

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

**Б1.О.14.02 ОСНОВЫ ОБЪЕКТНО-ОРИЕНТИРОВАННОГО
ПРОГРАММИРОВАНИЯ**

Направление подготовки (специальность) 03.05.02 Фундаментальная и прикладная физика

Профиль подготовки (специализация)

Форма обучения очная

Год набора 2024

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Программу составили
доцент, А.Е.Ершов

1 Цели и задачи изучения дисциплины

1.1 Цель преподавания дисциплины:

Изучение основных принципов алгоритмизации и программирования, обучение основам языков высокого уровня (C/Python)

1.2 Задачи изучения дисциплины:

- освоение общих принципов построения алгоритмов и получение практических навыков написания программ для решения прикладных задач
- приобретение рациональных качеств мысли, чуткая объективности, интеллектуальной честности; развитие внимания, способности сосредоточиться, настойчивости, закрепление навыков работы, т.е. развитие интеллекта и формирование характера

1.3 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы высшего образования:

Код и наименование индикатора достижения компетенции	Запланированные результаты обучения по дисциплине
ОПК-6 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	
ОПК-6.1 Разрабатывает алгоритмы и компьютерные программы	знает принципы написания компьютерных программ; умеет разрабатывать алгоритмы и компьютерные программы
ОПК-6.2 Решает практические задачи с использованием компьютерных программ	владеет практическими навыками написания программ для решения прикладных задач

Дисциплина реализуется без применения ЭО и ДОТ

2 Объем дисциплины (модуля)

Вид учебной работы	Всего, зачетных единиц (акад.час)	Семестр
		2
Общая трудоемкость дисциплины	2 (72)	2 (72)
Контактная работа с преподавателем:	1,4 (52)	1,4 (52)
занятия лекционного типа	0,4 (16)	0,4 (16)
лабораторные работы	1 (36)	1 (36)
Самостоятельная работа обучающихся	0,6 (20)	0,6 (20)
Вид промежуточной аттестации (Зачет)		Зачёт, КР

3 Содержание дисциплины (модуля)

№ п/п	Вид работ	Темы занятия	Объем часов	Семестр /курс	Часы в эл. формате
Раздел 1. Введение в программирование					
1.	Лаб	Введение. Создание проекта. Компиляция первой программы	2	2	
2.	Ср	Введение. Понятие языка программирования. Виды языков программирования. Парадигмы программирования	1	2	
Раздел 2. Язык программирования C					
1.	Лек	Структура программы. Компиляция и сборка программы. Инструкции и выражения	0,5	2	
2.	Лек	Типы данных. Переменные. Объявление переменных и инициализация. Целые типы фиксированных размеров	0,5	2	
3.	Ср	Типы данных. Переменные. Объявление переменных и инициализация. Целые типы фиксированных размеров	0,5	2	
4.	Лек	Операторы. Преобразования типов. Условия. Циклы	1	2	
5.	Ср	Операторы. Преобразования типов. Условия. Циклы	0,5	2	
6.	Лек	Указатели. Массивы. Выделение памяти. Динамические массивы	0,5	2	
7.	Лек	Структуры. Объединения. Нумераторы	0,5	2	
8.	Ср	Структуры. Объединения. Нумераторы	0,5	2	
9.	Лек	Функции. Области видимости. Квалификаторы и спецификаторы типов	1	2	
10.	Ср	Функции. Области видимости. Квалификаторы и спецификаторы типов	0,5	2	
11.	Лек	Ввод-вывод	1	2	
12.	Ср	Ввод-вывод	0,5	2	
13.	Ср	Математическая библиотека. Библиотека алгоритмов	1	2	
14.	Лек	Работа со строками	0,5	2	
15.	Лек	Параллельные вычисления	0,5	2	
16.	Ср	Параллельные вычисления	0,5	2	
17.	Ср	Обзор других функций стандартной библиотеки	1	2	
18.	Лаб	Решение квадратных уравнений	4	2	
19.	Ср	Решение квадратных уравнений	1	2	
20.	Лаб	Работа с массивами	8	2	
21.	Ср	Работа с массивами	2	2	
Раздел 3. Основы программирования на языке Python					
1.	Лек	Характеристика языка Python. Интерпретатор Python. Создание и запуск программы на языке Python. Структура программы на языке Python. Понятие объекта. Инструкции. Идентификаторы. Литералы.	0,5	2	
2.	Лек	Простые встроенные типы данных. Операции над простыми типами данных. Инструкция if. Цикл while.	0,5	2	
3.	Ср	Простые встроенные типы данных. Операции над простыми типами данных. Инструкция if. Цикл while.	0,5	2	
4.	Лек	Контейнеры. Списки, кортежи, множества, словари, строки и работа с ними.	0,5	2	

5.	Ср	Контейнеры. Списки, кортежи, множества, словари, строки и работа с ними.	0,5	2	
6.	Лек	Цикл for. Понятие итерируемого объекта. Контейнеры как итерируемые объекты. Включения.	0,5	2	
7.	Ср	Цикл for. Понятие итерируемого объекта. Контейнеры как итерируемые объекты. Включения.	0,5	2	
8.	Лек	Функции. Создание, вызов. Аргументы функции. Области видимости. Функции как объекты первого класса, функции высших порядков	0,5	2	
9.	Ср	Функции. Создание, вызов. Аргументы функции. Области видимости. Функции как объекты первого класса, функции высших порядков	0,5	2	
10.	Лек	Генераторы. Генераторы как итерируемые объекты.	0,5	2	
11.	Лек	Исключения. Обработка исключений. Генерация исключений	0,5	2	
12.	Лек	Понятие модуля. Импорт модуля. Создание простого модуля.	0,5	2	
13.	Лаб	Консольный калькулятор	8	2	
14.	Ср	Консольный калькулятор	2	2	

Раздел 4. Объектно-ориентированное программирование на языке Python

1.	Лек	Понятие класса и экземпляра. Создание класса. Атрибуты класса. Создание экземпляра класса. Методы.	1	2	
2.	Лек	Специальные методы класса. Конструктор, инициализатор и деструктор. Реализация операторов.	1	2	
3.	Ср	Специальные методы класса. Конструктор, инициализатор и деструктор. Реализация операторов.	0,5	2	
4.	Лек	Индексирование. Реализация итерируемого объекта	0,5	2	
5.	Ср	Индексирование. Реализация итерируемого объекта	0,5	2	
6.	Лек	Атрибутирование. Дескрипторы. Поиск специальных методов	0,5	2	
7.	Лек	Наследование. Порядок разрешения имён. Функция super.	1	2	
8.	Ср	Наследование. Порядок разрешения имён. Функция super.	0,5	2	
9.	Лаб	Класс матриц	8	2	
10.	Ср	Класс матриц	2	2	

Раздел 5. Библиотека языка Python

1.	Лек	Обзор встроенных функций. Обзор стандартные модулей языка Python	1	2	
2.	Ср	Обзор встроенных функций. Обзор стандартные модулей языка Python	0,5	2	
3.	Лек	Параллельное программирование. Многопоточность. Многопроцессность	0,5	2	
4.	Ср	Параллельное программирование. Многопоточность. Многопроцессность	0,5	2	
5.	Лек	Библиотеки numpy, scipy, matplotlib	0,5	2	
6.	Ср	Библиотеки numpy, scipy, matplotlib	0,5	2	
7.	Лаб	Графический интерфейс калькулятора	6	2	
8.	Ср	Графический интерфейс калькулятора	2	2	
9.	Зачёт	Зачет		2	
10.	КР	Курсовая работа		2	

4 Учебно-методическое обеспечение дисциплины

4.1 Печатные и электронные издания:

1. Кузин А. В., Чумакова Е. В. Основы программирования на языке Objective-C для iOS [Электронный ресурс]: Учебное пособие. - Москва: ООО "Научно-издательский центр ИНФРА-М", 2019. - 118 с. – Режим доступа: <https://znanium.com/catalog/document?id=339481> .

2. Лутц М. Изучаем Python. 3-е издание [Электронный ресурс]:. - Б. м.: б. и., . - 830 с. – Режим доступа: http://biblio.litres.ru/pages/biblio_book/?art=24500438 .

3. Маккинни У., Слинкин А.А. Python и анализ данных [Электронный ресурс]: Практическое пособие. - Москва: ДМК Пресс, 2020. - 540 с. – Режим доступа: <https://znanium.com/catalog/document?id=427328> .

4. Керниган Б. У., Ритчи Д., Бродовой В. Л. Язык программирования C: учебное пособие. - Москва: Вильямс, 2006. - 269 с..

5. МакГрат М. Python. Программирование для начинающих [Электронный ресурс]:. - Б. м.: б. и., . - 194 с. – Режим доступа: http://biblio.litres.ru/pages/biblio_book/?art=14108932 .

6. Царев Р. Ю. Программирование на языке Си [Электронный ресурс]: учебное пособие для студентов вузов. - Красноярск: СФУ, 2014. - 106 с. – Режим доступа: <http://lib3.sfu-kras.ru/ft/lib2/elib/u004/i-436501.pdf> .

7. Кузин А. В., Чумакова Е. В. Программирование на языке Си [Электронный ресурс]: Справочник: Учебное пособие. - Москва: Издательство "ФОРУМ", 2020. - 143 с. – Режим доступа: <https://znanium.com/catalog/document?id=362870> .

8. Шкаберина Г. Ш., Резова Н. Л. Программирование. Основы языка Python [Электронный ресурс]: учебное пособие. - Красноярск: СибГУ им. академика М. Ф. Решетнёва, 2018. - 92 с. – Режим доступа: <https://e.lanbook.com/book/147450> .

4.2 Лицензионное и свободно распространяемое программное обеспечение, в том числе отечественного производства (программное обеспечение, на которое университет имеет лицензию, а также свободно распространяемое программное обеспечение):

1. Python. Высокоуровневый язык программирования общего назначения.
2. Microsoft Visual Studio. Среда разработки программного обеспечения.

4.3 Интернет-ресурсы, включая профессиональные базы данных и информационные справочные системы:

1. УМО дисциплины "Основы объектно-ориентированного программирования" <https://e.sfu-kras.ru/course/view.php?id=38104>

2. Справочник по языку C <https://ru.cppreference.com/w/c>

3. Документация библиотеки numpy <https://numpy.org/doc/stable/>

4. Документация библиотеки scipy <https://docs.scipy.org/doc/scipy/>

5. Документация библиотеки matplotlib <https://matplotlib.org/stable/users/index>

6. Библиотечно-издательский комплекс СФУ <https://bik.sfu-kras.ru/elib/databases>

5 Фонд оценочных средств

Фонд оценочных средств является приложением к рабочей программе дисциплины (модуля), хранится на кафедре, обеспечивающей преподавание данной дисциплины (модуля).

6 Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю)

учебная аудитория для проведения лекционных, семинарских и практических занятий: Специализированная мебель, демонстрационное оборудование, АРМ преподавателя, подключение к сети «Интернет» и индивидуальный неограниченный доступ в ЭИОС университета

Компьютерный класс: Компьютеры, проектор

помещение для самостоятельной работы обучающихся: специализированная мебель, демонстрационное оборудование, АРМ преподавателя, АРМ обучающихся, подключение к сети «Интернет» и индивидуальный неограниченный доступ в ЭИОС университета

Министерство науки и высшего образования РФ
Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

По дисциплине (модулю)/ практике

Б1.О.14.02 Основы объектно-ориентированного программирования

Направление подготовки/специальность

03.05.02 Фундаментальная и прикладная физика

Образовательная программа

03.05.02.30 Фундаментальная и прикладная физика

Красноярск 2025

1. Перечень компетенций с указанием индикаторов их достижения, соотнесенных с результатами обучения по дисциплине (модулю), практики и оценочными средствами

Се- местр ¹	Код и содержание индикатора компе- тенции	Результаты обучения ²	Оценочные средства ³
ОПК-6: Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения			
2	ОПК-6.1: Разрабатывает алгоритмы и компьютерные программы	знает принципы написания компьютерных программ; умеет разрабатывать алгоритмы и компьютерные программы	защита лабораторных работ; курсовая работа; контрольные вопросы к зачету
2	ОПК-6.2: Решает практические задачи с использованием компьютерных программ	владеет практическими навыками написания программ для решения прикладных задач	защита лабораторных работ; курсовая работа; контрольные вопросы к зачету

¹ Семестры указываются по порядку, для каждого индикатора

² Указываются результаты обучения по дисциплине (модулю), практике, соотнесенные с индикатором достижения компетенции.

³ Указываются оценочные средства для каждого индикатора.

2. Типовые оценочные средства или иные материалы, с описанием шкал оценивания и методическими материалами, определяющими процедуру проведения и оценивания достижения результатов обучения

Перечень лабораторных работ:

1. Введение. Создание проекта. Компиляция первой программы
2. Решение квадратных уравнений
3. Работа с массивами
4. Консольный калькулятор
5. Класс матриц
6. Графический интерфейс калькулятора

Методические рекомендации по выполнению и защите лабораторных работ:

Лабораторные занятия по дисциплине проходят в компьютерных классах. В рамках каждой лабораторной работы студент разрабатывает программу в соответствии с заданием. При разработке программы студент может воспользоваться подходом, предложенным в методических рекомендациях, или использовать собственный подход к решению задачи. После этапа разработки следует этап тестирования работоспособности программы и, при необходимости, доработкой программы. Заканчивается лабораторная работа защитой программы.

Критерии оценки лабораторной работы:

«Зачтено» за лабораторную работу, если:

- программа работает и выполняет поставленную задачу;
- программа не содержит ошибок;
- при защите лабораторной работе студент демонстрирует знание устройства программы, алгоритмов её работы;
- студент владеет теоретическим материалом: знает особенности языка программирования и его синтаксис, демонстрирует понимания выразительных средств языка программирования, использованных в программе.

«Не зачтено» за лабораторную работу, если при защите лабораторной работы не было выполнено хотя бы одно из вышеперечисленных условий.

Темы курсовых работ:

1. Программа работы с массивом переменной длины (язык C)
2. Класс разреженных матриц
3. Инженерный калькулятор с возможностью обрабатывать выражения с учётом скобок
4. Программа для работы с матрицами (язык C)
5. Программа-графопостроитель с графическим интерфейсом
6. Программа, решающая алгебраические уравнения 4 порядка в комплексных числах
7. Программа поиска по тексту файлов в каталоге
8. Программа телефонный справочник
9. Динамический графопостроитель

Методические рекомендации по выполнению курсовой работы:

Учебной программой предусмотрено выполнение курсовой работы, тема которой определяется преподавателем.

Перед выполнением задания необходимо актуализировать теоретические знания по данной теме. Для этого можно воспользоваться лекциями, учебной литературой или ресурсами сети Интернет.

В процессе написания программы необходимо придерживаться следующих рекомендаций: максимально использовать возможности выбранного языка программирования, выделять части основной программы в отдельные функции и/или подпрограммы, элементы программы сопровождать комментариями, использовать стандартную программную нотацию.

Для успешной защиты программы необходимо удовлетворить следующим критериям: программа должна быть работоспособной (проверена на тестовом примере), соответствовать рекомендациям по написанию программ,

студент должен легко ориентироваться в программном коде и уметь сопоставить каждый этап вычисления с соответствующим разделом теории.

Критерии оценки курсовой работы:

Оценка «отлично» выставляется студенту, если он написал компьютерную программу, самостоятельно протестировал её, показал работоспособность программного кода, вывел результат расчёта в файл и построил соответствующий график, показал свободное владение теоретическим материалом.

Оценка «хорошо» выставляется студенту, если он написал компьютерную программу, показал работоспособность программного кода, вывел результат расчёта на экран или в файл, построил соответствующий график и показал хорошее владение теоретическим материалом.

Оценка «удовлетворительно» выставляется студенту, если он написал компьютерную программу, показал работоспособность программного кода, вывел результат расчёта на экран и показал удовлетворительное владение теоретическим материалом.

Оценка «неудовлетворительно» выставляется студенту, который не смог написать компьютерную программу и/или показал неудовлетворительное владение теоретическим материалом.

Перечень контрольных вопросов к зачету:

1. Виды языков программирования. Парадигмы программирования
2. Типы данных.
3. Циклы
4. Функции.
5. Массивы и списки. Словари.
6. Ввод-вывод
7. Понятие класса и экземпляра
8. Атрибуты класса
9. Наследование

Методические рекомендации для проведения зачета:

Студент получает зачет при успешном выполнении и защите всех лабораторных работ и курсовой работы. Иначе зачет проходит с использованием контрольных вопросов к зачету.

Критерии оценки зачета:

«Зачтено» выставляется обучающемуся, если в ответе верно изложено не менее 65% материала и не допущено существенных неточностей.

«Не зачтено» выставляется обучающемуся, который не знает значительной части (более 65 %) программного материала и допускает существенные ошибки.

Разработчик



А.Е.Ершов