

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
**Федеральное государственное автономное образовательное учреждение высшего
образования**
**«ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СИСТЕМ
УПРАВЛЕНИЯ И РАДИОЭЛЕКТРОНИКИ»**
(ТУСУР)

Кафедра технологий электронного обучения (ТЭО)

Д.С. Шульц

ИНФОРМАТИКА

Методические указания к практическим занятиям, самостоятельной работе студентов,
обучающихся по направлениям подготовки:

- 27.03.02 Управление качеством, профиль «Управление качеством в информационных системах»
- 27.03.05 Инноватика, профиль «Управление инновациями»

Томск
2024

УДК 004.432.2

ББК 32.973.2

Ш95

Рецензент:

Перминова М.Ю., доцент кафедры технологий электронного обучения ТУСУР,
канд. техн. наук

Шульц Денис Сергеевич

Ш95 **Информатика:** Методические указания к практическим занятиям, самостоятельной работе студентов, обучающихся по направлениям подготовки 27.03.02 Управление качеством, 27.03.05 Инноватика / Д.С. Шульц. – Томск, 2024. – 12 с.

Методические указания к практическим занятиям и самостоятельной работе по дисциплине «Информатика» содержат необходимые разъяснения по форме организации практических занятий и самостоятельной работы студентов.

Одобрено на заседании каф. ТЭО протокол № 9 от 27.09.2024 г.

УДК 004.432.2

ББК 32.973.2

© Шульц Д.С. 2024

© Томск. Томск. гос. ун-т систем упр. и
радиоэлектроники, 2024

Оглавление

1 ВВЕДЕНИЕ	4
2 ОРГАНИЗАЦИЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ	5
3 МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ПРОВЕДЕНИЮ ПРАКТИЧЕСКИХ ЗАНЯТИЙ	6
3.1 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №1. РАЗРАБОТКА ПРОСТЫХ ПРОГРАММ	6
3.2 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №2. РАЗРАБОТКА ПРОГРАММ С ВЕТВЛЕНИЕМ	6
3.3 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №3. РЕШЕНИЕ ЗАДАЧ С ИСПОЛЬЗОВАНИЕМ ЦИКЛОВ	7
3.4 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №4. ОПИСАНИЕ ФУНКЦИЙ	7
3.5 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №5. СТРОКИ	8
3.6 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №6. СПИСКИ	8
3.7 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №7. КОРТЕЖИ	9
3.8 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №8. СЛОВАРИ	9
3.9 ПРАКТИЧЕСКОЕ ЗАНЯТИЕ №9. КОДИРОВАНИЕ ТЕКСТОВОЙ ИНФОРМАЦИИ	10
4 ЗАКЛЮЧЕНИЕ	11
СПИСОК РЕКОМЕНДУЕМОЙ ЛИТЕРАТУРЫ	12

1 Введение

Дисциплина «Информатика» является одними из важнейших общих математических и естественнонаучных дисциплин. Современный уровень развития вычислительной техники требует от специалистов высокого уровня знаний и навыков работы с компьютером для решения вопросов получения, хранения, преобразования, передачи и использования информации.

Целью изучения дисциплины является получение представления о предмете информатики, информации и её видах, основных методах передачи, обработки, накопления и управления информацией; получение полного представления о современных языках, системах, алгоритмах программирования.

Основными задачами изучения дисциплины «Информатика» являются:

- формирование практических навыков поиска, обработки, хранения информации посредством современных информационных технологий для решения задач в различных областях профессиональной деятельности;
- формирование потребности обращаться к компьютеру при решении задач из любой предметной области, базирующуюся на осознанном владении информационными технологиями и навыками взаимодействия с компьютером;
- освоение основных приемов разработки алгоритмов с выбором наиболее подходящих алгоритмов в зависимости от постановки задач;
- освоение приемов разработки компьютерных программ на языке высокого уровня, использующихся для решения задач профессиональной деятельности.

Данное руководство содержит методические указания к практическим занятиям и организации самостоятельной работы студентов.

2 Организация самостоятельной работы студентов

Целью самостоятельной работы является систематизация, расширение и закрепление теоретических знаний, использование материала, собранного и полученного в ходе подготовки к лабораторным работам и практическим занятиям.

Самостоятельная работа включает в себя проработку лекционного материала и подготовку к практическим занятиям, контрольным и лабораторным работам, изучение тем дисциплины, вынесенных на самостоятельное изучение.

Самостоятельная работа по освоению теоретического материала лекций, учебного пособия и другой учебной и научной литературы – это важнейшее условие формирования научного способа познания. Грамотная работа с книгой предполагает соблюдение ряда правил, главное из которых – это не механическое заучивание, не простое накопление цитат, выдержек, а сознательное усвоение прочитанного, осмысление его, стремление дойти до сути.

Изучение теоретической части дисциплин призвано не только углубить и закрепить знания, полученные на аудиторных занятиях, но и способствовать развитию у студентов творческих навыков, инициативы и организовать свое время.

Проработка лекционного материала включает:

- чтение студентами рекомендованной литературы и усвоение теоретического материала дисциплины;
- знакомство с Интернет-источниками, в том числе с теоретической информацией в электронном курсе;
- подготовку к различным формам контроля (контрольные работы);
- подготовку ответов на вопросы по различным темам дисциплины в той последовательности, в какой они представлены.

Планирование времени, необходимого на изучение дисциплин, студентам лучше всего осуществлять весь семестр, предусматривая при этом регулярное повторение материала.

Материал, законспектированный на лекциях, необходимо регулярно прорабатывать и дополнять сведениями из других источников литературы, представленных не только в программе дисциплины, но и в периодических изданиях.

При изучении дисциплины сначала необходимо по каждой теме прочитать рекомендованную литературу и составить краткий конспект основных положений, терминов, сведений, требующих запоминания и являющихся основополагающими в этой теме для освоения последующих тем курса. Для расширения знания по дисциплине рекомендуется использовать Интернет-ресурсы; проводить поиски в различных системах и использовать материалы сайтов, рекомендованных преподавателем.

Для удобства студентов им предоставляется доступ из личного кабинета в электронный курс. Он содержит теоретический материал, задания к лабораторным и контрольным работам.

Закрепить знания, полученные студентами при изучении теоретического материала, и оценить их уровень позволяет выполнение контрольных работ. Они находятся в электронном курсе и представляют собой совокупность тестовых заданий. Перед выполнением контрольных работ настоятельно рекомендуется повторить теоретический материал.

Задачи, стоящие перед студентом при подготовке и написании контрольной работы:

- закрепление полученных ранее теоретических знаний;
- выработка навыков самостоятельной работы;
- выяснение подготовленности студентов к зачету.

Поскольку не всегда возможно самостоятельно найти ответы на возникшие в ходе изучения дисциплины вопросы, в электронном курсе предусмотрен форум «Консультация».

Здесь студенты могут задавать вопросы не только по теоретическому материалу, но и по подготовке к занятиям, оформлению отчетов и т.д.

3 Методические указания к проведению практических занятий

Самостоятельная подготовка к практическому занятию заключается в следующем:

- в изучении соответствующего теоретического материала, размещенного в электронном курсе;
- в изучении соответствующих разделов в учебнике, дополнительной литературе, обучающих Интернет-ресурсах;
- в прохождении тестов для самоконтроля, размещенных в электронном курсе

Знания, полученные в процессе такой самостоятельной работы, являются теоретической и практической базой для решения задач на практическом задании.

3.1 Практическое занятие №1. Разработка простых программ

Цель занятия

Познакомиться с интегрированной средой разработки программ. Написать простейшие программы и произвести их запуск различными способами - через командную строку, с использованием IDLE, в PyCharm.

Рекомендации по подготовке к занятию.

Изучить указанные темы теоретического материала: данные и их типы, переменные, присваивание, ввод и вывод данных, комментирование программ.

Варианты заданий:

1. С начала суток прошло n секунд. Определить сколько полных часов, минут, секунд прошло с начала суток.
2. Составить программу вывода на экран в одну строку трех любых чисел с двумя пробелами между ними.
3. Составить программу вывода на экран "столбиком" четырех любых чисел.
4. Составить программу вывода на экран числа, вводимого с клавиатуры. Выводимому числу должно предшествовать сообщение "Вы ввели число".
5. Записать по правилам изучаемого языка программирования следующее выражение:

$$g(x) = \frac{1 + tgx + \sqrt{x+1}}{\sqrt[3]{x^2 + \frac{|1-x|}{x \sin|x|}} + e^{\cos x}}. \text{ Сделать проверку в одном из математических редакторов.}$$

3.2 Практическое занятие №2. Разработка программ с ветвлением

Цель занятия

Изучить правила записи условного оператора: полный и неполный условный оператор, вложенный условный оператор. Построение алгоритмов с ветвлениями и написание программ.

Рекомендации по подготовке к занятию.

Изучить теоретический материал: ветвление.

Варианты заданий:

1. Дано двузначное число n . Определить является ли сумма его цифр двузначным числом.
2. Даны вещественные числа a, b, c , при этом $a \neq 0$. Выяснить, имеет ли уравнение $ax^2 + bx + c = 0$ вещественные корни.

3. Определить, является ли число a делителем числа b или, наоборот, число b делителем числа a . Ответом должны служить сообщения "Да, одно из чисел является делителем другого" или "Нет, ни одно из чисел не является делителем другого".
4. Даны три различных вещественных числа. Не используя полный условный оператор, определить наибольшее из них.
5. Даны четыре целых числа. Определить, сколько из них четных

3.3 Практическое занятие №3. Решение задач с использованием циклов

Цель занятия

Научиться решать задачи с использованием циклов

Рекомендации по подготовке к занятию.

Изучить теоретический материал: операторы циклов (цикл While, цикл For, операторы break и continue)

Варианты заданий:

1. Дано целое число n . Определить сумму цифр этого числа.
2. Найдите факториал числа n .
3. Сформировать из введенного числа обратное по порядку входящих в него цифр и вывести на экран. Например, если введено число 1234, то надо вывести число 4321.
4. Посчитать четные и нечетные цифры введенного натурального числа.
5. Напечатать таблицу умножения на число n . Значение n вводится с клавиатуры и принадлежит промежутку: $1 \leq n \leq 9$.

3.4 Практическое занятие №4. Описание функций

Цель занятия

Научиться записывать алгоритмы решения задач в виде подпрограмм – функций.

Рекомендации по подготовке к занятию.

Изучить теоретический материал: функции (оператор def, передача параметров, область видимости, специальные режимы передачи).

Варианты заданий:

1. Напишите функцию `draw_box()`, которая выводит звездный прямоугольник с размерами 14×10 в соответствии с образцом:

```
*****
*      *
*****
```

2. Напишите функцию `draw_box()`, которая выводит звездный прямоугольник с размерами 14×10 в соответствии с образцом:

```
*
**
***
****
*****
```

3. Напишите функцию `draw_box(width, height)`, которая выводит звездный прямоугольник с размерами $width \times height$ в соответствии с образцом:

```
*****
*      *
*****
```

4. Напишите функцию `convert_to_miles(km)`, которая принимает в качестве аргумента расстояние в километрах и возвращает расстояние в милях. Формула для преобразования: мили = километры * 0.6214.
5. Напишите функцию `number_of_factors(num)`, принимающую в качестве аргумента число и возвращающую количество делителей данного числа.

3.5 Практическое занятие №5. Строки

Цель занятия

Освоить основные алгоритмы для работы со строками.

Рекомендации по подготовке к занятию.

Изучить теоретический материал: понятие строки, операции над строками, обращение к элементам строки, функции для работы со строками.

Варианты заданий:

1. Из слова «информатика» получить слова «форма» и «тик».
2. Слово «преподаватель» написано с ошибкой. Необходимо её исправить и вывести на экран верное написание.
3. Дано предложение на русском языке. Определить, сколько в нем гласных букв.
4. Написать программу, которая удаляет из введенной с клавиатуры строки все буквы а.
5. Оставить в строке только последнее вхождение каждого символа, взаимный порядок оставшихся символов сохранить

3.6 Практическое занятие №6. Списки

Цель занятия

Освоить основные алгоритмы для работы со списками.

Рекомендации по подготовке к занятию.

Изучить теоретический материал: списки, генератор списков, вывод списков на экран, операции со списками, функции и методы для работы со списками.

Варианты заданий:

1. Напишите функцию `get_factors(num)`, принимающую в качестве аргумента натуральное число и возвращающую список всех делителей данного числа.
2. При анализе данных, собранных в рамках научного эксперимента, бывает полезно удалить самое большое и самое маленькое значение. На вход программе подается натуральное число n , а затем n различных натуральных чисел. Напишите программу, которая удаляет наименьшее и наибольшее значение из указанных чисел, а затем выводит оставшиеся числа каждое на отдельной строке, не меняя их порядок.
3. На вход программе подается строка, содержащая английский текст. Напишите программу, которая подсчитывает общее количество артиклей: 'a', 'an', 'the'. Формат входных данных: на вход программе подается строка, содержащая английский текст. Слова текста разделены символом пробела. Формат выходных данных: программа должна вывести общее количество артиклей 'a', 'an', 'the' вместе с поясняющим текстом. Артикли могут начинаться с заглавной буквы 'A', 'An', 'The'.
4. Напишите программу, на вход которой подаётся список чисел одной строкой. Программа должна для каждого элемента этого списка вывести сумму двух его соседей. Для элементов списка, являющихся крайними, одним из соседей считается элемент, находящийся на противоположном конце этого списка. Например, если на вход подаётся список "1 3 5 6 10", то на выход ожидается список "13 6 9 15 7" (без кавычек). Если на вход пришло только одно число, надо вывести его же. Вывод должен содержать одну строку с числами нового списка, разделёнными пробелом.

5. Напишите программу, которая принимает на вход список чисел в одной строке и выводит на экран в одну строку значения, которые встречаются в нём более одного раза. Для решения задачи может пригодиться метод `sort` списка. Выводимые числа не должны повторяться, порядок их вывода может быть произвольным.

3.7 Практическое занятие №7. Кортежи

Цель занятия

Освоить основные алгоритмы для работы с кортежами.

Рекомендации по подготовке к занятию.

Изучить теоретический материал на тему: кортежи

Варианты заданий:

1. Рассчитайте статистику частоты букв в кортеже. Частота букв — это среднее количество раз, которое буквы алфавита встречаются в письменной речи.
2. Создайте кортеж с цифрами от 0 до 9 и посчитайте сумму.
3. Напишите функцию `tpl_sort()`, которая сортирует кортеж, состоит из целых чисел по возрастанию и возвращает его. Если хотя бы один элемент не является целым числом, то функция возвращает исходный кортеж.
4. На вход функции `sieve()` поступает список целых чисел. В результате выполнения этой функции будет получен кортеж уникальных элементов списка в обратном порядке.
5. Создайте кортеж из 7-ми именованных кортежей учащихся ВУЗов. В именованном кортеже будут присутствовать следующие поля: имя студента, возраст, оценка за семестр, город проживания. Функция `good_students()` будет принимать этот кортеж, вычислять среднюю оценку по всем учащимся и выводить на печать следующее сообщение: "Ученики {список имен студентов через запятую} в этом семестре хорошо учатся!". В список студентов, которые выводятся по результатам работы функции, попадут лишь те, у которых оценка за семестр равна или выше средней по всем учащимся.

3.8 Практическое занятие №8. Словари

Цель занятия

Освоить основные алгоритмы для работы со словарями.

Рекомендации по подготовке к занятию.

Изучить теоретический материал: словари, создание словарей, операции со словарями, функции для работы со словарями.

Варианты заданий:

1. На вход программы подается текст. Определить частоту встречаемости для каждого символа. Определить какая буква встречается наибольшее количество раз.
2. На вход программы подается текст. Определить какое слово встречается в тексте наибольшее количество раз. Слова состоят только из букв, знаки препинания нужно игнорировать.
3. Напишите программу, которая принимает на вход две строки и определяет, являются ли они анаграммами. Знаки препинания, пробелы и регистр при этом игнорируются.
4. Напишите функцию `update_dictionary(d, key, value)`, которая принимает на вход словарь `d` и два числа: `key`, `value`.
5. Напишите функцию `to_dict(lst)`, которая принимает аргумент в виде списка и возвращает словарь, в котором каждый элемент списка является и ключом, и значением. Предполагается, что элементы списка будут соответствовать правилам задания ключей в словарях.

3.9 Практическое занятие №9. Кодирование текстовой информации

Цель занятия

Изучить алгоритм кодирования текстовой информации с помощью специальных таблиц, изучить стандарт Unicode

Рекомендации по подготовке к занятию.

Изучить теоретический материал на тему: «Кодирование текстовой информации».

Варианты заданий:

1. Запишите Unicode следующего символа в формате UTF-8 (в двоичном коде):



♌ Лев

Знак зодиака

U+264C

2. Используя расширенную таблицу ASCII кодов (таблица кодов символов Windows-125), расшифруйте фразу, которая здесь закодирована:
C2 E2 E5 E4 E8 F2 E5 20 C2 E0 F8 E5 20 E8 EC FF
3. Используя расширенную таблицу кодов CP-866, расшифруйте слово, которое здесь закодировано: 170 160 228 165 164 224 160
4. Unicode символа записан в формате UTF-8 в двоичном коде:
11100010 10011000 10111010. Определите номер этого символа в Unicode (только цифры).
5. Используя расширенную таблицу ASCII кодов, расшифруйте слово, которое здесь закодировано: 164 168 175 171 174 172.

4 Заключение

Изучение методических указаний к практическим занятиям и самостоятельной работе по дисциплине «Информатика» способствует успешному ее освоению и развитию у обучающихся приемов разработки компьютерных программ на языке высокого уровня, использующихся для решения задач профессиональной деятельности.

Список рекомендуемой литературы

1. Чернышев, С. А. Основы программирования на Python: учебное пособие для вузов / С. А. Чернышев. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023. — 349 с. [Электронный ресурс]: — Режим доступа: <https://urait.ru/bcode/532446> (дата обращения: 15.10.2024).
2. Стивенсон, Б. Python. Сборник упражнений: учебное пособие / Б. Стивенсон; перевод с английского А. Ю. Гинько. — Москва: ДМК Пресс, 2021. — 238 с. [Электронный ресурс]: — Режим доступа: <https://e.lanbook.com/book/241025> (дата обращения: 15.10.2024).
3. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2023. — 227 с. [Электронный ресурс]: — Режим доступа: <https://urait.ru/bcode/532868> (дата обращения: 15.10.2024).
4. Северанс, Ч. Р. Python для всех / Ч. Р. Северанс; перевод с английского А. В. Снастина. — Москва: ДМК Пресс, 2021. — 262 с. [Электронный ресурс]: — Режим доступа: <https://e.lanbook.com/book/241115> (дата обращения: 15.10.2024).
5. Лутц, Марк. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО “Диалектика”, 2019. — 832 с. : ил.
6. Лутц, Марк. Изучаем Python, том 2, 5-е изд.: Пер. с англ. — СПб. : ООО “Диалектика”, 2020. — 720 с. : ил.
7. Лутц М. Программирование на Python, том I, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.
8. Лутц М. Программирование на Python, том II, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011. – 992 с., ил.