

* + - 1. Цели освоения дисциплины

**Цель дисциплины**:

* является формирование общей информационной культуры студентов, подготовка их к деятельности, связанной с использованием современных информационных технологий;
* подготовка специалистов к деятельности, связанной с разработкой программного обеспечения для решения профессиональных задач;
* формирование теоретических знаний, освоение методов и средств программирования практических инженерных задач на современных пакетах объектно-ориентированного языка, подготовка специалистов к деятельности, связанной с разработкой и эффективной реализацией программных средств для решения профессиональных задач

Основные задачи дисциплины:

* знакомство с основными парадигмами и теоретическими основами программирования, терминологией объектно-ориентированного программирования;
* изучение принципов анализа синтаксиса и семантики;
* изучение формальных способов описания языков программирования;
* изучение типов данных;
* изучение линейного, ветвящегося и циклического алгоритмов;
* научиться составлять и читать блок-схемы;
* изучение способов и механизмов управления данными;
* изучение методов и основных этапов трансляции;
* практическое освоение реализации изученных алгоритмов;
* получение навыков работы с различными языками программирования;
* выработка навыков программирования и алгоритмизации с применением современных процедурных и объектно-ориентированных языков программирования.

.

* + - 1. Место дисциплины в структуре ОП

Дисциплина «Языки программирования» читается студентам 1, 2 курсов и относится к числу дисциплин базовой части Блока1 Дисциплины (Модули).

 Для успешного усвоения дисциплины «Языки программирования» необходимо, чтобы студент владел знаниями, умениями и навыками, сформированными в школе при изучении: математики, английского языка и информатики.

Параллельно с данной дисциплиной идет изучение таких взаимосвязанных с

«Языками программирования» с дисциплин как:

* «Физика»
* «Линейная алгебра»
* «Аналитическая геометрия»
* «Математический анализ»
* «Дискретная математика»
* «Иностранный язык»

 «Информатика»

* «Аппаратные средства вычислительной техники»

Дисциплина «Языки программирования» обеспечивает изучение следующих дисциплин:

* Интернет-технологии»
* «Информационные технологии»
* «Операционные системы»
* «Моделирование систем и сетей телекоммуникаций»;
* «Геоинформационные технологии в телекоммуникационных системах»
* «Сетевое администрирование»
* «Защищенные корпоративные сети»

Дисциплина «Языки программирования» необходима для подготовка курсовых работ и дипломного проектирования

* + - 1. Компетенции обучающегося, формируемые в результате освоения

**дисциплины.**

Процесс изучения дисциплины направлен на формирование следующих компетенций:

|  |  |
| --- | --- |
| **Код компетенции** | **Компетенция** |
| ОПК-4 | способностью понимать значение информации в развитии современного общества, применять достижения информационных технологий для поиска иобработки информации |
| ОПК-5 | способностью применять программные средства системного и прикладногоназначения, языки, методы и инструментальные средства программирования для решения профессиональных задач |

В результате освоения компетенций в рамках дисциплины обучающийся должен:

|  |  |
| --- | --- |
| **Код компетенции** | **Результаты обучения** |
| ОПК-4 | Знать:* общие принципы построения и использования современных языков программирования высокого уровня;
* парадигмы и методологии программирования;
* особенности наиболее распространенных объектно-ориентированных языков программирования;
* базовые структуры данных;
* основные комбинаторные и теоретико-графовые алгоритмы;
* современные технологии программирования

Уметь:* формализовать поставленную задачу;
* работать с интегрированными средами разработки программного обеспечения;
* применять в профессиональной деятельности современные объектно-ориентированные языки программирования;
* профессионально решать задачи производственной и технологической деятельности с учетом современных достижений науки и техники;
* оценивать качество готового программного обеспечения

Владеть:* навыками постановки математических и информационных задач;
* навыками описания алгоритмов решения поставленной задачи и разработки программного кода;
* навыками тестирования программного обеспечения; разработки алгоритмов решения типовых профессиональных задач
 |
| ОПК–5 | Знать:* основы и стандарты разработки алгоритмов вычислительных процессов;
* методы составления алгоритмов;
* методы формализации алгоритмов;
* основные приемы реализации алгоритмов на языках структурного программирования;
* основные приемы программирования,о бласти и особенности применения языков объектно-ориентированного программирования (ООП).
* паттерны проектирования;
* алгоритмы сортировки и поиска данных;
* методы тестирования и отладки ПО. Уметь:
* реализовывать на языке высокого уровня алгоритмы решения профессиональных задач, в том числе задач обработки потоков;
* работать с интегрированной средой разработки программного обеспечения;
* реализовывать на языке высокого уровня алгоритмы решения профессиональных задач, в том числе задач обработки потоков.

Владеть:* навыками обеспечения безопасности информации с помощью типовых программных средств (антивирусов, архиваторов, стандартных сетевых средств обмена информацией).
* навыками разработки, документирования, тестирования и отладки

программ |

Основные признаки проявленности формируемых компетенций в результате освоения дисциплины «Языки программирования» сведены в таблице.

Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания

|  |  |  |
| --- | --- | --- |
| **Уровень освоения компетенции** | **Результат обучения** | **Результат обучения** |
| ОПК-4: Знать, уметь, владеть | ОПК-5: Знать, уметь, владеть |
| минимальный | Способен критическуюматериала | дать оценку | собственную изучаемого | Владеет основными навыками работы с источниками и критическойлитературой |
| Может соотнести основные идеи с современными проблемами | Способенпроблему процессами | в | представить ее связи | с | ключевую другими |
| Способен выделить характерныйавторский подход | Понимает специфику основныхрабочих категорий |
| базовый | Способен сравнивать концепции, аргументированно излагает материал | Свободно излагает материал, однако не демонстрирует навыков сравненияосновных идей и концепций |
| Аргументированно проводит сравнение концепций по заданной проблематике | Способен выделить и сравнитьконцепции, но испытывает сложности с их практической привязкой |
| Способенконцепций в области | выделить заданной | специфику проблемной | Знает основные отличия концепций в заданной проблемной области |
| продвинутый | Способен грамотно обосновать собственную позицию относительно решения современных проблем взаданной области | Видит источники современных проблем в заданной области анализа, владеет подходами к их решению |
| Свободно ориентируется в заданной области анализа. Понимает ее основания и умеет выделитьпрактическое значение заданной области | Выявляет основания заданной области анализа, понимает ее практическую ценность, однако испытываетзатруднения в описании сложных объектов анализа |
| Может дать критический анализ современным проблемам в заданной области анализа | Знает основное содержание современных научных идей в рабочей области анализа, способен ихсопоставить |

**Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания**

|  |  |
| --- | --- |
| Этап (уровень) освоениякомпетенции | Основные признаки проявленности компетенции (дескрипторное описание уровня) |
| 1. | 2. | 3. | 4. | 5. |
| минимальный | не владеет | слабо ориентируетсяв терминологии и содержании | Способен выделить основные идеитекста, работает с критической литературой | Владеет основными навыками работы с источниками и критической литературой | Способен дать собственнуюкритическую оценку изучаемого материала |
| не умеет | не выделяетосновные идеи | Способен показать основную идею вразвитии | Способен представить ключевую проблемув ее связи с другими процессами | Может соотнести основные идеи ссовременными проблемами |
| не знает | допускает грубые ошибки | Знает основные рабочие категории, однако не ориентируется в ихспецифике | Понимает специфику основных рабочих категорий | Способен выделить характерный авторский подход |
| базовый | не владеет | плохо ориентируетсяв терминологии и содержании | Владеет приемами поиска исистематизации, но не способен свободно изложить материал | Свободно излагает материал, однако недемонстрирует навыков сравнения основных идей и концепций | Способен сравнивать концепции,аргументированно излагает материал |
| не умеет | выделяет основныеидеи, но не видит проблем | Выделяет конкретную проблему, однако излишне упрощает ее | Способен выделить и сравнить концепции,но испытывает сложности с их практической привязкой | Аргументированно проводитсравнение концепций по заданной проблематике |
| не знает | допускает много ошибок | Может изложить основные рабочие категории | Знает основные отличия концепций в заданной проблемной области | Способен выделить специфику концепций в заданной проблемнойобласти |
| продвинутый | не владеет | ориентируется в терминологии и содержании | В общих чертах понимает основную идею, однако плохо связывает ее с существующей проблематикой | Видит источники современных проблем в заданной области анализа, владеет подходами к их решению | Способен грамотно обосновать собственную позициюотносительно решениясовременных проблем в заданной области |
| не умеет | выделяет основные идеи, но не видит их в развитии | Может понять практическое назначение основной идеи, но затрудняется выявить ее основания | Выявляет основания заданной области анализа, понимает ее практическую ценность, однако испытывает затруднения в описании сложных объектов анализа | Свободно ориентируется в заданной области анализа. Понимает ее основания и умеетвыделить практическое значение заданной области |
| не знает | допускает ошибки при выделении рабочей областианализа | Способен изложить основное содержание современных научных идей в рабочей области анализа | Знает основное содержание современных научных идей в рабочей области анализа, способен их сопоставить | Может дать критический анализ современным проблемам в заданной области анализа |

* + - 1. Структура и содержание дисциплины

Общая трудоемкость (объем) дисциплины (модуля) составляет 7 зачетных единиц (ЗЕ\*), 252 академических часа.

Год набора: 2020

|  |  |  |
| --- | --- | --- |
| **Объём дисциплины** | **Всего часов** | **Часов по семестрам** |
| **1 семестр** | **2 семестр** |
| **Общая трудоёмкость дисциплины** | 252 | 144 | 108 |
| **Контактная работа обучающихся с преподавателям (по видам аудиторных учебных занятий) – всего:** | 98 | 56 | 42 |
| в том числе: |  |  |  |
| лекции | 42 | 28 | 14 |
| Лабораторные занятия | 56 | 28 | 28 |
| **Самостоятельная работа** (СРС) – всего: | 154 | 88 | 66 |
| **Вид промежуточной аттестации (зачет/экзамен)** | Экзамен | Экзамен | Экзамен |

* + - * 1. Содержание разделов дисциплины

Языки программирования

Введение

Начальные сведения о языках программирования. Роль языков программирования. Определение целей создания программы. Характеристики и свойства языков программирования. История развития языков программирования. Поколения языков.

Особенности языка С

Этапы разработки программы. Структурные особенности языка С.

Основные элементы языка программирования. Инструментальные средства разработки программ на языках высокого уровня в различных операционных системах. Ключевые слова и лексемы

Особенности языка Python

Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Базовые понятия языка С: Основные операции и конструкции

Алфавит, идентификаторы, служебные слова, константы, комментарии. Структура программы. Этапы создания исполняемой программы. Типизация языка. Определение типа. Контроль типов. Уровни типизации. Эквивалентность типов. Простые типы данных. Переменные. Классы памяти. Операции, выражения, операторы.

Операторы условия. Операторы организации циклов. Вложенные циклы.

Операторы передачи управления.

***Базовые понятия языка Python: Типы данных, условия***

Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Выражения. Операции. Порядок выполнения операций. Математические функции. Композиция. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор.

Массивы, функции пользователя в языке С

Указатели. Ссылки. Одномерные и многомерные массивы. Строки. Связь указателей и массивов. Способы инициализации числовых и текстовых массивов. Базовые алгоритмы и программы для обработки одномерных и двумерных массивов. Динамические массивы.

Шаблоны функций. Определение и вызов функций пользователей. Способы передачи параметров в функцию. Рекурсивные функции. Подставляемые функции. Перегрузка функций. Указатели на функции. Аргументы по умолчанию. Передача массивов в функции. Типы данных, определяемые пользователем

Понятие структуры, примеры использования структур, операции над структурами. Объединения, перечислимый тип. Определение размера памяти в байтах (операция sizeof). Динамическое выделение памяти. Динамические структуры.

Объединения, поля битов, доступ к отдельному биту.

***Базовые понятия языка Python: циклы, случайные числа***

Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random.

Препроцессорные средства, Обработка файлов в С

Стадии препроцессорной обработки. Директивы. Макроподстановки.

Условная компиляция. Препроцессоры и директивы условной компиляции.

Понятие потока. Открытие и закрытие файлов. Операции, функции ввода- вывода результатов вычислений и текстов.

Функции в Python

Создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала.

***Строки - последовательности символов в Python***

Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи).

Классы памяти и области видимости данных в С

Классы памяти. Описатели классов памяти. Модели памяти. Распределение памяти при работе программы. Управление экраном и памятью в текстовом режиме.

***Сложные типы данных в Python***

Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range.

Динамическое распределение памяти в С

Функции для динамического распределения памяти. Понятие очереди, Линейные списки, Двусвязные списки. Операции над списками. Методы хранения списков. Стеки и их обработка.

***Стиль программирования и отладка программ в Python***

Стиль программирования. Отладка программ. Стиль кода PEP 8.

Расширенное представление данных в С

Абстрактные типы данных. Получение абстрактного типа. Создание интерфейса. Использование интерфейса. Реализация интерфейса представления данных. Имитация реальных ситуаций с помощью очереди. Деревья двоичного поиска.

***Файлы и исключения в Python***

Работа с файлами. Чтение и запись. Исключения в Python. Конструкция try - except - else – finally.

Основные принципы объектно-ориентированного

программирования в С

Основные понятия объектно-ориентированного программирования (ООП). Отличия ООП от процедурного программирования. Понятие класса и объекта. Определение шаблона класса. Использование шаблона класса. Правила описания шаблонов. Специализация шаблона класса.

Понятия наследования и полиморфизма.

Технологии объектно-ориентированного программирования в С

Концепция и основные правила перегрузки операций. Перегрузка операций методами класса, дружественными и обычными функциями. Перегрузка различных операций.

Наследование и Полиморфизм в С

Механизм наследования. Базовые и производные классы. Типы наследования. Правила наследования различных методов. Множественное наследование. Виртуальные и чистые виртуальные функции. Абстрактные классы. Статическое и динамическое связывание. Механизм динамического связывания. Виртуальные деструкторы.

Объекты и классы в С

Спецификаторы доступа к членам класса. Объявление или спецификация класса. Доступ к членам класса. Типы конструкторов. Вызов конструкторов. Статистические данные класса. Константные методы класса. Конструкторы и деструкторы. Дружественные классы и функции. Статические элементы класса. Константные методы и объекты. Указатель this. Указатели и ссылки на объекты, массивы объектов.

Потоки данных в С

Классификация способов ввода – вывода данных. Принципы работы с потоками и файлами. Стандартные классы потоков. Форматированный ввод-вывод базовых типов. Манипуляторы ввода и вывода данных Файловые потоки данных.

* + - * 1. Лабораторные занятия, их содержание

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № п/п | № раздела дисциплины | Тема занятия | Форма проведения | Формируемые компетенции |
|  | 1 | Знакомство с языком Python, общие сведения о языке, переменные и выражения, базовые операции | Лабораторная | ОПК-4, ОПК-5 |
|  | 1 | Линейный вычислительный процесс в Python | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Программирование алгоритмов с ветвлениями в Python | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Циклы в Python | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Функции в Python | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Работа с массивами, кортежами и списками в Python | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Открытие, чтение и запись файлов в Python. Исключения | Лабораторная | ОПК-4, ОПК-5 |
|  | 1 | Программирование алгоритмовлинейной структуры в Си | Лабораторная | ОПК-4, ОПК-5 |
|  | 1 | Программирование алгоритмовразветвляющейся структуры в Си | Лабораторная | ОПК-4, ОПК-5 |
|  | 1 | Программирование алгоритмовциклической структуры в Си | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Базовые операции языка Си | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Организация вычислительныхпроцессов | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Условные операторы и операторывыбора | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Обработка одномерных массивов | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Обработка двумерных массивов | Лабораторная | ОПК-4, ОПК-5 |
|  | 2 | Функции пользователя идинамическое распределение памяти | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Организация работы с файлами | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Обработка списков | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Вывод графиков функций | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Обработка списков в С++ | Лабораторная | ОПК-4, ОПК-5 |
|  | 3 | Вывод графиков функций в С++ | Лабораторная | ОПК-4, ОПК-5 |

* + - 1. **Учебно-методическое обеспечение самостоятельной работы студента и оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины**
	1. **Текущий контроль**

Текущий контроль производится путем защиты лабораторных работ в каждом семестре.

В конце каждого семестра проводится тестирование по изученным тема дисциплины. Сами тесты представлены в Системе управления курсами РГГМУ (http://moodle.rshu.ru)/

Критерии оценивания лабораторных работ.

* оценка «зачтено»: работа полностью выполнена. Даны полные ответы на вопросы по теме работы;
* оценка «не зачтено»: работа не выполнена или при защите студент не может ясно и четко ответить на поставленные вопросы.
	1. Методические указания по организации самостоятельной работы

Самостоятельная работа связана с изучением и конспектированием отдельных вопросов лекционного материала, выделенного преподавателем. Для успешного выполнения самостоятельной работы необходимо:

* + - в соответствии с заданной темой проработать соответствующий лекционный материал;
		- прочитать литературу из рекомендованного списка;
		- при необходимости осуществить поиск нужной информации в сети. Контроль выполнения самостоятельной работы обучающегося осуществляется собеседованием по определению понимания изученного материала.

***Контроль исполнения*** самостоятельных работ осуществляется преподавателем с участием студентов в форме обсуждения выполненных заданий.

* 1. Промежуточный контроль: экзамен (1 и 2 семестры)

Перечень вопросов для промежуточной аттестации – экзамен (Раздел 1, 1семестр):

1. Какие стандартные функции содержатся в файлах stdio.h и conio.h ?
2. Что означает «инкремент»?
3. Что означает «декремент»?
4. Что означает операция присваивания?
5. Какой спецификатор применяется для ввода и вывода строки символов?
6. Назовите основные операторы циклических процессов.
7. Назовите основные параметры цикла.
8. Как образуется бесконечный цикл и как выйти из него?
9. Какие управляющие средства выполнения программ существуют в языке С? Объясните их синтаксис.
10. Объясните назначение всех функций, использованных в программах.
11. Какие операции используются в управляющих средствах выполнения

программ.

1. Объясните различия между циклами while и do … while.
2. Для каких целей используется оператор switch ?
3. С какой целью используется индексация массивов?
4. С какой целью используется тип данных?
5. Опишите примененный Вами алгоритм обработки массива.
6. Как определить адрес (i,j) – того элемента в памяти машины?
7. Как распределяется двумерный массив в памяти машины?
8. Что означает транспонирование матрицы?
9. Как поменять местами строку и столбец?
10. Какие циклы называются вложенными?
11. Какой переход называется условным / безусловным?
12. Для чего используется оператор #define?
13. Назовите операторы условной компиляции.
14. В чем состоит отличие объявление функции от ее определения?
15. Как объявить функцию с переменным числом параметров?
16. Какие типы объектов могут быть использованы в качестве формальных параметров?
17. Что такое локальные объекты? Какова их область видимости и «время жизни»?
18. В чем состоит отличие автоматических переменных от статических?
19. Объясните механизм передачи параметров по значению, по указателю, по ссылке.
20. Встроенный тип str. Методы объекта str.
21. Что такое print()? форматирование вывода.
22. Как можно принять результат ввода на клавиатуре?
23. Встроенные типы чисел — int, float, complex. Машинное представление чисел с плавающей точкой и целых. Преобразование типов при сравнении чисел.
24. Как работать с числами, которые не входят в десятичную систему счисления?
25. Как можно конвертировать строку в число?
26. Расскажите про арифметические операторы //, %, и \*\*.
27. Опишите инструкции if/else/elif, логические операторы и выражения сравнения. Приведите примеры.
28. Опишите циклы while и for в Python, приведите примеры.
29. Сколько аргументов может принять range()?
30. Функции в Python. Основные понятия и примеры.
31. Передачу аргументов в функцию. Специальные режимы сопоставления аргументов.
32. Что такое рекурсия?
33. Объясните как работают операторы для списков: s.append(x), s.count(x), s.index(x), i.insert(i,j), s.remove(x), s.reverse(), s.sort()
34. Объясните как работают операторы для словарей: len(a), a[k], a[k] = x, del a[k], a.items(), a.keys(), a.values(), a.has\_key(k)
35. Объясните как работают операторы с файловыми объектами: f.close(), f.read([size]), f.readline(), f.readlines(), f.seek(offset, mode), f.tell(), f.write(str)
36. Объясните как работают операторы для импортирования модулей на языке Python. import <модуль1> [, <модуль2> ]\*, from <модуль> import <имя1> [, <имя2> ]\*, from <модуль> import \*
37. Объясните, как работают операторы для работы со строками: index (s, sub, i=0), lower (s), upper (s), strip (s), splitfields (s, sep)
38. Объясните, как работают операторы: float (x), hex (x), int (x), long (x), ord (c), pow (x, y), str (<объект>)
39. Как можно переключить регистр строки?
40. Запишите как выглядят на языке Python арифметические операторы: умножения, деления, сложения, вычитания, деления на цело, возведения в степень.
41. Покажите, как на языке Python объявляется одномерный список и заполняется нулевыми значениями.
42. Покажите, как на языке Python объявляется двумерный список и заполняется нулевыми значениями.
43. Покажите, как на языке Python можно организовать два вложенных друг в друга цикла со счетчиками.
44. Покажите сокращенную форму сложения, вычитания, умножения и деления на языке Python.
45. Запишите функцию ввода на языке Python. Объясните для чего она используется. Приведите пример.
46. Операторы break и continue. Для чего они используются. Приведите примеры.
47. Что такое РЕР8?
48. Как лучше всего поменять местами числовые значения объектов?
49. Как можно выполнить несколько операций присвоения в одном предложении?

Образец билета:

Российский государственный гидрометеорологический университет Кафедра информационных технологий и систем безопасности ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

Дисциплина «Языки программирования»

1. Для чего используется оператор #define?
2. Функции в Python. Основные понятия и примеры.
3. Напишите программу вычисляющую факториал введенного числа.

«Утверждаю»

Зав. кафедрой / /

Критерии выставления оценки

Оценка «**отлично**» ставится студенту, ответ которого содержит:

* + глубокое знание программного материала, а также основного содержания и новаций лекционного курса но сравнению с учебной литературой;
	+ знание концептуально-понятийного аппарата всего курса; а также свидетельствует о способности:
	+ самостоятельно критически оценивать основные положения курса;
	+ увязывать теорию с практикой.

Оценка «отлично» не ставится в случаях систематических пропусков студентом семинарских и лекционных занятий по неуважительным причинам, а также неправильных ответов на дополнительные вопросы преподавателя.

Оценка «**хорошо**» ставится студенту, ответ которого свидетельствует о полном знании материала по программе, а также содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

Оценка «хорошо» не ставится в случаях пропусков студентом семинарских и лекционных занятий по неуважительным причинам.

Оценка «**удовлетворительно**» ставится студенту, ответ которого содержит:

* + поверхностные знания важнейших разделов программы и содержания лекционного курса;
	+ затруднения с использованием научно-понятийного аппарата и терминологии курса;
	+ стремление логически четко построить ответ, а также свидетельствует о возможности последующего обучения.

Оценка «**неудовлетворительно**» ставится студенту, имеющему существенные пробелы в знании основного материала по программе, а также допустившему принципиальные ошибки при изложении материала.

Перечень вопросов для промежуточной аттестации экзамен (Разделы 2 и 3, 2 семестр):

1. Основные этапы и лексемы создания программы.
2. Назначение компилятора и компоновщика.
3. Классификация языков программирования.
4. Структурное программирование
5. Переменные, константы, строки.
6. Имена и типы, область видимости.
7. Стандартные математические функции.
8. Стандартные библиотечные функции.
9. Типы данных. Простые типы.
10. Операции над данными, приведение типов. 11.Инкрементные операции языка С.

12.Арифметические операции языка. Приоритет операций. 13.Бинарные операции.

14.Вывод результатов. Функции вывода. 15.Функции форматированного ввода данных 16.Спецификации полей данных.

17.Функции ввода строк и символов 18.Адресация данных. Операции над адресами. 19.Функции ввода данных.

20.Поразрядные (побитовые) операции. 21.Назначение и использование оператора if. 22.Вложенные конструкции оператора if 23.Оператор для организации циклов for.

24.Операторы для организации циклов while, do… while 25.Операторы передачи управления.

26.Синтаксис использования операторов break и continue. 27.Директивы условной компиляции.

28.Препроцессоры языка С. 29.Макросы: назначение и использование

1. Объявления и инициализация массивов.
2. Символьные массивы. Объявления и инициализация.
3. Указатели: назначение, объявления и инициализация. Операции над указателями.
4. Связь массива и указателя. Различие операций & и \*. 34.Типовая структура программы на языке С с функциями. 35.Функции пользователя: основные определения.
5. Рекурсивные функции.
6. Назначение и использование оператора return. 38.Передача аргументов в функцию main().
7. Динамическое распределение памяти. Функции для выделения и освобождения памяти.
8. Назначение и использование двойных указателей. Динамические массивы. 41.Структуры языка С. Инициализация структуры.
9. Массивы структур языка С.
10. Глобальные и локальные переменные. Правила инициализации. 44.Функции обработки строк strcmp(), strncmp().

45.Функции обработки строк strcpy(), strncpy(). 46.Функции обработки строк strcat(), strncat(). 47.Назначение и содержание заголовочного файла ctype.h 48.Организация работы с файлами: Открытие и закрытие.

1. Организация работы с файлами: режимы работы с файлами.
2. Файловые функции обработки символов и строк.
3. Форматированный ввод/вывод в файл/из файла. 52.Управление экраном, курсором и цветом. 53.Организация видеопамяти в текстовом режиме 54.Какой массив называется динамическим? 55.Что такое поток данных?
4. В чем состоит различие ввода данных в стандартном потоке и с внешнего устройства?
5. Напишите команды открытия файла для чтения и записи, закрытия файла. 58.В чем состоит различие команд fputs() и fgets()?

59.Какой список называется линейным? 60.Какой список называется кольцевым?

1. Как проверить: входит ли заданный элемент в список?
2. Как найти список с заданным значением информационного поля? 63.Дайте определение класса и объекта.

64.Дайте определение инкапсуляции и наследования. 65.Какие функции называются перегружаемые и почему? 66.Что общего между классом и структурой в языке С++? 67.. Структурные и функциональные схемы ПО.

1. Проектирование структур данных.
2. Проектирование ПО, основанное на декомпозиции данных. 70.Метод пошаговой детализации.

71.Структурные карты Констатайна. 72.Проектирование ПО. Методика Джексона

73.Case- технологии, основанные на методологиях анализа и проектирования. 74.Программирование с защитой от ошибок.

75.Динамические массивы. 76.Динамические структуры.

1. Списки. Линейные и связанные списки.
2. Стеки: Организация хранения в стеке и доступа к элементам. 79.Сортировка данных. Постановка задачи, виды сортировки.

80.Линейная сортировка, метод пузырька. 81.Сортировка вставкой, посредством выбора. 82.Сортировка списков путем слияния.

83.Быстрая сортировка. 84.Челночная сортировка. 85.Алгоритмы поиска.

1. Принципы и технологии программирования в ООП.
2. Методология работы в Visual Studio.
3. Методология объектно-ориентированного программирования. 89.Функции ввода/вывода в С++. Форматы вывода.

90.Понятие класса и объекта в С++. 91.Перегруженные функции.

1. Шаблонные функции.
2. Наследование в С++.
3. Конструкторы и деструкторы в С++.
4. В чем разница между списками (list) и кортежами (tuple)?
5. Как в Python работает трёхместный (тернарный) оператор?
6. Что такое отрицательный индекс (negative index)?
7. Python чувствителен к регистру?
8. Предельно допустимая длина идентификатора в Python?
9. Как можно преобразовать строку (string) в нижний регистр (lowercase)?
10. Для чего нужен pass (pass statement) в Python?
11. Расскажите про функции help() и dir() в Python.
12. Как получить список из всех ключей словаря (dictionary keys)?
13. Что такое срез в Python?
14. Как пишутся комментарии в Python?
15. Как проверить, что все символы строки относятся к алфавитно-цифровым?
16. Как перевести первый символ строки в верхний регистр?
17. Недостатки языка Python?
18. Как в Python узнать, в какой мы сейчас директории?
19. Как вставить объект, чтобы он оказался под определенным индексом?
20. Как можно обратить (reverse) порядок элементов в списке?
21. Что такое приглашение интерпретатора (interpreter prompt)?
22. Что нужно сделать, чтобы функция возвратила значение?
23. Что такое блок?
24. Если мы не поставим двоеточие в конце строки для цикла "do-while", он все равно сработает?
25. Напишите в одну строку, как можно получить самую позднюю по значению букву в строке.
26. В каких областях питон имеет преимущество?
27. Можете назвать десять встроенных функций Python?
28. Как конвертировать список в строку?
29. Как убрать из списка дубликат элемента?
30. Что такое словарь (dictionary)?

Образец билета:

Российский государственный гидрометеорологический университет Кафедра информационных технологий и систем безопасности

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

Дисциплина «Языки программирования»

1. Организация работы с файлами: режимы работы с файлами.
2. Связь массива и указателя. Различие операций & и \*.
3. Заполните массив случайными числами в диапазоне 20..100 и подсчитайте отдельно число чётных и нечётных элементов.

«Утверждаю»

Зав. кафедрой \_\_ / /

Критерии выставления оценки

Оценка «**отлично**» ставится студенту, ответ которого содержит:

* + глубокое знание программного материала, а также основного содержания и новаций лекционного курса но сравнению с учебной литературой;
	+ знание концептуально-понятийного аппарата всего курса; а также свидетельствует о способности:
	+ самостоятельно критически оценивать основные положения курса;
	+ увязывать теорию с практикой.

Оценка «отлично» не ставится в случаях систематических пропусков студентом семинарских и лекционных занятий по неуважительным причинам, а также неправильных ответов на дополнительные вопросы преподавателя.

Оценка «**хорошо**» ставится студенту, ответ которого свидетельствует о полном знании материала по программе, а также содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

Оценка «хорошо» не ставится в случаях пропусков студентом семинарских и лекционных занятий по неуважительным причинам.

Оценка «**удовлетворительно**» ставится студенту, ответ которого содержит:

* + поверхностные знания важнейших разделов программы и содержания лекционного курса;
	+ затруднения с использованием научно-понятийного аппарата и терминологии курса;
	+ стремление логически четко построить ответ, а также свидетельствует о возможности последующего обучения.

Оценка «**неудовлетворительно**» ставится студенту, имеющему существенные пробелы в знании основного материала по программе, а также допустившему принципиальные ошибки при изложении материала.

1. Учебно-методическое и информационное обеспечение дисциплины а) основная литература:
	1. Кувшинов, Д. Р. Основы программирования : учебное пособие для вузов / Д. Р. Кувшинов. — М. : Издательство Юрайт, 2018. — 104 с. — (Серия : Университеты России). — ISBN 978-5-534-07559-5. — Режим доступа : [www.biblio-online.ru/book/3D35AAB8-863B-47A8-BA39-ABF5D579204D](http://www.biblio-online.ru/book/3D35AAB8-863B-47A8-BA39-ABF5D579204D)
	2. Малявко, А. А. Формальные языки и компиляторы : учебное пособие для вузов / А. А. Малявко. — М. : Издательство Юрайт, 2018. — 429 с. — (Серия : Университеты России). — ISBN 978-5-534-04288-7. — Режим доступа : [www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3.](http://www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3)
	3. Огнева, М. В. Программирование на языке с++: практический курс : учебное пособие для бакалавриата и специалитета / М. В. Огнева, Е. В. Кудрина. — М. : Издательство Юрайт, 2018. — 335 с. — (Серия : Бакалавр и специалист). — ISBN 978-5-534-05123-0. — Режим доступа : www.biblio- online.ru/book/7670D7EC-AC37-4675-8EAE-DD671BC6D0E4
	4. Малявко, А. А. Формальные языки и компиляторы : учебное пособие для вузов / А. А. Малявко. — М. : Издательство Юрайт, 2018. — 429 с. — (Серия : Университеты России). — ISBN 978-5-534-04288-7. — Режим доступа : [www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3](http://www.biblio-online.ru/book/B4D96654-71D5-4748-986D-66E8309C25E3)
	5. Зыков, С. В. Программирование : учебник и практикум для академического бакалавриата / С. В. Зыков. — М. : Издательство Юрайт, 2018.

— 320 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-02444-

1. — Режим доступа : [www.biblio-online.ru/book/122D27F3-13E4-4095-8946-](http://www.biblio-online.ru/book/122D27F3-13E4-4095-8946-)

C619F0FCC5C3

* 1. A Byte of Python (Russian) Версия 2.02 / Swaroop C H (Перевод: Владимир Смоляр) Режим доступа: https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf

б) дополнительная литература:

* 1. Шишкин А.Д., Чернецова Е.А. Практикум "Программирование на языке Си" [Текст] : учеб. пособие / РГГМУ. - Санкт-Петербург : РГГМУ, 2003. - 51 с. - 25.00 р.
	2. Шишкин А.Д., Чернецова Е.А. Практикум по дисциплине "Языки программирования". Раздел: Программирование С и С++ [Текст] : практикум / РГГМУ. - 2-е изд., испр. и доп. - Санкт-Петербург : РГГМУ, 2012. - 68 с. - 29.43 р.
	3. Шишкин А.Д. Программирование на языке Си [Текст] : учебное пособие / РГГМУ. - Санкт-Петербург : РГГМУ, 2003. - 103 с. - 70.00 р.
	4. Назаров, С. В. Программирование в пакетах MS Office [Электронный ресурс] : учеб. пособие / С. В. Назаров, П. П. Мельников, Л. П. Смольников и др.; под ред. С. В. Назарова. - М.: Финансы и статистика, 2007. - 656 c.: ил. - ISBN 978-5-279-02926-6 - Режим доступа:<http://znanium.com/bookread2.php?book=369386>

в) программное обеспечение и Интернет-ресурсы:

*Программное обеспечение:*

* + - windows
		- office
		- Dev-C++ GNU General Public License
		- PyCharm: IDE

*Интернет-ресурсы*

* + - https://inf1.info/ - Планета информатики
		- https://www.onlinegdb.com/online\_python\_compiler
		- <http://taskcode.ru/>- Решение задач по программированию
		- [http://moodle.rshu.ru](http://moodle.rshu.ru/) - система управления курсами РГГМУ

*Информационно-справочные системы:*

* + - [https://biblio-online.ru](https://biblio-online.ru/) – ЭБС Юрайт
		- [http://znanium.com](http://znanium.com/) – ЭБС Знаниум
		- [http://www.prospektnauki.ru](http://www.prospektnauki.ru/) – ЭБС Проспект науки
		- [http://elib.rshu.ru](http://elib.rshu.ru/) ЭБС ГидроМетеоОнлайн
		- https://нэб.рф - Национальная электронная библиотека

*Профессиональные базы данных*

* + - Профессиональные базы данных не используются
1. Методические указания для обучающихся по освоению дисциплины

**(модуля)**

|  |  |
| --- | --- |
| Виды учебных занятий | Организация деятельности студента |
| Лекция | Написание конспекта лекций: кратко, схематично, последовательно фиксировать основные положения, выводы, формулировки, обобщения; помечать важные мысли, выделять ключевые слова, термины. Проверка терминов, понятий с помощью энциклопедий, словарей, справочников с выписыванием толкований в тетрадь. Обозначить вопросы, термины, материал, который вызывает трудности, пометить и попытаться найти ответ в рекомендуемой литературе. Если самостоятельно не удается разобраться в материале,необходимо сформулировать вопрос и задать преподавателю наконсультации или на лабораторном занятии. |
| Лабораторные | На лабораторных занятиях выполняются работы по программированию физических и математических задач, изученные во время лекций. Как правило, на каждом занятии студент должен показать результаты выполнения лабораторной преподавателю. После выполнения лабораторной работы студент готовится к ее защите изащищает ее. |
| Внеаудиторная работа | представляет собой вид занятий, которые каждый студент организует ипланирует самостоятельно. Самостоятельная работа студентов включает самостоятельное изучение разделов дисциплины. |
| Подготовка к экзамену/ зачёту/курсовой работе | При подготовке к экзамену, зачету и защите курсовой работы необходимо ориентироваться на конспекты лекций, рекомендуемуюлитературу и выполненные лабораторные работы. |

1. **Информационные технологии, используемые при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем (при необходимости)**

|  |  |  |
| --- | --- | --- |
| Тема (раздел) дисциплины | Образовательные и информационные технологии | Перечень программного обеспечения и информационныхсправочных систем |
| Языки программирования | Лекции. Лабораторные работы | [https://biblio-online.ru](https://biblio-online.ru/) [http://znanium.com](http://znanium.com/) [http://www.prospektnauki.ru](http://www.prospektnauki.ru/) [http://elib.rshu.ru](http://elib.rshu.ru/) https://нэб.рфhttps://www.onlinegdb.comwindows office Dev-C++ GNUPyCharm: IDE |

1. **Особенности освоения дисциплины для инвалидов и лиц с ограниченными возможностями здоровья**

Обучение обучающихся с ограниченными возможностями здоровья при необходимости осуществляется на основе адаптированной рабочей программы с использованием специальных методов обучения и дидактических материалов, составленных с учетом особенностей психофизического развития, индивидуальных возможностей и состояния здоровья таких обучающихся (обучающегося).

При определении формы проведения занятий с обучающимся-инвалидом учитываются рекомендации, содержащиеся в индивидуальной программе реабилитации инвалида, относительно рекомендованных условий и видов труда. При необходимости для обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья создаются специальные рабочие места с учетом нарушенных функций и ограничений жизнедеятельности.

1. **Материально-техническое обеспечение дисциплины**

**Учебная аудитории для проведения занятий лекционного типа** – укомплектована специализированной (учебной) мебелью, набором демонстрационного оборудования и учебно-наглядными пособиями, обеспечивающими тематические иллюстрации, соответствующие рабочим учебным программам дисциплин (модулей).

**Учебная аудитории для проведения занятий лабораторного типа** -компьютерный класс с ЛВС, связанной Интернет

**Учебная аудитория для групповых и индивидуальных консультаций** - укомплектована специализированной (учебной) мебелью, техническими средствами обучения, служащими для представления учебной информации.

**Учебная аудитория для текущего контроля и промежуточной аттестации** - укомплектована специализированной (учебной) мебелью, техническими средствами обучения, служащими для представления учебной информации.

**Помещение для самостоятельной работы** – укомплектовано специализированной (учебной) мебелью, оснащено компьютерной техникой с возможностью подключения к сети "Интернет" и обеспечено доступом в электронную информационно-образовательную среду организации

**Помещение для хранения и профилактического обслуживания учебного оборудования** – укомплектовано специализированной мебелью для хранения оборудования и техническими средствами для его обслуживания.