

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ
КЫРГЫЗСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. И. АРАБАЕВА
Отделение среднего профессионального образования
Института новых информационных технологий

«УТВЕРЖДАЮ»
Директор ИНИТ
КГУ им. И. Арабаева
К.Т.Н. и.о.доц. У.Т. Керимов
« 25 » 10 2023г.

Силлабус

по дисциплине: «Основы алгоритмизации и программирования»
по специальностям: 230109 «Программное обеспечение вычислительной техники и
автоматизированных систем»

Курс: 1 на базе 11 класса / 2 на базе 9 класса

Семестр: 1, 2 на базе 11 класса / 3,4 на базе 9 класса

Всего часов по учебному плану: 98 ч.

Количество кредитов: 2

из них: аудиторных - 72 ч., 1 полугодие: лекция - 22 ч., практических - 14 ч.

2 полугодие: лекция - 22 ч., практических - 14 ч., СРС - 24 ч.

з/о: лекция - 16 ч., практических - 10 ч., СРС - 24 ч.

Итоговый контроль: 1 полугодие - аттестация / 2 полугодие - экзамен

Рабочая программа разработана на основании Государственного образовательного стандарта среднего профессионального образования, утвержденного Министерством образования и науки КР Приказ № 863/1 от 10 мая 2022 года Рег. № 8 Код 200724.

Рассмотрено педагогическим советом

ОСПО ИНИТ КГУ им. И. Арабаева.

Протокол № 2 от « 24 » 10 2023 г.

Председатель: Н.С. Сейткаева

Одобрена учебно - методическим советом

ИНИТ КГУ-им. И. Арабаева

Протокол № 2 от « 06 » 09 2023 г.

председатель УМС:

Разработчик рабочей программы:

Преподаватель Тыналиева Чынара Таласбековна

Бишкек-2023

Сведения о преподавателе:

Тыналиева Чынара Таласбековна, преподаватель отделения среднего профессионального образования Института новых информационных технологий КГУ им. И. Арабаева.

Контактный телефон: +996556888350 (сот.), cunaratynalieva1976@gmail.com

Сведения о дисциплине:

«Основы алгоритмизации и программирования». Количество кредитов -2.

3. Место проведения: учебный корпус № 1, компьютерные классы.

Введение

Настоящий учебно-методический комплекс дисциплины устанавливает минимальные требования к знаниям и умениям студента и определяет содержание и виды учебных занятий и отчетности.

Учебно-методический комплекс предназначен для преподавателей и студентов, изучающих дисциплину «Основы алгоритмизации и программирования».

Цели и задачи дисциплины, ее значение в учебном процессе

Цели освоения дисциплины

Целью дисциплины «Основы алгоритмизации и программирования» является: изучение и освоение базовых понятий и приемов программирования, применяемых на всех основных этапах разработки программ; изучение методов программирования для овладения знаниями в области технологии программирования; подготовка к осознанному использованию как языков программирования, так и методов программирования. Воспитательной целью дисциплины является формирование у студентов научного, творческого подхода к освоению технологий, методов и средств производства программного обеспечения.

Задачи:

Обучающие:

- Введение в основные понятия: Знакомство с различными понятиями алгоритмизации и программирования, такими как алгоритмы, структура данных, типы данных, управляющие конструкции и т.д.
- Изучение языков программирования: Предоставление студентам базовых знаний о популярных языках программирования и их применение в различных областях.
- Разработка алгоритмического мышления: Помощь студентам в освоении навыков абстрактного мышления, способным разработать эффективные алгоритмы для различных задач.
- Изучение методов решения проблем: Подготовка студентов к разработке методов и стратегий решения эффективных проблем с использованием вычислительных методов.
- Практическое программирование: Проведение практических занятий и лабораторных работ, в ходе которых студенты изучают, как применять изученные концепции на примере создания программного обеспечения.
- Обучение основам отладки и тестирования: подготовка студентов к умению выявлять и устранять ошибки в коде, а также тестировать программу для обеспечения их работы.
- Развитие навыков работы в сети: Содействие развитию навыков совместной работы над программными проектами, что важно для работы в среде разработчиков.

Развивающие:

- Подготовить сознание студентов к системно-информационному восприятию мира, развивать стремление к самообразованию, обеспечить в дальнейшем социальную адаптацию в информационном обществе и успешную профессиональную и личную самореализацию;

- Раскрыть креативные способности, подготовить к художественно-эстетическому восприятию окружающего мира;
- Развивать композиционное мышление, художественный вкус, графическое умение;

- Развивать творческое воображение;
- Развивать эмоциональную сферу, чувства, душу.
- Развивать моторику руки, зрительную память, глазомер.

Воспитательные:

- Формировать информационную и эстетическую культуру обучающихся;
- Воспитывать толерантное отношение в группе.
- Добиться максимальной самостоятельности детского творчества;
- Воспитывать собранность, аккуратность при подготовке к занятию;
- Воспитывать умение планировать свою работу;
- Воспитывать умственные и волевые усилия, концентрацию внимания, логичность и развитого воображения;

Правила поведения в аудитории:

Преподаватель ожидает, что: студенты не опаздывают на занятия, не пропускают занятия без уважительной причины, отрабатывают пропущенные занятия по согласованию с преподавателем. Во время занятий нельзя разговаривать, пользоваться сотовыми телефонами, покидать аудиторию, слушать музыку, жевать резинку, кушать, читать газеты и журналы...

Политика академического поведения и этики:

Быть толерантным, уважать мнение окружающих. Возражения формулировать в корректной форме. Плагиат и другие формы нечестной работы недопустимы. Недопустимы подкашивание и списывание во время тестов, экзаменов, практических занятий, сдача экзамена за другого студента, неразрешенное копирование материалов. В случае нарушения одного из вышеперечисленных пунктов студент не аттестовывается по курсу.

Требования к студенту при изучении курса:

Прочитать необходимую литературу, выполнить весь комплекс практических работ, посетить все занятия, своевременно сдать все модули.

Виды занятий и работ студента:

- Самостоятельная работа студентов
- Выполнение практических заданий
- Написание рефератов
- Бланочное или компьютерное тестирование

Итоговый контроль: аттестация или экзамен

Успешность изучения дисциплины в системе кредитных технологий оценивается суммой набранных баллов (из 100 возможных):

Распределение баллов распределяется следующим образом:

Отлично (85%-100%)

Хорошо (70%-84%)

Удовлетворительно (55%-69%)

Неудовлетворительно (0%-54%)

Межпредметные связи. Перечень дисциплин и их разделов, усвоение которых необходимо при изучении данной дисциплины.

Пререквизиты: Основы информационной и вычислительной техники, Информатика.

Постреквизиты: Программирование, Объектно-ориентированное программирование (ООП), а также для прохождения производственной и квалификационной практики.

Компетенции по Госстандарту.

Выпускник в соответствии с целями основной профессиональной образовательной программы и задачами профессиональной деятельности, указанными в пунктах 11 и 15 настоящего Государственного образовательного стандарта, должен обладать следующими компетенциями:

а) общими (ОК):

ОК-1. Уметь организовывать собственную деятельность, выбирать методы и способы выполнения профессиональных задач, оценивать их эффективность и качество.

ОК-2. Решать проблемы, принимать решения в стандартных и нестандартных ситуациях, проявлять инициативу и ответственность.

ОК-3. Осуществлять поиск, интерпретацию и использование информации, необходимой для эффективного выполнения профессиональных задач, профессионального и личностного развития.

ОК-4. Использовать информационно – коммуникационные технологии в профессиональной деятельности.

ОК-5. Уметь работать в команде, эффективно общаться с коллегами, руководством, клиентами.

ОК-6. Брать ответственность за работу членов команды (подчиненных), за результат выполнения заданий.

ОК-7. Управлять собственным личностным и профессиональным развитием, адаптироваться к изменениям условий труда и технологий в профессиональной деятельности.

ОК-8. Быть готовым к организационно – управленческой работе с малыми коллективами. ОК-9. Способен приобретать новые знания, с большой степенью самостоятельности, с использованием современных образовательных и информационных технологий.

ОК-10. Способен на научной основе оценить свой труд, оценивать с большой степенью самостоятельности, результаты своей деятельности.

б) профессиональными, соответствующими основным видам профессиональной деятельности (ПК):

производственно-технологическая деятельность:

ПК-1. Владеет знаниями об архитектуре и технических характеристиках персональных компьютеров;

организационно-управленческая деятельность:

ПК-11. Владеет знаниями о правилах и нормах охраны труда, техники безопасности, промышленной санитарии и противопожарной защиты.

ПК-13. Способен реализовывать функции сопровождения программных продуктов;

ПК-16. Способен обеспечивать эффективное применение пакетов прикладных программ;

Структура дисциплины

№	Наименование раздела, тем лекционных занятий	Кол-во часов	Наименование тем практических занятий	Кол-во часов	СРС кол-во часов
МОДУЛЬ I					
1	Алгоритм. Свойства алгоритма. Формы записи алгоритмов. Назначение функциональных блоков.	2			
2	Основные этапы решения задач. Основные структуры алгоритмов. Данные и их типы.	2			
3	Языки программирования. Классификация языков программирования. Методы и принципы программирования. Виды программного обеспечения (ПО).	2			2
4	Введение в язык программирования Python. Среда Python. Первый запуск рабочей среды. Основные элементы языка Python.	2			2
5	Операции, переменные, литералы, типы данные в Python.	2			2

	Ввод и вывод данных в программах на языке Python. Операторы и выражения.				
6	Линейные алгоритмы: операции над целочисленными, вещественными, логическими данными.	2			2
7	Готовые модули Math, Random в Python.	2			2
8	Разветвляющийся алгоритм: Простой условный оператор, Сокращенный условный оператор, Составной условный оператор.	2			2
	Всего:	16			12
МОДУЛЬ II					
9	Многозначные ветвления. Алгоритмы поиска максимального и минимального элементов.	2			
10	Циклический алгоритм: Оператор цикла While. Инструкции break и continue.	2			
11	Оператор цикла For. Функция range().	2			2
12			Практическая работа №1 Введение в язык программирования Python. Изучить основные типы данных, команды ввода и вывода данных.	2	2
13			Практическая работа №2 Линейный алгоритм: Математические операции в Python. Библиотека (модуль) math	2	2
14			Практическая работа №3	2	2

			Разветвляющиеся алгоритм: Структура ветвление в Python.		
15			Практическая работа №4 Работа с множественным ветвлением.	2	2
16			Практическая работа №5 Циклический алгоритм: Работа с циклами в Python: Цикл while в Python.	2	2
17			Практическая работа №6 Работа циклами for в Python.	2	
18			Практическая работа №7 Работа с вложенными циклами.	2	
	Всего:	6		14	12
	ИТОГО за 1 полугодие	22		14	24
2 полугодие					
№	Наименование раздела, тем лекционных занятий	Кол-во часов	Наименование тем практических занятий	Кол-во часов	СРС кол-во часов
МОДУЛЬ I					
1	Работа со строками: основные понятия, операции с ними. Методы строк. Форматирование строк. Базовые алгоритмы обработки строк.	2			
2	Множества, работа с множествами в Python.	2			
3	Списки, работа со списками в Python.	2			
4	Кортежи, работа с кортежами в Python.	2			2
5	Словари, работа со словарями в Python.	2			2

6	Обработка вложенных последовательностей (ВП). Базовые алгоритмы ВП.	2			2
7	Введение в функции Python. Основные принципы функций в Python. Синтаксис создания функций. Аргументы и возвращаемые значения.	2			2
8	Область видимости и рекурсия. Область видимости переменных в Python. Локальные и глобальные переменные. Рекурсивные функции и их применение.	2			2
9	Модули в Python. Роль модулей в организации кода. Создание собственных модулей. Импорт модулей в Python.	2			2
	Всего:	18			12

МОДУЛЬ II

10	Продвинутые функциональные возможности. Анонимные функции (лямбда-выражения) Функции высшего порядка. Замыкания.	2			
11	Работа с графикой в Python.	2			
13			Практическая работа №8 Работа со строками в Python. Методы работы со строками	2	2
14			Практическая работа №9 Работа с множествами	2	2
15			Практическая работа №10 Работа со списками. Операции над списками в Python	2	2
16			Практическая работа №11	2	2

			Работа с кортежами. Классические способы обработки кортежей.		
17			Практическая работа №12 Работа со словарями	2	2
18			Практическая работа №13 Работа над формированием вложенных последовательностей	2	2
19			Практическая работа №14 Работа над создание пользовательских функций	2	
	Всего:	4		14	12
	ИТОГО за 2 полугодие	22		14	24

Структура дисциплины заочного отделения

№	Наименование раздела, тем лекционных занятий	Кол-во часов	Наименование тем практических занятий	Кол-во часов	СРС кол-во часов
1	Введение в язык программирования Python. Среда Python. Первый запуск рабочей среды. Основные элементы языка Python.	2			4
2	Операции, переменные, литералы, типы данные в Python. Ввод и вывод данных в программах на языке Python. Операторы и выражения.	2			4
3	Линейные алгоритмы: операции над целочисленными, вещественными, логическими данными.	2			4
4	Готовые модули Math, Random в Python.	2			4
5			Введение в язык программирования	2	4

			Python.Изучить основные типы данных, команды ввода и вывода данных.		
6			Линейный алгоритм: Математические операции в Python.Библиотека (модуль) math	2	4
	ИТОГО за 1 полугодие	8		4	24
2 полугодие					
1	Разветвляющийся алгоритм: Простой условный оператор, Сокращенный условный оператор, Составной условный оператор.	2			4
2	Многозначные ветвления. Алгоритмы поиска максимального и минимального элементов.	2			4
3	Циклический алгоритм: Оператор цикла While. Оператор цикла For.	2			4
4			Разветвляющиеся алгоритм: Структура ветвление в Python.Множественное ветвление	2	4
5			Циклический алгоритм: Работа с циклами в Python:Цикл while в Python.	2	4
6			Цикл for в Python.Функция range().Работа с вложенными циклами.	2	4
	ИТОГО за 2 полугодие	6		6	24

Учебно-методические материалы.

1.clck.ru/38FN4d

Лекция по дисциплине-



2. Практические электронные методички:

1. Практическая работа №1. Введение в язык программирования Python.

Изучить основные типы данных, команды ввода и вывода данных. -

https://docs.google.com/document/d/1pDYqL49PgphHRQ9a_rjzIQJFjGIPpMrQ7-pT7TaE3I4/edit?usp=sharing

2. Практическая работа № 2. Линейный алгоритм: Математические операции в Python. Библиотека (модуль) math -

<https://docs.google.com/document/d/1zx2bxvraL7KqsgbuSGUK00FF9Nj8JGWM3I5PFsbuDu0/edit?usp=sharing>

3. Практическая работа № 3. Разветвляющийся алгоритм: структура ветвления в Python. Множественное ветвление. -

https://docs.google.com/document/d/1cnAYVziI_03qM5nur6aW5jd5SI9a6etjTEx4zrq12Go/edit?usp=sharing

4. Практическая работа №4. Циклический алгоритм: Работа с циклами в Python: цикл while, цикл for.

<https://docs.google.com/document/d/1rvJm3himeeIvw1Han8s2T-bqVcrpxK2yL3FRquDVOZE/edit?usp=sharing>

<https://docs.google.com/document/d/1itG2cWyRZuS1kcfdiKPSxGa8oqEYpKOJ0Zow9xZYNPA/edit?usp=sharing>

Практическая работа №8-

https://docs.google.com/document/d/1bEeC_-gimtQSE7an3zliCuADoctxD6gHQREm_xAySI/edit?usp=sharing

Практическая работа №9-

<https://docs.google.com/document/d/1J8RmnDmq97gH-v2Hr6Vlf7orKEbQIa0tDo5YwWlEJ4c/edit?usp=sharing>

Практическая работа №10-

<https://docs.google.com/document/d/13I3XSjgW3QRm2N7Ok9w3BWASP-rfQStSkkOxbRZQJS0/edit?usp=sharing>

Практическая работа №11-

https://docs.google.com/document/d/1rSSvodYvJrSjgdwP6-lsm5a8z_HinNL_H7kH50Ft1vg/edit?usp=sharing

Практическая работа №12-

<https://docs.google.com/document/d/1cfAmcXnhjdSsCh28-IOPefkSTLXaukp-sdFv1U5s8Pk/edit?usp=sharing>

Критерии баллов — рейтинговой оценки знаний и умений студентов.

Деятельность студентов в течение семестра оценивается следующим образом: работа на лекционных и практических занятиях, самостоятельные работы студентов, активность, посещение занятий, итоговый контроль.

Работа на лекционных и практических занятиях

Чтение текстов и участие в дискуссиях являются важными составляющими работы на лекционных и практических занятиях. Приветствуются вопросы по структуре и содержанию текста, комментарии, помогающие уяснить значение основных категорий и т.п.

Пропущенные занятия необходимо отработать письменно или печатном виде. «Отработка» должна содержать основные моменты пропущенной темы занятия. Оценка за «отработки» не выставляется. Последний срок сдачи «отработок» - заключительное занятие по курсу (тем, кто не сможет присутствовать на заключительном занятии «отработку» необходимо принести заранее).

Критерии оценки: регулярное присутствие и активное участие, уместность и глубина вопросов и комментариев, способность задавать живой импульс дискуссии и вовлекать других студентов в дебаты.

Самостоятельная работа студентов

Самостоятельные работы выполняются на отдельном листочке письменно от руки или печатном виде. Указывается имя, фамилия, группа и дата сдачи работы.

Все письменные работы НЕ принимаются позже установленных сроков сдачи, за исключением документально подтвержденных случаев отсутствия вследствие болезни или форс-мажорных обстоятельств.

Реферативная работа оформляется письменное или печатное исполнение титульного листа, списка литературы, графических и табличных приложений.

Студенты, вовремя не сдавшие реферат, защищают свою работу на консультации или в дополнительно отведенное время.

Своевременное выполнение работ является предпосылкой к обоснованию возможности допуска студента к аттестации (экзамену).

Проверка уровня усвоения лекционных занятий, включая теоретических СРС, проводится письменно, устно или тестированием по рейтинго-модульной системе. Результаты практических работ, включая, практических СРС принимаются в виде графических и контрольных работ, рефератов и собеседования.

Система оценки знаний

№	Этапы проверки	Вид средства проверки	Баллы	Сроки
1	1 модуль	Тестирование	35	Согласно графику учебного процесса
2	2 модуль	Проверка заданий	35	Согласно графику учебного процесса
3	Практические СРС	Контрольные и графические работы, рефераты, собеседование	10	В течение семестра, до итогового контроля
4	Поощрительные баллы за активность		7	В конце семестра, до итогового контроля
5	Посещение занятий		3	В течение семестра
6	Итоговый контроль	Письменный или устный	10	Согласно графику учебного процесса
	Итого:		100	

Шкала оценки знаний

Процентное содержание (баллы)	Цифровой эквивалент баллов	Оценка по графической системе (по 10 балльной шкале)	Оценка по традиционной системе (4-х балльной)
94,5-100	4,0	A	«5» - отлично
90-94	3,67	A-	
85-89	3,33	B+	
80-84	3,0	B	«4» - хорошо
75-79	2,67	B-	
70-74	2,33	C+	
65-69	2,0	C	«3» - удовлетворительно
60-64	1,67	C-	
55-59	1,33	D+	
50-54	1,0	D	Неудовлетворительно
0-49	0	F	
X	X	X	Студент отстранен от дисциплины

Формы текущего и итогового контроля

Вопросы к модулям.

1. Этапы решения задач на ЭВМ.
2. Алгоритм. Свойства алгоритма. Формы записи алгоритма.
3. Данные и их типы. Методы сортировки данных.
4. Основные структуры алгоритмов.
5. Логические высказывания и операции.
6. Таблицы истинности. Свойства логических операций.
7. Язык программирования. Поколения, классификация, элементы языков программирования.
8. Система программирования. Классификация СП.
9. Структура системы программирования. Методы программирования.
10. Виды программного обеспечения. Общие принципы разработки ПО. Жизненный цикл ПО.
11. Язык программирования Python. Среда программирования Wing IDE.

12. Основные элементы языка. Типы данных в Python. Операторы и выражения.
13. Для каких целей используются комментарии в программах? Как можно закомментировать участок программного кода в Python?
14. Каково назначение операторов print и input? Приведите примеры использования таких операторов.
15. Программирование алгоритмов линейной структуры.
16. Программирование алгоритмов ветвления. Вложенные условия.
17. Программирование алгоритмов циклической структуры. True и False, break и continue. Вложенные циклы.
18. Расскажите о работе оператора цикла while. Приведите примеры.
Расскажите о работе цикла с оператором for по убывающим значениям параметра цикла.
19. Коллекции в Python, их классификация.
20. Строки. Функции и методы строк.
21. Множества. Операции над множествами.
22. Списки. Методы списков. Вложенные списки.
23. Напишите синтаксис объявления списков.
24. Кортежи. Преобразование коллекций.
25. Напишите синтаксис объявления кортежей.
26. Словари. Методы словаря.
27. Напишите синтаксис создания словаря.

Темы для самостоятельной работы студентов.

№	Темы для самостоятельной работы студентов
	1 полугодие
1	Разработка программного обеспечения сегодня
2	Алгоритм: способы представления и свойства
3	Языки программирования
4	Классификация языков программирования
5	Выбор языка программирования
6	Почему Python стоит изучать?
7	Алфавит языка Python
8	Основные типы данные в Python.
9	Методы ввода и вывода данных
10	Линейные алгоритмы. Операции над целочисленными данными
11	Запись математических функций, модуль MATH

12	Разветвляющиеся алгоритм
	2 полугодие
13	Основные алгоритмические инструкции языка Python
14	Многозначные ветвления
15	Циклический алгоритм, оператор цикла FOR
16	Циклический алгоритм, оператор цикла While
17	Вложенные условные операции и циклы
18	Кортежи в Python
19	Методы работы со строками
20	Базовые алгоритмы обработки строк
21	Словари в Python
22	Множества в Python
23	Списки в Python
24	Коллекции в Python, их классификация.

Тестовые задания.

Google форма Тыналиевой Ч.Т. :

1. Тест на строки: clck.ru/36EYNT
2. Тест на множества - clck.ru/33qoBV
3. Тест на списки - clck.ru/36FLK3
4. Тест на кортежи - clck.ru/36GPXd
5. Тест на словари - clck.ru/36GTfq
6. Тест на ВП- clck.ru/36GXa

Методические рекомендации студентам по освоению дисциплины

Студентам необходимо ознакомиться: – с содержанием рабочей программы дисциплины (далее - РПД), с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, методическими разработками по данной дисциплине, имеющимся на образовательном портале avp или на сайте института, с графиком консультаций преподавателей отделения. Проработка учебного материала осуществляется в порядке самостоятельной работы с использованием соответствующей учебной литературы.

Выполнение практических работ начинается на аудиторных занятиях и завершается в процессе самостоятельной работы. Поэтому рекомендуется выполнять работу в максимальном объеме на аудиторных занятиях, чтобы уменьшить время для их завершения в порядке самостоятельной работы. Для обеспечения ритмичности

учебного процесса в курсе предусмотрен централизованный регулярный рубежный контроль успеваемости и посещаемости занятий.

Цель контроля – оценить равномерность выполнения учебной нагрузки и побудить студентов работать систематично, исключая малые загрузки или перегрузки в течение семестра. Аттестация по итогам освоения дисциплины проводится в форме аттестации или экзамена. Итоговая оценка знаний и умений студентов определяется комплексно на основе познавательной активности на занятиях, оценки качества и своевременности выполнения работ. Поэтому студентам рекомендуется уделять особое внимание качеству выполнения работ, а также их оформления. Для большего осознания роли и места дисциплины «Компьютерная графика и мультимедиа» в системе среднего профессионального образования, а также значения графической грамотности для будущей профессиональной деятельности студентам рекомендуется приобщение к научной работе и участие в студенческих научных форумах. Контроль самостоятельной работы студентов осуществляется в течение всего семестра. Результаты контроля самостоятельной работы студентов учитываются при осуществлении промежуточного контроля по дисциплине. Самостоятельная работа является неотъемлемой частью обучения. На этот вид работы отводится до 50% от общего объема часов. Самостоятельная работа может быть аудиторной (выполнение отдельных заданий на занятиях) и внеаудиторной. Для выполнения самостоятельной работы используются:

1. Учебники и учебные пособия.
2. Мультимедийные средства: работа в сети Интернет (использование обучающих программ и учебных сайтов, электронных образовательных ресурсов).

Самостоятельная работа студента по дисциплине включает:

- Проработку материала, изучаемого на практических занятиях;
- Подготовку к практическим работам;
- Подготовка к аттестации или экзамену.

Список литературы и учебно-методическая литература по дисциплине.

Основные источники:

1. Голицына О.Л., Попов И.И. Основы алгоритмизации и программирования: Уч.пособие.- М.: ФОРУМ: ИНФРА-М, 2004.
- 2.Златопольский Д. М. Основы программирования на языке Python,Издательство "ДМК Пресс",2018- 396с.
3. Доусон М. Програмируем на Python. - СПб.: Питер, 2016. - 416 с.
4. Прохоренок *Н.А.* Python 3 и PyQt. Разработка приложений. - СПб.: БХВ-Петербург, 2015. - 704 с.
- 5.Цыбуля И.Н. , Самыкбаева Л.Ф. Информатика. - Бишкек, 2020.-203с.
6. Семакин И.Г., Шестаков А.П. Основы программирования: Учебник.- М.: Мастерство; НМЦ СПО; Высшая школа, 2001.
- 7.Сысоева М. В., Сысоев И. В. Программирование для «нормальных» с нуля на языке Python: Учебник. В двух частях. Часть 1 / Ответственный редактор: В. Л. Черный : — М.: Базальт СПО; МАКС Пресс, 2018. — 176 с. [+4 с. вкл]: ил. — (Библиотека ALT).
- 8.Федоров Д.Ю. Программирование на языке высокого уровня Python, Учебное пособие для СПО. - Москва:Издательство Юарт , 2019 -211с.

Дополнительные источники:

1. Рапаков Г.Г., Ржеуцкая С.Ю. – Программирование на Python 3/0 для студентов и школьников. – СПб.:БХВ-Петербург, 2007.-352 с.:
2. Ускова О.Ф. – Программирование на языке Питон Задачник. Изд. Питер. 2002. - 336с.
3. Ушаков Д.М., Юркова Т.А. – Python для школьников. –СПб.: Питер, 2006г. – 256с.: 12

4. Чернов А.Ф. – Олимпиадные задачи с решениями и подробным анализом. – Волгоград: Учитель, 2007. – 207с.:

Интернет-ресурсы:

1. Studref - Студенческие реферативные статьи и материалы (info{at}studref.com) © 2017 - 2021
- 2.nportal.ru - Образовательная социальная сеть.
- 3.Основы программирования на Python. Учебное пособие для СПО.-Электронная книга...,Дата выхода на ЛитРес: 01 сентября 2021
Дата написания: 2021,Объем: 287 стр
- 4.Kopilkaurokov.ru - сайт для учителей.

Глоссарий

- 1.**Алгоритм** - последовательность шагов, описывающая решение конкретной задачи или преобразование данных.
- 2.**Программирование** - процесс создания программного кода для автоматизации задач или выполнения определенных операций.
- 3.**Язык программирования** - формальный язык, используемый для написания компьютерных программ.
- 4.**Инструкция** - одна команда или операция в программе.
- 5.**Переменная** - имя, связанное с определенным значением или данными в программе.
- 6.**Тип данных** - спецификация, определяющая, какие виды данных может содержать переменная (например, целые числа, строки, булевы значения).
- 7.**Цикл** - конструкция в программе, позволяющая выполнять определенные действия многократно.
- 8.**Условие** - логическое выражение, которое определяет, какие действия выполнить в программе в зависимости от истинности выражения.
- 9.**Функция** - фрагмент кода, который может быть вызван для выполнения определенной операции.
- 10.**Параметр функции** - значение, передаваемое в функцию в момент вызова для использования внутри функции.
- 11.**Массив (список)** - структура данных, которая позволяет хранить множество значений под одним именем.
- 12.**Рекурсия** - техника, при которой функция вызывает саму себя для решения задачи.
- 13.**Алгоритмическая сложность** - мера того, насколько эффективно работает алгоритм при обработке данных или выполнении задачи.
- 14.**Отладка** - процесс поиска и исправления ошибок в программном коде.
- 15.**Интерфейс пользователя (UI)** - визуальные и функциональные элементы, с которыми взаимодействует пользователь при работе с программой.
- 16.**Интеграция** - процесс объединения различных компонентов программы или системы для их совместной работы.
- 17.**API (Интерфейс программирования приложений)** - набор функций и процедур, которые позволяют разным программам взаимодействовать друг с другом.
- 18.**Подпрограмма** - многократно используемый фрагмент кода, обычно выполняющий определенную задачу.

- 19. Псевдокод** - подход к описанию алгоритмов, использующий смешанный набор натурального языка и кода для упрощения понимания.
- 20. Целостность данных** - принцип обеспечения согласованности и надежности данных в программе или системе.
- 21. Python** - популярный интерпретируемый язык программирования, известный своей простотой и читаемостью.
- 22. Интерпретатор Python** - программа, выполняющая код Python, строка за строкой.
- 23. Значение None** - специальное значение, представляющее отсутствие данных или ничего.
- 24. Комментарий** - текст в коде, который игнорируется интерпретатором и используется для объяснения кода или оставления заметок.
- 25. Инструкция print()** - инструкция для вывода данных на экран.
- 26. Переменная** - место для хранения данных, которое можно изменить в ходе выполнения программы.
- 27. Тип данных** - определяет, какие виды данных могут храниться в переменных (например, int, float, str).
- 28. Строка (string)** - тип данных, представляющий текст, заключенный в кавычки.
- 29. Список (list)** - структура данных, представляющая упорядоченную коллекцию элементов.
- 30. Словарь (dictionary)** - структура данных, представляющая коллекцию пар "ключ-значение".
- 31. Условное выражение (if)** - конструкция для выполнения кода в зависимости от условия.
- 32. Цикл for** - конструкция для итерации по элементам последовательности (например, списка).
- 33. Цикл while** - конструкция для выполнения кода, пока условие выполняется.
- 34. Функция** - блок кода, который может быть вызван с определенными аргументами для выполнения определенной задачи.
- 35. Модуль** - файл, содержащий Python-код и функции, который может быть импортирован в другие программы.
- 36. Исключение (exception)** - ошибка, возникающая во время выполнения программы, которую можно обработать.
- 37. Итератор** - объект, позволяющий итерировать по последовательности данных (например, for в цикле).
- 38. Индексация** - обращение к элементу списка или строки по его порядковому номеру.
- 39. Метод** - функция, связанная с объектом (например, методы строки str.upper() или список list.append()).
- 40. Модуль math** - стандартный модуль Python, предоставляющий математические функции и константы.