: center;">2 </div> <div style="text-align: center;">3 </div> <div style="text-align: center;"></div> </div> <p>а) Блок, характеризующий начало/конец алгоритма (для подпрограмм – вызов/возврат); б) Блок - процесс, предназначенный для описания отдельных действий; в) Блок - предопределенный процесс, предназначенный для обращения к вспомогательным алгоритмам (подпрограммам)</p> | ОПК-4 |
| 31 | <p>Задача на соответствие. Фамилия- что изобрел.</p> <ol style="list-style-type: none"> 1)Линус Торвальдс 2)Тим Бернерс-Ли 3)Джеймс Гослинг <p>а) Linux б) создание HTTP, URL, HTML в) Создатель языка Java</p> | ОПК-4 |
| 32 | <p>Задача на соответствие. Фамилия- что изобрел.</p> <ol style="list-style-type: none"> 1)Брендан Эйх 2)Бьёрн Страуструп 3)Марк Цукерберг <p>а) JavaScript б) Создатель языка C++ в) Facebook</p> | ОПК-4 |
| 33 | <p>Задача на соответствие.Функция – команда на C++</p> <ol style="list-style-type: none"> 1)Синус 2)Косинус 3)Возведение числа в степень 4)Корень квадратный <p>а) sin(x) б) cos(x) в) pow (x, y)</p> | ОПК-4 |

| | | |
|----|---|-------|
| | г) \sqrt{x} | |
| 34 | <p>Задача. Разработайте алгоритм, который проверяет введенное пользователем значение величины целого типа A, и если оно отрицательное, то увеличивает введенное значение на 2, иначе уменьшает его на 4, а затем выводит на экран полученное значение A.</p> <p>Решение.</p> <p>Проектирование</p> <p>1 Постановка задачи.</p> <p>Входные данные: величина A целого типа, цел A; Выходные данные: величина A целого типа, цел A; Связь между входными и выходными данными: если $A < 0$ то $A := A + 2$ иначе $A := A - 4$</p> <p>Поставьте в правильном порядке выполняемые действия</p> <p>Выполняемые действия</p> <p>а) Начало алгоритма б) Описание используемых переменных в) Ввод данных (ввод значения переменной A, тип целый) г) Анализ введенных данных (значение переменной A отрицательное) д) Присваивание значения (переменной A присваивается значение $A + 2$) е) Присваивание значения (переменной A присваивается значение $A - 4$) ж) Вывод данных (вывод полученного значения переменной тип целый) з) Конец алгоритма</p> | ОПК-4 |
| 35 | <p>Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S.</p> <p>Решение.</p> <p>Проектирование</p> <p>1 Постановка задачи.</p> <p>Входные данные: величина n целого типа, цел n; Выходные данные: величина S целого типа, цел S; Связь между входными и выходными данными: для i от 1 до n шаг 1 нц $S := S + i$ Кц</p> <p>Поставьте в правильном порядке выполняемые действия</p> <p>Выполняемые действия</p> <p>и) Начало алгоритма ж) Описание используемых переменных (целые n, S, i) з) Ввод данных (ввод значения переменной n, тип целый) л) Присваивание значения (переменной S присваивается значение 0) м) Анализ данных (значение переменной i от 1 до n) н) Присваивание значения (переменной S присваивается значение $S + i$) о) Вывод данных (вывод вычисленного значения переменной величины S, тип целый) п) Конец алгоритма</p> | ОПК-4 |
| 36 | <p>У вас есть проект на языке программирования C++, в котором есть три файла с различными определениями: <code>utility.cpp</code>, <code>lexer.cpp</code> и <code>parser.cpp</code>. Каждому файлу соответствует заголовочный файл с необходимыми объявлениями: <code>utility.hpp</code>, <code>lexer.hpp</code> и <code>parser.hpp</code>. Каждый из файлов определений непосредственно (т. е. с помощью директивы <code>include</code>) подключает соответствующий ему заголовочный файл с объявлениями (т. е. <code>utility.cpp</code> подключает <code>utility.hpp</code>, а <code>lexer.cpp</code> подключает <code>lexer.hpp</code> и аналогично для <code>parser.cpp</code> и <code>parser.hpp</code>). Кроме того, файл <code>parser.hpp</code></p> | ОПК-4 |

| | | |
|----|---|--------|
| | <p>непосредственно подключает заголовки <code>lexer.hpp</code> и <code>utility.hpp</code>, а файл <code>lexer.hpp</code> подключает заголовок <code>utility.hpp</code>. В этой задаче вам нужно выбрать все верные утверждения из списка. Перед решением задачи полезно нарисовать дерево включений.</p> <ol style="list-style-type: none"> 1) Файл <code>utility.hpp</code> включается только в <code>utility.cpp</code> 2) Если в заголовке <code>utility.hpp</code> отсутствует защита от повторного включения, то он будет включен в <code>parser.cpp</code> один раз 3) Если в заголовке <code>utility.hpp</code> отсутствует защита от повторного включения, то он будет включен в <code>parser.cpp</code> дважды 4) Файл <code>parser.cpp</code> включает заголовочный файл <code>utility.hpp</code> 5) Файл <code>utility.cpp</code> подключает только один заголовочный файл из перечисленных | |
| 37 | <p>Отметьте все верные утверждения для языка программирования C++.</p> <ol style="list-style-type: none"> 1) Скомпилировать программу на C++ для некоторой архитектуры X можно только на компьютере с архитектурой X. 2) Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 3) Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 4) Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5) Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка. 6) Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка. | ОПК-4 |
| 38 | <p>Напишите существующие модификаторы доступа в языке программирования C++</p> | ОПК-4 |
| 39 | <p>Укажите, где процессор будет искать файл стандартной библиотеки ввода/вывода: <code>#include</code> (язык программирования C, C++)</p> <ol style="list-style-type: none"> 1) Во всех папках включения файлов 2) В локальных по отношению к проекту папках включения файлов 3) В глобальных по отношению к проекту папках включения файлов | ОПК-4 |
| 40 | <p>Выберете, чем является конструктор класса в языке программирования C++.</p> <ol style="list-style-type: none"> 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями | ОПК-4 |
| 41 | <p>Укажите, где процессор будет искать файл стандартной библиотеки ввода/вывода: <code>#include</code> (язык программирования C, C++)</p> <ol style="list-style-type: none"> 1) Во всех папках включения файлов 2) В локальных по отношению к проекту папках включения файлов 3) В глобальных по отношению к проекту папках включения файлов | ОПК-14 |
| 42 | <p>Выберете, чем является конструктор класса в языке программирования C++.</p> <ol style="list-style-type: none"> 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями | ОПК-14 |

| | | |
|----|--|--------|
| 43 | Напишите ответ, как правильно объявить одномерный массив с именем array, состоящий из 10 элементов целочисленного типа в языке программирования C++. | ОПК-14 |
| 44 | В языке программирования C++ дан массив <code>int array[5] = { 3, 10, 7, 9, 2}</code> . Напишите ответ, где можно обратиться к числу 7. | ОПК-14 |
| 45 | Выберите из списка объявления, которые стоит помещать в заголовочные файлы в языке программирования C++ 1) <code>void foo()</code> 2) <code>void bar() { foo(); }</code> 3) <code>int a;</code> 4) <code>void foo();</code> 5) <code>extern int a;</code> | ОПК-14 |
| 46 | Дан следующий код на языке программирования C++. <pre>#include <iostream> int main() { std::cout << "Доброе утро" << "\n"; std::cout << "Сегодня есть пара?"; return 0; }</pre> Укажите правильный результат данного кода. 1) Доброе утро Сегодня есть пара? 2) Доброе утро/nСегодня есть пара? 3) Доброе утро 4) Сегодня есть пара? 5) Доброе утроСегодня есть пара? | ОПК-14 |
| 47 | Дан код программы на языке программирования C++: <pre>#include <iostream> int main() { int i=4; int a=12; while(i<5) { a += i; i++; std::cout << a; } return 0; }</pre> Укажите значение переменной «a» после выполнения этого цикла? 1) 4 2) 5 3) 0 4) 12 | ОПК-14 |
| 48 | Дан код программы на языке программирования C++: <pre>#include <iostream> int x=1; int main() { int g=90; g +=2; std::cout << g; return 0; }</pre> Напишите ответ, что появится на экране (в консоли) в результате выполнения данного кода. | ОПК-14 |
| 49 | Напишите ответ, сколько параметров можно передать в деструктор (язык программирования C++) | ОПК-14 |

| | | |
|----|--|--------|
| | <p>Поставьте в правильном порядке выполняемые действия</p> <p>Выполняемые действия</p> <p>a) Начало алгоритма</p> <p>b) Описание используемых переменных (целые n, S, i)</p> <p>c) Ввод данных (ввод значения переменной n, тип целый)</p> <p>d) Присваивание значения (переменной S присваивается значение 0)</p> <p>e) Присваивание значения (переменной i присваивается значение 1)</p> <p>f) Анализ данных (значение переменной i не превосходит n)</p> <p>g) Присваивание значения (переменной S присваивается значение $S + i$)</p> <p>h) Присваивание значения (переменной i присваивается значение $i+1$)</p> <p>i) Вывод данных (вывод вычисленного значения переменной S, тип целый)</p> <p>j) Конец алгоритма</p> | |
| 55 | <p>Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S.</p> <p>Решение.</p> <p>Проектирование</p> <p>1 Постановка задачи.</p> <p>Входные данные: величина n целого типа, цел n;</p> <p>Выходные данные: величина S целого типа, цел S;</p> <p>Связь между входными и выходными данными:</p> <p>выполнять</p> <p>$S := S + i$</p> <p>$i := i + 1$</p> <p>до $i > n$</p> <p>Поставьте в правильном порядке выполняемые действия</p> <p>Выполняемые действия</p> <p>a) Начало алгоритма</p> <p>b) Описание используемых переменных (целые n, S, i)</p> <p>c) Ввод данных (ввод значения переменной n, тип целый)</p> <p>d) Присваивание значения (переменной S присваивается значение 0)</p> <p>e) Присваивание значения (переменной i присваивается значение 1)</p> <p>f) Присваивание значения (переменной S присваивается значение $S + i$)</p> <p>g) Присваивание значения (переменной i присваивается значение $i + 1$)</p> <p>h) Анализ данных (значение переменной i превосходит n)</p> <p>i) Вывод данных (вывод полученного значения переменной S, тип целый)</p> <p>j) Конец алгоритма</p> | ОПК-14 |
| 56 | <p>Задача. Разработайте алгоритм, который вычисляет сумму S первых n натуральных чисел и выводит на экран полученное значение S.</p> <p>Решение.</p> <p>Проектирование</p> <p>1 Постановка задачи.</p> <p>Входные данные: величина n целого типа, цел n;</p> <p>Выходные данные: величина S целого типа, цел S;</p> <p>Связь между входными и выходными данными:</p> <p>для i от 1 до n шаг 1</p> <p>нц</p> <p>$S := S + i$</p> <p>Кц</p> <p>Поставьте в правильном порядке выполняемые действия</p> <p>Выполняемые действия</p> <p>q) Начало алгоритма</p> <p>r) Описание используемых переменных (целые n, S, i)</p> <p>s) Ввод данных (ввод значения переменной n, тип целый)</p> <p>t) Присваивание значения (переменной S присваивается значение 0)</p> | ОПК-14 |

| | | |
|----|--|--------|
| | <p>u) Анализ данных (значение переменной i от 1 до n)</p> <p>v) Присваивание значения (переменной S присваивается значение $S + i$)</p> <p>w) Вывод данных (вывод вычисленного значения переменной величины S, тип целый)</p> <p>x) Конец алгоритма</p> | |
| 57 | <p>У вас есть проект на языке программирования C++, в котором есть три файла с различными определениями: <code>utility.cpp</code>, <code>lexer.cpp</code> и <code>parser.cpp</code>. Каждому файлу соответствует заголовочный файл с необходимыми объявлениями: <code>utility.hpp</code>, <code>lexer.hpp</code> и <code>parser.hpp</code>. Каждый из файлов определений непосредственно (т. е. с помощью директивы <code>include</code>) подключает соответствующий ему заголовочный файл с объявлениями (т. е. <code>utility.cpp</code> подключает <code>utility.hpp</code>, а <code>lexer.cpp</code> подключает <code>lexer.hpp</code> и аналогично для <code>parser.cpp</code> и <code>parser.hpp</code>). Кроме того, файл <code>parser.hpp</code> непосредственно подключает заголовки <code>lexer.hpp</code> и <code>utility.hpp</code>, а файл <code>lexer.hpp</code> подключает заголовок <code>utility.hpp</code>. В этой задаче вам нужно выбрать все верные утверждения из списка. Перед решением задачи полезно нарисовать дерево включений.</p> <ol style="list-style-type: none"> 1) Файл <code>utility.hpp</code> включается только в <code>utility.cpp</code> 2) Если в заголовке <code>utility.hpp</code> отсутствует защита от повторного включения, то он будет включен в <code>parser.cpp</code> один раз 3) Если в заголовке <code>utility.hpp</code> отсутствует защита от повторного включения, то он будет включен в <code>parser.cpp</code> дважды 4) Файл <code>parser.cpp</code> включает заголовочный файл <code>utility.hpp</code> 5) Файл <code>utility.cpp</code> подключает только один заголовочный файл из перечисленных | ОПК-14 |
| 58 | <p>Отметьте все верные утверждения для языка программирования C++.</p> <ol style="list-style-type: none"> 1) Скомпилировать программу на C++ для некоторой архитектуры X можно только на компьютере с архитектурой X. 2) Код программы, написанный на интерпретируемом языке, можно без предварительной компиляции запустить на любой операционной системе, где установлен интерпретатор этого языка. 3) Код программы, написанный на языке, который компилируется в байт код виртуальной машины, достаточно скомпилировать однажды, чтобы программу можно было запускать на любой операционной системе, где есть соответствующая виртуальная машина. 4) Для запуска программы, код которой был написан на компилируемом языке, на компьютере должен быть установлен компилятор этого языка. 5) Код программы, написанный на языке, который компилируется в машинный код, достаточно скомпилировать однажды, и потом программу можно будет запустить на любой операционной системе, для которой существует компилятор этого языка. 6) Для запуска программы, код которой был написан на интерпретируемом языке, на компьютере должен быть установлен интерпретатор этого языка. | ОПК-14 |
| 59 | <p>Укажите существующие модификаторы доступа в языке программирования C++</p> <ol style="list-style-type: none"> 1) <code>protected</code> 2) <code>included</code> 3) <code>private</code> | ОПК-14 |
| 60 | <p>Напишите ответ, где процессор будет искать файл стандартной библиотеки ввода/вывода: <code>#include</code> (язык программирования C, C++)</p> | ОПК-14 |
| 61 | <p>Выберете, чем является конструктор класса в языке программирования C++.</p> <ol style="list-style-type: none"> 1) Метод инициализации экземпляра, который проверяет возможность создания экземпляра и реализует его 2) Способ создания класса | ОПК-14 |

| | | |
|--|---|--|
| | 3) Специальный метод класса, который предназначен для инициализации элементов класса некоторыми начальными значениями | |
|--|---|--|

Перечень тем контрольных работ по дисциплине обучающихся заочной формы обучения, представлены в таблице 19.

Таблица 19 – Перечень контрольных работ

| № п/п | Перечень контрольных работ |
|-------|----------------------------|
| | Не предусмотрено |

10.4. Методические материалы, определяющие процедуры оценивания индикаторов, характеризующих этапы формирования компетенций, содержатся в локальных нормативных актах ГУАП, регламентирующих порядок и процедуру проведения текущего контроля успеваемости и промежуточной аттестации обучающихся ГУАП.

11. Методические указания для обучающихся по освоению дисциплины

11.1. Методические указания для обучающихся по освоению лекционного материала.

Основное назначение лекционного материала – логически стройное, системное, глубокое и ясное изложение учебного материала. Назначение современной лекции в рамках дисциплины не в том, чтобы получить всю информацию по теме, а в освоении фундаментальных проблем дисциплины, методов научного познания, новейших достижений научной мысли. В учебном процессе лекция выполняет методологическую, организационную и информационную функции. Лекция раскрывает понятийный аппарат конкретной области знания, её проблемы, дает цельное представление о дисциплине, показывает взаимосвязь с другими дисциплинами.

Планируемые результаты при освоении обучающимися лекционного материала:

- получение современных, целостных, взаимосвязанных знаний, уровень которых определяется целевой установкой к каждой конкретной теме;
- получение опыта творческой работы совместно с преподавателем;
- развитие профессионально-деловых качеств, любви к предмету и самостоятельного творческого мышления;
- появление необходимого интереса, необходимого для самостоятельной работы;
- получение знаний о современном уровне развития науки и техники и о прогнозе их развития на ближайшие годы;
- научиться методически обрабатывать материал (выделять главные мысли и положения, приходить к конкретным выводам, повторять их в различных формулировках);
- получение точного понимания всех необходимых терминов и понятий.

Лекционный материал может сопровождаться демонстрацией слайдов и использованием раздаточного материала при проведении коротких дискуссий об особенностях применения отдельных тематик по дисциплине.

Выделяются следующие виды лекций:

- Вводная лекция

Вводная лекция к дисциплине знакомит обучающихся с целью и назначением курса, его ролью и местом в системе дисциплин. В ходе такой лекции связывается теоретический и практический материал с практикой будущей работы, рассказывается общая методика работы над курсом, предлагаются литературные источники, помогающие усвоению материала дисциплины и освоению компетенций, ставятся научные проблемы,

выдвигаются гипотезы, определяется форма текущего контроля и промежуточной аттестации.

Вводная лекция к разделу. Аналогично вводной лекции к дисциплине раскрывает ряд вопросов, но связанных не с дисциплиной в целом, а с тематикой конкретного раздела.

- Обзорная лекция

Проводится с целью систематизации знаний на более высоком уровне, рассмотрения особо трудных вопросов дисциплины.

- Проблемная лекция

На данной лекции новое знание вводится как неизвестное, которое необходимо "открыть". В рамках лекции создается проблемная ситуация, которую обучающие решают поэтапно с подсказками и помощью преподавателя.

- Лекция вдвоем

Эта разновидность лекции является продолжением и развитием проблемного изложения материала в диалоге двух преподавателей. Здесь моделируются реальные ситуации обсуждения теоретических и практических вопросов двумя специалистами.

- Лекция с заранее запланированными ошибками

Данная лекция призвана активизировать внимание студентов, развивать их мыслительную деятельность, формировать умение выступать в роли экспертов.

Задача преподавателя состоит в том, чтобы заложить в лекцию определенное количество ошибок содержательного, методического, поведенческого характера. Подбираются наиболее типичные ошибки, которые обычно не выпячиваются, а как бы затушевываются. Задача студентов состоит в том, чтобы по ходу лекции отмечать ошибки, фиксировать и называть их в конце.

- Лекция-пресс-конференция

Преподаватель просит студентов задавать письменно вопросы по данной теме. В течение двух-трех минут студенты формулируют наиболее интересующие их вопросы и передают преподавателю, который в течение трех-пяти минут сортирует вопросы по их содержанию и начинает лекцию. Лекция излагается не как ответы на вопросы, а как связный текст, в процессе изложения которого формируются ответы.

- Лекция-консультация

Материал излагается в виде вопросов и ответов или вопросов, ответов и дискуссий.

Структура предоставления лекционного материала:

- Вводная часть лекции

Первое представление о лекции содержится уже в формулировке темы. Она должна быть краткой, выражать суть основной идеи, быть привлекательной по форме. Целесообразно здесь сказать на значение этой темы для последующего усвоения знаний и развития личности студентов, для будущей профессиональной деятельности. Далее можно сообщить цели лекции и ее план. Желательно сориентировать слушателей на последующий контроль знаний, полезно указать на связь нового материала с пройденным и предыдущим. Темп изложения этой части лекции, как правило, должен быть выше темпа изложения основного, что заставляет студентов психологически собраться и сосредоточиться. Вводная часть лекции обычно занимает 5-7 минут.

- Основная часть лекции

Переходу к изложению первого вопроса, как правило, должна предшествовать пауза. В это время лектор может проверить, все ли слушатели готовы к восприятию лекции (позы, выражения лиц, разговоры). Заметив студентов, не готовых к восприятию, опытные преподаватели произносят краткую мобилизующую фразу, останавливают взгляд на нерадивых, реже - называют фамилию, имя и не тратят время на длительные замечания.

Для того чтобы преодолеть потенциальную пассивность слушателей, необходимо всеми возможными способами придать лекции проблемный характер, побуждая слушателей к самостоятельной познавательной активности и творчеству.

К таким активным средствам можно отнести:

- обращение к студентам с вопросами, уточняющими понимание основных идей и фактов темы;
 - организацию мини-столкновений различных точек зрения по выдвинутым преподавателем положениям;
 - постановку вопросов, задач с множественностью решений и др.;
 - индивидуальный стиль изложения материала;
 - обеспечение обратной связи.
- Заключение

В процессе чтения лекции преподаватель должен позаботиться о ее завершении. Рассчитать время, а не прерывать лекцию на полуслове. Обычно для заключения материала бывает достаточно 5-7 минут. Завершая лекцию, преподаватель отвечает на вопросы слушателей, подводит итог, дает методические указания к самостоятельной работе, комментирует предлагаемую литературу. Заканчивать лекцию нужно конструктивно по содержанию и положительно по эмоциональному настрою. Студенты должны уйти заинтересованными, заинтригованными, желающими опробовать завтра же предложения лектора, а также в хорошем настроении и активном тоне.

11.2. Методические указания для обучающихся по выполнению лабораторных работ.

В ходе выполнения лабораторных работ обучающийся должен углубить и закрепить знания, практические навыки, овладеть современной методикой и техникой эксперимента в соответствии с квалификационной характеристикой обучающегося. Выполнение лабораторных работ состоит из экспериментально-практической, расчетно-аналитической частей и контрольных мероприятий.

Выполнение лабораторных работ обучающимся является неотъемлемой частью изучения дисциплины, определяемой учебным планом, и относится к средствам, обеспечивающим решение следующих основных задач обучающегося:

- приобретение навыков исследования процессов, явлений и объектов, изучаемых в рамках данной дисциплины;
- закрепление, развитие и детализация теоретических знаний, полученных на лекциях;
- получение новой информации по изучаемой дисциплине;
- приобретение навыков самостоятельной работы с лабораторным оборудованием и приборами.

Задание и требования к проведению лабораторных работ.

Задания и требования к лабораторным работам размещены в Личном кабинете ГУАП в разделе дисциплины.

Структура и форма отчета о лабораторной работе.

Отчет о лабораторной работе сдается в электронном виде (документ Word, документ PDF) через Личный кабинет ГУАП. Отчет к лабораторной работе содержит следующие элементы:

- титульный лист с названием дисциплины, номером и названием лабораторной работы;
- цели и задачи работы;
- приборы и реактивы (при необходимости);
- задание;
- ход работы (при необходимости);
- контрольные примеры (при необходимости);
- выводы;

Требования к оформлению отчета о лабораторной работе.

– Общие требования и рекомендации по выполнению письменных работ : методические указания / С.-Петербург. гос. ун-т аэрокосм. приборостроения ; сост. А. А. Сорокин. - СПб. : Изд-во ГУАП, 2017. - 32 с.

– Общие требования и рекомендации по выполнению письменных работ : методические указания (с изменениями от 09.01.2019) [Электронный ресурс] / Ивангородский филиал С.-Петербург. гос. ун-т аэрокосм. приборостроения ; сост. А. А. Сорокин. - Ивангород : 2019. - 37 с. URL: <http://ifguap.ru/rp/ReportsFormattingRules.pdf>, Личный кабинет ГУАП

11.3 Методические указания для обучающихся по прохождению самостоятельной работы

В ходе выполнения самостоятельной работы, обучающийся выполняет работу по заданию и при методическом руководстве преподавателя, но без его непосредственного участия.

Для обучающихся по заочной форме обучения, самостоятельная работа может включать в себя контрольную работу.

В процессе выполнения самостоятельной работы, у обучающегося формируется целесообразное планирование рабочего времени, которое позволяет им развивать умения и навыки в усвоении и систематизации приобретаемых знаний, обеспечивает высокий уровень успеваемости в период обучения, помогает получить навыки повышения профессионального уровня.

Методическими материалами, направляющими самостоятельную работу обучающихся являются:

- учебно-методический материал по дисциплине.

11.4 Методические указания для обучающихся по прохождению текущего контроля успеваемости.

Текущий контроль успеваемости предусматривает контроль качества знаний обучающихся, осуществляемого в течение семестра с целью оценивания хода освоения дисциплины.

Возможные методы текущего контроля:

- устный опрос на занятиях;
- систематическая проверка выполнения индивидуальных и домашних заданий;
- защита отчетов по лабораторным работам;
- проведение контрольных работ;
- тестирование;
- контроль самостоятельных работ;
- проведение контрольных работ;
- доклад на научной конференции;
- написание научной статьи.

11.5 Методические указания для обучающихся по прохождению тестирования.

Использование тестовых заданий возможно как при текущем контроле, так и при проведении промежуточной аттестации. Тесты могут проводиться как в письменной форме, так и с использованием электронных средств обучения.

Можно выделить основные уровни теста, в которых проверка возрастает от контроля знаний (индикатор достижения компетенции - "знать") до применения навыков при решении типовых и нетиповых задач ((индикаторы достижения компетенции - "уметь" и "владеть")):

- Первый уровень - узнавание ранее изученного материала;

- Второй уровень - репродуктивный - в заданиях не содержится материала для ответа или же его извлечение требует не только запоминания материала, но и его понимания (подстановка, конструктивный тест, типовая задача);
- Третий уровень - нетиповые задачи повышенной сложности, для которых требуется самостоятельное нахождение методов решения;
- Смешанный - использование элементов всех трех уровней для проверки разных индикаторов достижения компетенций.

Критерии оценки тестовых работ базируются на 100-бальной шкале согласно МДО ГУАП. СМК 2.77 "Положение о модульно-рейтинговой системе оценки качества учебной работы студентов в ГУАП" (допустимо применение любого количественного показателя оценки с приведением его к 100-процентной шкале):

- менее 55 - "не зачтено" или "неудовлетворительно" (2);
- от 55 до 69 - "зачтено" или "удовлетворительно" (3);
- от 70 до 84 - "зачтено" или "хорошо" (4);
- от 85 до 100 - "зачтено" или "отлично" (5).

11.6 Методические указания для обучающихся по прохождению промежуточной аттестации.

Промежуточная аттестация обучающихся предусматривает оценивание промежуточных и окончательных результатов обучения по дисциплине. Она включает в себя:

- экзамен – форма оценки знаний, полученных обучающимся в процессе изучения всей дисциплины или ее части, навыков самостоятельной работы, способности применять их для решения практических задач. Экзамен, как правило, проводится в период экзаменационной сессии и завершается аттестационной оценкой «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Экзамен проводится в письменной форме в виде теста.

В случае дистанционной формы промежуточной аттестации, экзамен проводится в виде теста с применением средств электронного обучения.

Выполнение обучающимся лабораторных работ не в полном объеме может привести к понижению оценки за дисциплину из-за низкого уровня освоения компетенций:

- выполнение менее 75% лабораторных работ - понижение максимальной оценки на 1 балл;
- выполнение менее 50% лабораторных работ - понижение максимальной оценки на 2 балла;
- невыполнение лабораторных работ - понижение максимальной оценки на 3 балла.

Лист внесения изменений в рабочую программу дисциплины

| Дата внесения изменений и дополнений. Подпись внесшего изменения | Содержание изменений и дополнений | Дата и № протокола заседания кафедры | Подпись зав. кафедрой |
|---|-----------------------------------|--------------------------------------|-----------------------|
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |