



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**
федеральное государственное бюджетное образовательное учреждение
высшего образования
«ИРКУТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
(ФГБОУ ВО «ИГУ»)
Институт математики и информационных технологий



Рабочая программа дисциплины (модуля)

Б1.О.26 Информатика и программирование

Направление подготовки	01.03.02 Прикладная математика и информатика
Направленность (профиль) подготовки	Системы искусственного интеллекта
Квалификация выпускника	бакалавр
Форма обучения	очная

Иркутск 2023 г.

1. ЦЕЛИ И ЗАДАЧИ ДИСЦИПЛИНЫ

Цели: изучение основных понятий информатики и программирования на одном из языков высокого уровня.

Задачи:

- 1) ознакомление с основными понятиями и методами информационных технологий;
- 2) ознакомление с основами языка программирования Python;
- 3) формирование устойчивого алгоритмического мышления у студентов.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Учебная дисциплина Б1.О.26 Информатика и программирование относится к обязательной части Блока 1 образовательной программы.

Перечень последующих учебных дисциплин, для которых необходимы знания, умения и навыки, формируемые данной учебной дисциплиной: Языки и системы программирования, Вычислительные системы и компьютерные сети, Производственная практика.

3. ТРЕБОВАНИЯ К РЕЗУЛЬТАТАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Процесс освоения дисциплины направлен на формирование следующих компетенций в соответствии с ФГОС ВО и ОП ВО по направлению подготовки 01.03.02 Прикладная математика и информатика:

ОПК-2 Способен использовать и адаптировать существующие математические методы и системы программирования для разработки и реализации алгоритмов решения прикладных задач;

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

ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.

В результате освоения дисциплины обучающийся должен
знать: основные понятия информатики; методы разработки алгоритмов решения математических и прикладных задач; синтаксис и семантику языка программирования Python;

уметь: разрабатывать алгоритмы решения математических задач и реализовать их на языке Python;

владеть: навыками работы на компьютере, оперирования с файлами, использования прикладного и системного программного обеспечения; офисными программами Word и Excel; навыками программирования на языке Python.

4. СОДЕРЖАНИЕ И СТРУКТУРА ДИСЦИПЛИНЫ

Объем дисциплины составляет 8 зачетных ед., 288 час.

Форма промежуточной аттестации: экзамен, экзамен.

4.1. Содержание дисциплины, структурированное по темам, с указанием видов учебных занятий и отведенного на них количества академических часов

Раздел дисциплины / тема	Сем.	Виды учебной работы				Самост. работа	Формы текущего контроля; Формы промежут. аттестации
		Контактная работа преподавателя с обучающимися					
		Лекции	Лаб. занятия	Практ. занятия			
Информатика							
Программирование							
Итого (1 семестр):		34	68		33	экз.	
Итого (2 семестр):		18	36		9	экз.	

4.2. План внеаудиторной самостоятельной работы обучающихся по дисциплине

Раздел дисциплины / тема	Самостоятельная работа обучающихся			Оценочное средство	Учебно-методическое обеспечение самост. работы
	Вид самост. работы	Сроки выполнения	Затраты времени		
Информатика					
Программирование					
Общая трудоемкость самостоятельной работы (час.)			42		
Из них с использованием электронного обучения и дистанционных образовательных технологий (час.)					

4.3. Содержание учебного материала

1. Информатика

1.1. Основные понятия информатики

Автоматизация обработки информации. Программы. Данные. Файл. Память. Алгоритм. Команда. Программирование. Язык программирования. Парадигмы программирования.

1.2. Файловая система

Понятие файла. Операционные системы и файловые системы. Файловые менеджеры. Свойства файла. Папки. Жёсткие и символические ссылки. Связи. Элементы файловых систем. Носители. Специальные устройства ввода-вывода. Шаблоны имён файлов. Типы файлов. Mime-стандарт. Операции над файлами.

1.3. Системное программное обеспечение

Операционная система. Назначение. Интерфейс. Команды ОС. Переменные окружения. Перенаправление ввода, вывода. Конвейеры. Пакетная обработка.

1.4. Файловые менеджеры

Проводник. Файловый менеджер Far. Навигация по файловой системе. Операции с файлами в Far. Архивация и работа с архивами. Настройки Far на определённые типы файлов. Клавиатурные макросы в Far.

1.5. Текстовый редактор Word

Средства форматирования текстов в Word. Понятие стиля. Использование и разработка собственных стилей. Шаблоны. Макросы.

1.6. Табличный процессор Excel

Типы данных в Excel. Концепция электронной таблицы. Функции рабочего листа. Разработка таблиц для решения прикладных задач. Макросы.

2. Программирование

2.1. Основные характеристики и особенности языка программирования Python

Интерпретируемые и компилируемые языки программирования. Реализации Python. Интегрированная среда разработки. Дистрибутивы.

2.2. Основы языка Python

Концепция интерпретируемого языка. Среда разработки Spider. Синтаксис языка Python. Концепция типов данных. Вывод и ввод данных. Средства отладки.

2.3. Типы данных языка Python

Простые типы данных. Числа: целые, вещественные и комплексные. Последовательности. Списки, кортежи. Строки.

2.4. Основные операторы языка Python

Проверка условий. Циклы for и while. Операторы continue и break.

2.5. Функции. Рекурсия

Определений функций в Python. Концепция чёрного ящика. Глобальные и локальные переменные. Побочные эффекты функции. Аргументы и параметры. Значения аргументов по умолчанию. Задание имён аргументам. Рекурсия. Взаимозаменяемость рекурсии и итерации. Определение функции как оператор.

2.6. Работа со списками

Функции над списками и методы списков. Двумерные списки. Задачи нахождения минимального элемента, сортировки и поиска.

2.7. Множества и словари

Тип данных dict. Запись словарей. Списки ключей и значений. Множество.

Преобразование множества в список. Задачи, требующие использования словарей и множеств.

2.8. Файлы. Обработка исключительных ситуаций

Файл как тип данных. Ввод и вывод из файла. Обработка и создание исключительных ситуаций. Оператор assert.

2.9. Модули.

Функции и модули. Подключение модулей. Модули os, sys, math, time.

2.10. Использование графики

Модуль turtle. Модуль tkinter. Организация пользовательского интерфейса. Построение графиков функций.

2.11. Объектно-ориентированное программирование на Python

Концепция объектов. Классы объектов. Объектная декомпозиция. Определение классов в Python. Конструктор и деструктор. Поля и методы объектов. Поля классов. Наследование. Переопределение методов. Множественное наследование. Полиморфизм. Инкапсуляция. Свойства property. Статические методы.

2.12. Коллекции и итераторы

Понятие коллекции. Стандартные коллекции. Создание коллекций. Реализация работы итератора над коллекцией. Итераторы без коллекции.

2.13. Функциональное программирование на Python

Функциональная форма записи. Композиция как основной способ построения функций. Функции высшего порядка. Лямбда-выражения. Карринг. Генераторы.

2.14. Технология разработки и отладки программ

Разработка программ сверху вниз и снизу вверх. Работа с отладчиком и профайлером. Работа в команде.

4.3.1. Перечень семинарских, практических занятий и лабораторных работ

Тема занятия	Всего часов	Оценочные средства	Формируемые компетенции
Информатика			
Программирование			

4.3.2. Перечень тем (вопросов), выносимых на самостоятельное изучение студентами в рамках самостоятельной работы

Тема	Задание	Формируемые компетенции
Информатика		
Программирование		

4.4. Методические указания по организации самостоятельной работы студентов

Самостоятельная работа студентов всех форм и видов обучения является одним из обязательных видов образовательной деятельности, обеспечивающей реализацию требований Федеральных государственных стандартов высшего образования. Согласно требованиям нормативных документов самостоятельная работа студентов является обязательным компонентом образовательного процесса, так как она обеспечивает закрепление получаемых на лекционных занятиях знаний путем приобретения навыков осмысления и расширения их содержания, навыков решения актуальных проблем формирования общекультурных и профессиональных компетенций, научно-исследовательской деятельности, подготовки к семинарам, лабораторным работам, сдаче зачетов и экзаменов. Самостоятельная работа студентов представляет собой совокупность аудиторных и внеаудиторных занятий и работ. Самостоятельная работа в рамках образовательного процесса в вузе решает следующие задачи:

- закрепление и расширение знаний, умений, полученных студентами во время аудиторных и внеаудиторных занятий, превращение их в стереотипы умственной и физической деятельности;
- приобретение дополнительных знаний и навыков по дисциплинам учебного плана;
- формирование и развитие знаний и навыков, связанных с научно-исследовательской деятельностью;
- развитие ориентации и установки на качественное освоение образовательной программы;
- развитие навыков самоорганизации;
- формирование самостоятельности мышления, способности к саморазвитию, самосовершенствованию и самореализации;
- выработка навыков эффективной самостоятельной профессиональной теоретической, практической и учебно-исследовательской деятельности.

Подготовка к лекции. Качество освоения содержания конкретной дисциплины прямо зависит от того, насколько студент сам, без внешнего принуждения формирует у себя установку на получение на лекциях новых знаний, дополняющих уже имеющиеся по данной дисциплине. Время на подготовку студентов к двухчасовой лекции по нормативам составляет не менее 0,2 часа.

Подготовка к практическому занятию. Подготовка к практическому занятию включает следующие элементы самостоятельной деятельности: четкое представление

цели и задачи его проведения; выделение навыков умственной, аналитической, научной деятельности, которые станут результатом предстоящей работы. Выработка навыков осуществляется с помощью получения новой информации об изучаемых процессах и с помощью знания о том, в какой степени в данное время студент владеет методами исследовательской деятельности, которыми он станет пользоваться на практическом занятии. Подготовка к практическому занятию нередко требует подбора материала, данных и специальных источников, с которыми предстоит учебная работа. Студенты должны дома подготовить к занятию 3–4 примера формулировки темы исследования, представленного в монографиях, научных статьях, отчетах. Затем они самостоятельно осуществляют поиск соответствующих источников, определяют актуальность конкретного исследования процессов и явлений, выделяют основные способы доказательства авторами научных работ ценности того, чем они занимаются. В ходе самого практического занятия студенты сначала представляют найденные ими варианты формулировки актуальности исследования, обсуждают их и обосновывают свое мнение о наилучшем варианте. Время на подготовку к практическому занятию по нормативам составляет не менее 0,2 часа.

Подготовка к семинарскому занятию. Самостоятельная подготовка к семинару направлена: на развитие способности к чтению научной и иной литературы; на поиск дополнительной информации, позволяющей глубже разобраться в некоторых вопросах; на выделение при работе с разными источниками необходимой информации, которая требуется для полного ответа на вопросы плана семинарского занятия; на выработку умения правильно выписывать высказывания авторов из имеющихся источников информации, оформлять их по библиографическим нормам; на развитие умения осуществлять анализ выбранных источников информации; на подготовку собственного выступления по обсуждаемым вопросам; на формирование навыка оперативного реагирования на разные мнения, которые могут возникать при обсуждении тех или иных научных проблем. Время на подготовку к семинару по нормативам составляет не менее 0,2 часа.

Подготовка к коллоквиуму. Коллоквиум представляет собой коллективное обсуждение раздела дисциплины на основе самостоятельного изучения этого раздела студентами. Подготовка к данному виду учебных занятий осуществляется в следующем порядке. Преподаватель дает список вопросов, ответы на которые следует получить при изучении определенного перечня научных источников. Студентам во внеаудиторное время необходимо прочитать специальную литературу, выписать из нее ответы на вопросы, которые будут обсуждаться на коллоквиуме, мысленно сформулировать свое мнение по каждому из вопросов, которое они выскажут на занятии. Время на подготовку к коллоквиуму по нормативам составляет не менее 0,2 часа.

Подготовка к контрольной работе. Контрольная работа назначается после изучения определенного раздела (разделов) дисциплины и представляет собой совокупность развернутых письменных ответов студентов на вопросы, которые они заранее получают от преподавателя. Самостоятельная подготовка к контрольной работе включает в себя: — изучение конспектов лекций, раскрывающих материал, знание которого проверяется контрольной работой; повторение учебного материала, полученного при подготовке к семинарским, практическим занятиям и во время их проведения; изучение дополнительной литературы, в которой конкретизируется содержание проверяемых знаний; составление в мысленной форме ответов на поставленные в контрольной работе вопросы; формирование психологической установки на успешное выполнение всех заданий. Время на подготовку к контрольной работе по нормативам составляет 2 часа.

Подготовка к зачету. Самостоятельная подготовка к зачету должна осуществляться в течение всего семестра. Подготовка включает следующие действия: перечитать все лекции, а также материалы, которые готовились к семинарским и

практическим занятиям в течение семестра, соотнести эту информацию с вопросами, которые даны к зачету, если информации недостаточно, ответы находят в предложенной преподавателем литературе. Рекомендуется делать краткие записи. Время на подготовку к зачету по нормативам составляет не менее 4 часов.

Подготовка к экзамену. Самостоятельная подготовка к экзамену схожа с подготовкой к зачету, особенно если он дифференцированный. Но объем учебного материала, который нужно восстановить в памяти к экзамену, вновь осмыслить и понять, значительно больше, поэтому требуется больше времени и умственных усилий. Важно сформировать целостное представление о содержании ответа на каждый вопрос, что предполагает знание разных научных трактовок сущности того или иного явления, процесса, умение раскрывать факторы, определяющие их противоречивость, знание имен ученых, изучавших обсуждаемую проблему. Необходимо также привести информацию о материалах эмпирических исследований, что указывает на всестороннюю подготовку студента к экзамену. Время на подготовку к экзамену по нормативам составляет 36 часов для бакалавров.

В ФБГОУ ВО «ИГУ» организация самостоятельной работы студентов регламентируется Положением о самостоятельной работе студентов, принятым Ученым советом ИГУ 22 июня 2012 г.

5. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

а) основная литература:

1. Буйначев, С.К. Основы программирования на языке Python : учеб. пособие / Н.Ю. Боклаг, С.К. Буйначев. — Екатеринбург : Издательство Уральского университета, 2014. — 94 с. — ISBN 978-5-7996-1198-9— Режим доступа: <https://rucont.ru/efd/292806> — Неогранич. доступ. 2. Информатика [Текст] : учеб. пособие для студ. вузов, обуч. по спец. гр. "Экономика и управление" и напр. "Информатика и вычислит. техника" / О. П. Новожилов; Московский гос. индустриальный ун-т. - 2-е изд., испр. и доп. - М. : Юрайт, 2012. - 564 с. : ил. ; 21 см. - (Бакалавр). - Библиогр.: с. 562-564. - ISBN 978-5-9916-1756-7 : 550.60 р. (50 экз.)

б) дополнительная литература:

2. Основы программирования [Текст] : [Учеб. пособие] / В. В. Борисенко ; Моск. гос. ун-т им. М.В. Ломоносова. - М. : Интернет-Ун-т информ. технологий, 2005. - 314 с. : ил. ; 22 см. - (Основы информатики и математики). - Библиогр.: с. 309-310. - Предм. указ.: с. 311-314. - ISBN 5-9556-0039-6 : 240.00 р., 246.82 р., 175.00 р. (13 экз.)

в) базы данных, информационно-справочные и поисковые системы:

- 1.
- 2.
- 3.

6. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ

6.1. Учебно-лабораторное оборудование

ЭТОТ РАЗДЕЛ НЕ ЗАПОЛНЯТЬ

6.2. Программное обеспечение

ПЕРЕЧИСЛИТЬ ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ТРЕБУЕМОЕ ДЛЯ ДИСЦИПЛИНЫ

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ И ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ

7.1. Оценочные средства текущего контроля

Вид контроля	Контролируемые темы	Контролируемые компетенции

Примеры оценочных средств текущего контроля

Файловая система:

1. Восстановление расширений файлов.
- Системное программное обеспечение:
2. Создание жёстких ссылок и связей с папками и файлами.
 3. Сравнение содержимого папок.

Файловые менеджеры:

4. Настройка файлового менеджера Far на типы файлов.
5. Клавиатурные макросы в Far Manager.

Текстовый редактор Microsoft Word:

6. Разработка собственных стилей.

Табличный процессор Microsoft Excel:

7. Решение задач с использованием инструментария таблиц.

Язык программирования Python:

8. Простые типы данных языка Python.
9. Проверка условий.
10. Цикл.
11. Кортежи, списки.
12. Функции.
13. Обработка списков.
14. Двумерные списки.
15. Словари.
16. Множества.
17. Ввод-вывод из файлов.
18. Модули.
19. Графика. Построение графиков функций.
20. Объектно-ориентированное программирование. Создание классов.
21. Объектно-ориентированное программирование. Наследование.
22. Объектно-ориентированное программирование. Инкапсуляция.
23. Контейнеры и итераторы.
24. Рекурсия.
25. Функциональное программирование. Функции высших порядков.

- 26. Лямбда-функции. Частичное применение.
Технология разработки и отладки программ:
- 27. Отладка программ.

Например:

- Демонстрационный вариант контрольной работы №1 (№2, №3)
- Демонстрационный вариант теста №1 (№2, №3)
- Вопросы для собеседования №1 (№2, №3)
- Вопросы для коллоквиума №1 (№2, №3)
- Темы рефератов и др.

7.2. Оценочные средства для промежуточной аттестации

Список вопросов для промежуточной аттестации:

1. Основные понятия информатики
2. Файловая система
3. Системное программное обеспечение
4. Файловые менеджеры
5. Текстовый редактор Word
6. Табличный процессор Excel
7. Основные характеристики и особенности языка программирования Python
8. Основы языка Python
9. Типы данных языка Python
10. Основные операторы языка Python
11. Функции. Рекурсия
12. Работа со списками
13. Множества и словари
14. Файлы. Обработка исключительных ситуаций
15. Модули
16. Использование графики
17. Объектно-ориентированное программирование на Python
18. Коллекции и итераторы
19. Функциональное программирование на Python
20. Технология разработки и отладки программ

Примеры оценочных средств для промежуточной аттестации:

- 1.
- 2.
- 3.

Разработчик: Черкашин Е. А., кандидат физ.-мат. наук, доцент