

Федеральное государственное образовательное учреждение
высшего образования
«КАЛИНИНГРАДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»

Е. Ю. Заболотнова

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ

Учебно-методическое пособие по изучению дисциплины
для студентов бакалавриата по направлению подготовки
09.03.03 Прикладная информатика

Калининград
Издательство ФГБОУ ВО «КГТУ»
2025

УДК 004.4(075)

Рецензент

кандидат технических наук, доцент, доцент кафедры теории механизмов и машин и деталей машин ФГБОУ ВО «Калининградский государственный технический университет» О. С. Витренко

Заболотнова, Е. Ю.

Программирование на языках высокого уровня: учеб.-метод. пособие по изучению дисциплины для студентов бакалавриата по направлению подготовки 09.03.03 Прикладная информатика / Е. Ю. Заболотнова. – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2025. – 22 с.

В учебно-методическом пособии приведен тематический план по дисциплине и даны методические указания по её самостоятельному изучению, подготовке к лабораторным занятиям, выполнению расчетно-графической работы, подготовке и сдаче экзамена, выполнению самостоятельной работы.

Пособие подготовлено в соответствии с требованиями утвержденной рабочей программы дисциплины направления подготовки 09.03.03 Прикладная информатика.

Список лит. – 8 наименований

Учебно-методическое пособие по изучению дисциплины рекомендовано к использованию в качестве локального электронного методического материала в учебном процессе методической комиссией института цифровых технологий ФГБОУ ВО «Калининградский государственный технический университет» 24 июня 2025 г., протокол № 5

УДК 004.4(075)

© Федеральное государственное
бюджетное образовательное учреждение
высшего образования
«Калининградский государственный
технический университет», 2025 г.
© Заболотнова Е. Ю., 2025 г.

ОГЛАВЛЕНИЕ

1. Введение	4
2. Тематический план	5
3. Содержание дисциплины	9
4. Методические указания по выполнению практических работ	12
5. Методические указания по выполнению расчетно-графической работы	12
6. Методические указания по выполнению самостоятельной работы	14
7. Методические указания по проведению занятий и освоению дисциплины	14
8. Требования к аттестации по дисциплине	15
Примерные вопросы экзамену по дисциплине	16
9. Заключение	19
Список литературы	20

1. ВВЕДЕНИЕ

Данное учебно-методическое пособие предназначено для студентов направления 09.03.03 Прикладная информатика, изучающих дисциплину «Программирование на языках высокого уровня».

Целью освоения дисциплины «Программирование на языках высокого уровня» является формирование у студентов знаний по методам и технологиям разработки и отладки программ на алгоритмических языках программирования высокого уровня с применением интегрированных сред разработки программ.

В результате изучения дисциплины студент должен:

знать:

- законы эволюции программного обеспечения;
- сущность объектно-ориентированного подхода к проектированию и разработке программ базовые понятия визуального и событийного программирования;

– структуру современного программного интерфейса с базами данных;

уметь:

- проводить анализ современных методов и средств программирования в процессе их выбора при решении прикладных задач различных классов;
- использовать преимущества высокоуровневых технологий при создании программных приложений;
- тестировать и отлаживать программные системы, реализованные на основе современных технологий программирования;

владеть:

- навыками разработки программных приложений на основе современных высокоуровневых технологий.

Дисциплина «Программирование на языках высокого уровня» входит в состав Естественного и инженерного модуля основной профессиональной образовательной программы бакалавриата по этому направлению подготовки.

Для успешного освоения дисциплины, в соответствии с учебным планом, ей предшествуют дисциплины «Информатика и основы программирования».

Дисциплина «Программирование на языках высокого уровня» изучается на втором курсе в третьем семестре студентами очной формы обучения и в зимнем семестре на 2-м курсе студентами заочной формы обучения.

Далее в пособии представлен тематический план, содержащий перечень изучаемых тем, выполняемых практических работ, мероприятий текущей аттестации и отводимое на них аудиторное время (занятия в соответствии с расписанием) и самостоятельную работу. При формировании личного

образовательного плана на семестр следует оценивать рекомендуемое время на изучение дисциплины, возможно, вам потребуется больше времени на выполнение отдельных заданий или проработку отдельных тем.

В разделе Содержание дисциплины приведены подробные сведения об изучаемых вопросах, по которым вы можете ориентироваться в случае пропуска каких-то занятий, а также методические рекомендации преподавателя для самостоятельной подготовки, каждая тема имеет ссылки на литературу (или иные информационные ресурсы), а также контрольные вопросы для самопроверки.

Раздел «Текущая аттестация» содержит описание обязательных мероприятий контроля самостоятельной работы и усвоения разделов или отдельных тем дисциплины. Далее изложены требования к завершающей аттестации – зачету и/или экзамену.

В разделе «Балльно-рейтинговая система» приведен порядок применения балльно-рейтинговой системы контроля успеваемости.

Помимо данного пособия, студентам следует использовать материалы, размещенные в соответствующем данной дисциплине разделе ЭИОС, в которые более оперативно вносятся изменения для адаптации дисциплины под конкретную группу.

ТЕМАТИЧЕСКИЙ ПЛАН

Общая трудоемкость дисциплины для всех форм обучения составляет пять зачетных единицы (ЗЕТ), т. е. 180 академических часа контактной (лекционных и практических занятий, а также контрактной работы посредством электронной-информационно-образовательной среды) и самостоятельной работы студента, в том числе связанной с текущей и промежуточной (заключительной) аттестацией по дисциплине.

Формы аттестации по дисциплине:

- очная форма, третий семестр – экзамен, расчетно-графическая работа;
- заочная форма, зимний семестр 2-го курса – экзамен, расчетно-графическая работа.

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч	Объем самост. работы, ч	Объем аудиторно й работы, ч	Объем самост. работы , ч
			Очная форма обучения		Заочная форма обучения	
		Лекции				
1	Реализация базовых алгоритмов на языке высокого уровня Python»	1.1. Типы и структуры данных. Разработка базовых алгоритмов	4		2	5
		1.2 Работа с файлами	4			5
		1.3 Обработка исключений в Python	4			5
2	Компьютерная графика и анимация	2.1 Компьютерная графика и анимация	8		2	15
3	Объектно-ориентированное программирование	3.1 Реализация принципов ООП в языке Python	8		2	15
4	Итераторы, генераторы и декораторы	4.1 Итераторы, генераторы и декораторы	4		2	15
5	Разработка графического интерфейса	5.1 Разработка приложений с графическим интерфейсом»	6		2	15
			48		10	75

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч	Объем самост. работы, ч	Объем аудиторно й работы, ч	Объем самост. работы , ч
			Очная форма обучения		Заочная форма обучения	

Практические занятия						
1	Реализация базовых алгоритмов на языке высокого уровня Python	<u>Практическое занятие № 1</u> Реализация линейного, разветвляющегося и циклического алгоритмов	4	2	2	4
		<u>Практическое занятие № 2</u> Операции и методы для списков и кортежей	4	2		4
		<u>Практическое занятие № 3</u> Операции и методы для словарей и множеств	4	2		4
		<u>Практическое занятие № 4</u> Работа с файлами»	8	2	2	4
		<u>Практическое занятие № 5</u> Применение исключений	6	2		4
2	Компьютерная графика и анимация	<u>Практическое задание № 6</u> Использование библиотеки graph для создания графических примитивов и анимации»	4	2	2	4
3	Объектно-ориентированное программирование	<u>Практическое задание № 7</u> Объектно-ориентированное программирование	8	2	2	4
4	Итераторы, генераторы и декораторы	<u>Практическое задание № 8</u> Применение генераторов и декораторов в программировании	4	2		4
5	Разработка	<u>Практическое занятие № 9</u>	6	2	2	4

	Раздел (модуль) дисциплины	Тема	Объем аудиторной работы, ч	Объем самост. работы, ч	Объем аудиторно й работы, ч	Объем самост. работы , ч
			Очная форма обучения		Заочная форма обучения	
	графического интерфейса программы	Разработка программы с графическим интерфейсом				
			48	18	10	36

		Расчетно-графическая работа				
1	Разработка графического интерфейса программы	Разработка программы согласно варианту задания		7		24
		Оформление пояснительной записки. Защита		2		6
				10		30

		Рубежный (текущий) и итоговый контроль				
		Итоговый контроль (экзамен)		34,75		9

3. СОДЕРЖАНИЕ ДИСЦИПЛИНЫ

Содержательно структура дисциплины представлена пятью тематическими блоками (разделами).

Тема 1. Реализация базовых алгоритмов на языке высокого уровня Python»

Перечень изучаемых вопросов

1. Реализация базовых алгоритмов на языке Python
2. Типы и структуры данных.
3. Методы для работы со списками и кортежами.
4. Методы и операции со строками.
5. Методы для работы со словарями и множествами.
6. Работа с файлами.
7. Исключения и их обработка.

Литература

[2, с. 6–123]; [3, с. 4–75].

Контрольные вопросы

1. В чем отличие полного и неполного ветвления?
2. Когда используется конструкция `elif`?
3. Как записать сложное условие в операторе?
4. Что собой представляет алгоритмическая структура повторения? Какие основные операции она предусматривает?
5. Как в программе выделить область действия цикла?
6. Каким образом структура повторения «повторять пока» записывается в программах?
7. В чем отличие операторов `for` и `while`?
8. Что такое бесконечный цикл?
9. Как можно определить список?
10. Какой метод определяет длину списка?
11. Элементами списка могут переменные разных типов?
12. Как определить наличие элементов в списке?
13. Какой метод предназначен для добавления элемента в список?
14. Как можно определить строку?
15. Какой метод возвращает длину строки?
16. Как разбить строку по пробелам?
17. С помощью какого метода можно сформировать из списка строку?
18. Как организовать доступ к элементам строки?

19. Что такое словарь?
20. В чем отличие словаря от списка?
21. Какие средства предназначены для создания словаря?
22. Какие элементы словаря должны быть уникальными?
23. Перечислите методы для работы со словарями
24. Для чего используют исключения?
25. Какие наиболее часто встречаемые исключения?
26. Какие средства предназначены для обработки исключений?
27. Почему исключения представляют в виде иерархической структуры?
28. Для чего используются ключи при открытии файлов?
29. Перечислите три метода чтения из файла?
30. Как выполнить запись в файл?
31. Какой тип данных наиболее часто используется?
32. Как закрыть файл?

Тема 2. Компьютерная графика и анимация

Перечень изучаемых вопросов

1. Построение графических примитивов средствами библиотеки graph.
2. Построение анимированного изображения.

Литература

[4, с. 51–70].

Контрольные вопросы

1. Какие средства языка предназначены для изображения простейших фигур?
2. Как задать цвет и толщину линий?
3. Для каких фигур можно выполнить заливку и как задать цвет заливки?
4. Как представлены координаты точек на холсте?
5. Что понимают под объектом при создании анимации?
6. Какие средства языка предназначены для перемещения объектов?
7. Что такое «обработчики событий»?
8. Как задать шаг изменения координат при перемещении объекта?
9. Как завершить работу программы?

Тема 3. Объектно-ориентированное программирование

Перечень изучаемых вопросов

1. Парадигмы программирования.
2. Создание классов.
3. Принципы ООП.
4. Реализация принципов ООП в программах на языке Python.

Литература

[4, с. 122–124].

Контрольные вопросы

1. Что такое атрибуты и методы класса?
2. Отличие класса от объекта?
3. В чем отличие метода от функции?
4. Какой параметр указывается при использовании в методе атрибута класса?
5. Для чего защищают поля и методы класса?
6. Как можно изменить значение закрытого поля?
7. Как получить доступ к закрытым полям и методам?
8. Что называется инкапсуляцией?
9. В чем назначение конструктора в Python?
10. Что такое базовый класс и класс – наследник?
11. Какой класс называется абстрактным?
12. Что такое полиморфизм?

Тема 4. Итераторы, генераторы и декораторы

Перечень изучаемых вопросов

1. Итераторы и декораторы.
2. Практическое применение генераторов и итераторов в программах

Литература

[5, с. 133–135].

Контрольные вопросы

1. Что такое генератор в Python и как он отличается от обычной функции?
2. Какие преимущества дает использование генераторов по сравнению с обычными функциями или списками? Приведите пример, где использование генератора значительно экономит память.
3. Что такое декоратор в Python и как он используется?
4. Как можно использовать декораторы для вызовов функций и их аргументов?
5. Как можно комбинировать несколько декораторов для одной функции?

Тема 5. Разработка графического интерфейса программы

Перечень изучаемых вопросов

1. Понятие виджета. Основные типы виджетов.
2. Понятие события. Событийное программирование
3. Метод bind.
4. Разработка программы с графическим интерфейсом.

Литература

[3, с. 5–51].

Контрольные вопросы

1. Что понимают под виджетами?
2. Что образует главный цикл обработки событий в tkinter?
3. Как организовано размещение виджетов?
4. Какие свойства общие для всех виджетов?
5. Назначение виджетов Button, Label, Entry, Text?
6. Назначение виджетов Listbox, Frame, Radiobutton, Scrollbar?

4. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

Количество практических работ, их тематика и часы аудиторной и самостоятельной работы, отводимые на выполнение каждой работы, приведены в разделе 2.

При выполнении каждой практической работы студент должен написать программу, согласно своему варианту и представить преподавателю отчет в электронном виде, содержащий следующие разделы:

1. ФИО студента и группа.
2. Тема работы и индивидуальное задание.
3. Текст программы.
4. Скриншоты выполнения программы в разных режимах.

5. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ РАСЧЕТНО-ГРАФИЧЕСКОЙ РАБОТЫ

Расчетно-графическая работа по дисциплине состоит из двух разделов: описание теоретического вопроса по варианту и представление программы для решения задачи. Задание второй части (программа) состоит в выборе по запросу данные из созданного файла, отсортировать записи и вывести результат с определением количества записей, удовлетворяющих условию. Исходные данные файла и ключ сортировки результатов определяет студент самостоятельно.

Программа должна быть оформлена графическим интерфейсом GUI(англ. graphical user interface). При разработке формы GUI рекомендуется использовать следующие виджеты:

- кнопки начала и завершения работы;
- кнопки для поиска информации и расчетов;
- метка с указанием имени и фамилии студента, а также его группы;
- однострочное тестовое поле для ввода имени файла, который будет обрабатываться в программе;
- однострочное тестовое поле для ввода условия отбора записей из файла;
- метки для вывода результатов расчета;
- список или многострочное тестовое поле для вывода данных из файла, удовлетворяющих условию.

Дизайн формы студент также определяет сам.

Следует включить в программу блок/блоки обработки исключений, например, отсутствие файла, ошибочный запрос – ввод букв вместо цифр и т. д.

Имя файла вводится в окно ввода, по кнопке «Открыть» файл открывается. В поле вводится условие поиска информации, в список или многострочное тестовое поле выводятся записи, удовлетворяющие заданному условию, либо сообщение об отсутствии данных. Кнопка «Расчет» инициирует расчеты и вывод результата. Кнопка «Закрыть» закрывает файл и форму. При написании программы обязательно использовать принципы объектно-ориентированного программирования.

Расчетно-графическая работа (пояснительная записка) должна включать следующие разделы:

Титульный лист

Содержание

Глава 1. Теоретический вопрос:

- Описание темы.
- Примеры задач по теме.
- Список используемых источников.

Глава 2. Программирование задачи

- Постановка задачи.
- Описание используемых в программе классов.
- Макет GUI с описанием использованных виджетов.
- Описание событий и методов, с ними связанных.
- Текст программы и текст исходного файла.
- Скриншоты выполнения программы.

6. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫПОЛНЕНИЮ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Самостоятельная работа студентов по дисциплине выполняется с использованием приведенного списка литературы, интернет источников по темам дисциплины, а также с материалами, представленными в ЭИОС.

Самостоятельная работа студента включает в себя освоение теоретического учебного материала, особенно для студентов очно-заочной формы обучения для успешного прохождения текущего и промежуточного контроля.

Помимо источников, приведенных в списке, используйте официальную документацию по языку программирования Python(<https://docs.python.org/3/>), описание популярных библиотек, которые являются важными ресурсами для понимания возможностей языка и его экосистемы. Это поможет студентам лучше понять функции и методы, а также научиться эффективно использовать их в своих проектах.

Практика является ключевым элементом в освоении программирования. Регулярное выполнение заданий поможет закрепить теоретические знания и развить навыки решения реальных задач.

7. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ПРОВЕДЕНИЮ ЗАНЯТИЙ И ОСВОЕНИЮ ДИСЦИПЛИНЫ

Для успешного изучения дисциплины возможно применение следующих методических приемов:

1. Интерактивное обучение и практические занятия:

Практическое применение знаний является ключевым аспектом обучения программированию. Интерактивные занятия помогают студентам лучше усваивать материал и развивать навыки решения задач. При проведении практических занятий можно использовать кодинг-сессии, когда студенты могут наблюдать за процессом написания кода и задавать вопросы в реальном времени.

2. Использование реальных проектов и кейсов:

Применение знаний на реальных проектах помогает студентам понять, как теоретические концепции работают на практике, и развивает навыки проектирования и разработки программного обеспечения. Это могут быть проекты по анализу данных, разработке веб-приложений или автоматизации

задач. Разделите проекты на этапы и предоставляйте студентам обратную связь на каждом этапе.

3. Поддержка самостоятельного обучения:

Самостоятельное обучение является важной частью развития навыков программирования. Рекомендуйте студентам онлайн-курсы, книги и другие ресурсы для самостоятельного изучения. Организуйте консультации и офисные часы, где студенты могут задавать вопросы и получать помощь. Поощряйте участие в онлайн-сообществах и форумах.

4. Обратная связь и оценка:

Регулярная обратная связь помогает студентам понимать свои успехи и области для улучшения, а также мотивирует их к дальнейшему обучению. Предоставляйте студентам регулярную обратную связь по выполненным заданиям и проектам. Используйте разнообразные методы оценки, такие как тесты, практические задания, проекты и презентации. Поощряйте студентов к самооценке и рефлексии.

8. ТРЕБОВАНИЯ К АТТЕСТАЦИИ ПО ДИСЦИПЛИНЕ

Текущая аттестация (текущий контроль) проводится с целью оценки освоения теоретического учебного материала, в том числе в рамках самостоятельной работы студента.

Контроль на лекциях и практических занятиях производится в виде тестирования или устного опроса. Типовые контрольные вопросы для устного опроса по темам приведены в п. 3 настоящего пособия. Тестовые задания представлены в фонде оценочных средств (приложение к рабочей программе дисциплины).

Положительная оценка («зачтено») по результатам каждого контроля (опроса) выставляется в соответствии с универсальной системой оценивания, приведенной в таблице ниже. В случае получения оценки «не зачтено» студент должен пройти повторный контроль по данной теме в ходе последующих консультаций.

Система оценок и критерии выставления оценки при прохождении
контроля (опроса).

Критерий	Система оценок			
	«не зачтено»	«зачтено»		
Системность и полнота знаний в отношении изучаемых объектов	Обладает частичными и разрозненными знаниями, которые не может научно-корректно связывать между собой (только некоторые из которых может связывать между собой)	Обладает минимальным набором знаний, необходимым для системного взгляда на изучаемый объект	Обладает набором знаний, достаточным для системного взгляда на изучаемый объект	Обладает полнотой знаний и системным взглядом на изучаемый объект

Завершающим этапом изучения дисциплины является промежуточная аттестация, представляющая собой защиту расчетно-графической работы и экзамен.

Условием допуска к экзамену является выполнение практических работ. Экзамен может проводится как в традиционной форме, так и в виде экзаменационного тестирования.

Примерные вопросы экзамену по дисциплине

1. Элементарные конструкции языка программирования Python. Требования к идентификаторам программных единиц (переменных, констант, функций и т. д.). Структура программы. Комментарии.

2. Типы данных в Python. Динамическая типизация. Преобразование типов данных.

3. Арифметические операции в Python. Операции над целыми и вещественными числами.

4. Оператор присваивания в Python. Команды ввода-вывода данных.

5. Функции в Python. Глобальные и локальные переменные. Создание функций.

6. Логический тип данных в Python. Операции отношения. Логические операции. Условный оператор в Python.

7. Циклы с заданным числом повторений и по условию в Python.

8. Структуры данных. Списки в Python. Операции со списками и основные методы списков.

9. Структуры данных. Списки списков в Python (матрицы). Создание и обработка.

10. Структуры данных. Строки в Python. Операции со строками и основные методы строк.

11. Структуры данных. Кортежи в Python. Операции с кортежами и методы кортежей.

12. Структуры данных. Словари в Python. Операции со словарями и методы словарей.

13. Структуры данных. Работа с файлами.

14. Модуль graph. Создание графических примитивов.

15. Модуль graph. Создание анимированных изображений.

Понятие ООП. Основные принципы ООП.

1. Описание классов в Python. Члены класса: поля класса, поля экземпляра класса, конструктор, методы. Структура метода. Параметры метода. Параметр self.

2. Реализация принципов ООП в Python.

3. Событийно –ориентированное программирование. Программы с графическим интерфейсом.

4. Модуль tkinter. Создание компонентов (виджетов): кнопка, метка (надпись), текст однострочный и многострочный. Свойства и методы этих виджетов.

5. Модуль tkinter. Создание компонентов (виджетов): рамка, флажки и радиокнопки, списки, шкала, окно верхнего уровня. Свойства и методы этих виджетов.

6. Модуль tkinter. Менеджеры геометрии.

7. Модуль tkinter. Метод bind. Типы переменных и события.

Система оценивания и критерии выставления оценок по экзамену (экзаменационному тестированию) приведена в нижеследующей таблице.

Система оценок и критерии выставления оценки по экзамену (экзаменационному тестированию).

Критерий	Система оценок			
	2	3	4	5
	0–40 %	41–60 %	61–80 %	81–100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
1 Системность и полнота знаний в отношении изучаемых объектов	Обладает частичными и разрозненными знаниями, которые не может научно-	Обладает минимальным набором знаний, необходимым для системного взгляда на	Обладает набором знаний, достаточным для системного взгляда на изучаемый	Обладает полнотой знаний и системным взглядом на изучаемый объект

Критерий	Система оценок			
	2	3	4	5
	0–40 %	41–60 %	61–80 %	81–100 %
	«неудовлетворительно»	«удовлетворительно»	«хорошо»	«отлично»
	корректно связывать между собой (только некоторые из которых может связывать между собой)	изучаемый объект	объект	
2 Работа с информацией	Не в состоянии находить необходимую информацию, либо в состоянии находить отдельные фрагменты информации в рамках поставленной задачи	Может найти необходимую информацию в рамках поставленной задачи	Может найти, интерпретировать и систематизировать необходимую информацию в рамках поставленной задачи	Может найти, систематизировать необходимую информацию, а также выявить новые, дополнительные источники информации в рамках поставленной задачи
3 Научное осмысление изучаемого явления, процесса, объекта	Не может делать научно-корректных выводов из имеющихся у него сведений, в состоянии проанализировать только некоторые из имеющихся у него сведений	В состоянии осуществлять научно-корректный анализ предоставленной информации	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные данные	В состоянии осуществлять систематический и научно-корректный анализ предоставленной информации, вовлекает в исследование новые релевантные данные поставленной задачи, предлагает новые ракурсы поставленной задачи
4 Освоение стандартных алгоритмов решения профессиональных задач	В состоянии решать только фрагменты поставленной задачи в соответствии с заданным алгоритмом, не освоил предложенный алгоритм, допускает ошибки	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом	В состоянии решать поставленные задачи в соответствии с заданным алгоритмом, понимает основы предложенного алгоритма	Не только владеет алгоритмом и понимает его основы, но и предлагает новые решения в рамках поставленной задачи

9. ЗАКЛЮЧЕНИЕ

В данном учебно-методическом пособии по изучению дисциплины «Программирование на языках высокого уровня» представлены все темы лекционных занятий, контрольные вопросы по ним и ссылки на соответствующую литературу. Так же приведен порядок оценивания знаний по дисциплине и перечень вопросов для подготовки к экзамену. Литература, указанная в списке доступна в электронных библиотечных системах «Университетская книга» и «Лань».

СПИСОК ЛИТЕРАТУРЫ

Основная литература

1. Рагимханова, Г. С. Программирование на Python: учеб. пособие / Г. С. Рагимханова. – Махачкала: ДГПУ, 2022. – 126 с. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/330071> (дата обращения: 28.05.2024). – Режим доступа: для авториз. пользователей.

2. Титов, А. Н. Python. Обработка данных: учеб.-метод. пособие / А. Н. Титов, Р. Ф. Тазиева. – Казань: КНИТУ, 2022. – 104 с. – ISBN 978-5-7882-3171-6. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/331019> (дата обращения: 28.05.2024). – Режим доступа: для авториз. пользователей.

3. Букунов, С. В. Разработка приложений с графическим пользовательским интерфейсом на языке Python / С. В. Букунов, О. В. Букунова. – Санкт-Петербург: Лань, 2023. – 88 с. – ISBN 978-5-507-45191-3. – Текст : электронный // Лань : электронно-библиотечная система. – URL: <https://e.lanbook.com/book/292856> (дата обращения: 28.05.2024). – Режим доступа: для авториз. Пользователей.

Дополнительная литературы

4. Буйначев, С. К. Основы программирования на языке Python: учеб. пособие / С. К. Буйначев, Н. Ю. Боклаг; Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. – Екатеринбург: Издательство Уральского университета, 2014. – 92 с. : табл., ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=275962> (дата обращения: 21.03.2022). – Библиогр. в кн. – ISBN 978-5-7996-1198-9. – Текст : электронный.

5. Шелудько, В. М. Основы программирования на языке высокого уровня Python: учеб. пособие / В. М. Шелудько. – Ростов-на-Дону; Таганрог: Южный федеральный университет, 2017. – 147 с. : ил. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500056> (дата обращения: 21.03.2022). – Библиогр. в кн. – ISBN 978-5-9275-2649-9. – Текст : электронный.

6. Шелудько, В. М. Язык программирования высокого уровня Python: функции, структуры данных, дополнительные модули: учеб. пособие / В. М. Шелудько. – Ростов-на-Дону; Таганрог: Южный федеральный университет, 2017. – 108 с. – Режим доступа: по подписке. – URL: <https://biblioclub.ru/index.php?page=book&id=500060> (дата обращения:

21.03.2022). – Библиогр. в кн. – ISBN 978-5-9275- 2648-2. – Текст : электронный.

Учебно-методические пособия, нормативная литература

1. Заболотнова, Е. Ю. Программирование: учеб.-метод. пособие по выполнению лаб. работ для студентов направлений подгот. 09.03.01 Информатика и вычисл. техника и 09.03.03 Приклад. информатика / Е. Ю. Заболотнова, С. А. Калинина; Калинингр. гос. техн. ун-т. - Калининград: КГТУ, 2023. – 108 с.

2. Заболотнова, Е. Ю. Программирование на языках высокого уровня: учеб.-метод. пособие по вып. расчетно-графической работы для студентов бакалавриата по напр. подгот. 09.03.02 Информационные системы и технологии, 09.03.03 Прикладная информатика / Е. Ю. Заболотнова, С. А. Калинина – Калининград: Изд-во ФГБОУ ВО «КГТУ», 2024. – 26 с.

Интернет ресурсы

- Python 3 для начинающих URL: <https://pythonworld.ru/>
- Курс Python разработчик URL: <https://practicum.yandex.ru/backend-developer/>
- Официальный сайт Python URL: <https://www.python.org/>
- Бесплатный курс Python , самостоятельное обучение программированию с нуля URL : <https://code-basics.com/ru/languages/python>

Локальный электронный методический материал

Елена Юрьевна Заболотнова

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ

Редактор С. Кондрашова
Корректор Т. Звада

Уч.-изд. л. 1,4. Печ. л. 1,4.

Издательство федерального государственного бюджетного образовательного
учреждения высшего образования
«Калининградский государственный технический университет».
236022, Калининград, Советский проспект, 1