

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной
математики

УЧЕБНЫЕ МАТЕРИАЛЫ
для студентов заочной формы обучения

по дисциплине «Программирование»

для направления подготовки 09.03.01
Информатика и вычислительная техника

Общая трудоемкость дисциплины – 11 зачетных единиц.

Форма текущего контроля в 1 семестре – контрольная работа.

Форма промежуточного контроля в 1,2 семестре – экзамен.

Курсовая работа (КР) – 2 семестр.

Чита 2020 г.

Краткое содержание курса

Перечень изучаемых разделов дисциплины.

Семестр 1

1. Понятие программирования. Языки программирования высокого уровня. Язык программирования Python.
2. Интерактивный режим. Математические вычисления.
3. Литералы, переменные и типы данных.
4. Условный оператор. Тернарный оператор.
5. Циклический оператор. Цикл с предусловием. Совместный цикл.
6. Кортежи, списки, строки
7. Подпрограммы: процедуры и функции.
 - 7.1. Передача параметров и возврат значений.
 - 7.2. Понятие рекурсии. Простая и косвенная рекурсии.
8. Модульное программирование.

Семестр 2

9. Работа с внешней памятью. Файлы. Функции работы с файлами. Бинарные и текстовые файлы.
10. Обработка исключений.
11. Работы с графикой. Создание приложений с GUI.

Материалы курса (слайды лекции, примеры):

github.com/VetrovSV/Programming

Рекомендуемый онлайн-курс: stepik.org/course/67/promo

Связь с преподавателем: vetrov-zabgu@outlook.com

Форма текущего контроля

Контрольная работа (1 семестр)

Работа состоит из задач, которые требуется решить с использованием языка высокого уровня Python.

Решить задачу означает: разработать алгоритм, изучить все необходимые для кодирования средства языка программирования (операторы, типы данных, функции и модули), протестировать программу доказав её корректность.

Номера задач распределяются согласно приведенной ниже таблице вариантов и соответствуют номеру задачи в задачнике по программированию: ivtipm.github.io/Programming/Files/spisocall.htm (или файл Задачник.zip).

Непосредственно номер варианта – последняя цифра номера зачетной книжки.

При написании исходных текстов программ рекомендуется придерживаться стандарта оформления кода (PEP8, pythonworld.ru/osnovy/pep-8-rukovodstvo-po-napisaniyu-koda-na-python.html, habrahabr.ru/post/180509; проверка на соответствие PEP8: pep8online.com/checkresult).

Тема	№	Вариант									
		1	2	3	4	5	6	7	8	9	10
Арифметика действительных чисел	1	1	2	3	4	5	6	7	8	9	10
Разветвления	2	44	43	42	41	40	39	38	37	36	35б
Простейшая целочисленная арифметика	3	66	64	63	62	61	76в	76б	76а	74	73
Простейшие циклы	4	114б	114в	114г	114д	114е	114ж	114з	115а	115б	115в
Массивы. Простейшие массивы	5	136н	136д	136о	136н	136м	136л	136к	136и	136з	136ж
Массивы. Циклы и разветвления	6	178а	178б	178в	178г	178д	178е	179а	179б	180	181а
Вложенные циклы и итерации	7	320	334а	334б	334в	334г	335а	335б	335в	334г	336а
Обработка матриц	8	673	674	675	676а	676б	677а	677б	677в	677г	678
Использование процедур и функций	9	424	425	426	427	428	429	430	431	432	433
	10	692а	692б	692в	692г	692д	692е	692ж	692з	692и	692к

- Задачи 1-5 должны быть решены в двух вариантах: 1) без использования собственных процедур и модулей, 2) с собственными процедурами и модулями.
- Все последующие программы должны быть написаны согласно парадигме модульного программирования.
- Для ввода и вывода данных требуются отдельные функции.

Защита контрольных работ проходит на лабораторных занятиях. Для защиты нужно иметь исходные тексты программ в электронном виде. Программы оцениваются по критериям:

- корректность и надёжность — программа решает поставленную задачу и правильно функционирует для любых вариантов (из возможных) входных данных.
- эффективность — программа использует минимальное количество процессорного времени (вычислений) и памяти.
- сопровождаемость — код программы легко доступен для понимания и модификации.
- удобство использования — программа понятна пользователю, не требует от него лишней работы или запоминания.

Во время защиты задачи входящей в контрольную работу после запуска программы с различными входными параметрами требуется проверка: подтвердить корректность работы программы проведя вычисления иным способом (в ручную с калькулятором, в интерактивном режиме python, с помощью математических пакетов, excel или другим способом).

Помимо соответствия программы выше указным критериям, студент должен продемонстрировать полное понимание алгоритма программы и возможностей языка программирования по соответствующей теме. При необходимости нужно будет самостоятельно внести изменения в исходный код программы.

Алгоритмы, такие как поиск элемента в последовательности, сортировка, нахождения минимума и максимума и т.п. требуется реализовывать самостоятельно. Использование аналогичных алгоритмов, включённых в пакеты Python, не допускается.

Рекомендованные среды разработки:

- [PyCharm](#)
- [Jupyter](#) (входит в состав [Anaconda](#)). Подойдёт для экспериментов и приложений без GUI. Отсутствует отладчик.
- [Wing](#)

Оформление контрольной работы

Контрольная работа выполняется на листах формата А4. Рекомендуется скреплять листы степлером. Пример титульного листа приведён в приложении. Для каждого задания обязательно приводится условие и только после этого решение. Каждая задача начинается с нового листа.

Если сессия проходит дистанционно, то задачи загружаются на github – в специальный репозиторий, по предоставленной преподавателем ссылке.

Форма промежуточного контроля

Курсовая работа (второй семестр)

Защита курсовой работы проходит до начала экзамена. Работа состоит из программы и пояснительной записки к ней. Пояснительная записка оформляется согласно МИ 4.2-5/47-01-2013 ([Общие требования к построению и оформлению учебной текстовой документации](#)). Рекомендуется использовать стили при оформлении. Допускается использование LaTeX для оформления пояснительной записки. После защиты курсовой работы пояснительная записка (с подписью преподавателя) должна быть загружена в личный кабинет студента.

Пояснительная записка курсовой работы должна обязательно содержать следующие разделы:

- титульные листы, содержание,
- введение,
 - постановка задачи (техническое задание),
 - краткое описание средств разработки (Python, библиотеки)
- основная часть,
 - описание модульной структуры программы,
 - описание способов хранения данных,
 - описание созданных алгоритмов (сортировка, поиск и т. п.),
 - системные требования, краткое руководство пользователя.
- заключение,
- список использованных источников;
- приложения (включая исходный код программы).

В заключении, помимо описания разработанного приложения, можно включить описание опыта разработки, какие трудности возникли. Описать пути улучшения программы.

Разработку рекомендуется вести согласно принципу «от простого к сложному», проводя регулярное тестирование и резервное копирование.

Рекомендуется применять принципы модульного подхода. Разделять модель (бизнес-логику, игровую логику), взаимодействие с пользователем и отображение информации. Придерживаться стандарта оформления кода. Во время разработки целесообразно использование системы контроля версий, например, Git.

Если сессия проходит дистанционно, то работа загружается на github – в специальный репозиторий, по предоставленной преподавателем ссылке.

Задания для курсовых работ

Вариант курсовой работы соответствует последней цифре шифра зачётной книжки. Студент может предложить свою тему и утвердить её после обсуждения с преподавателем.

Нижеприведённые темы курсовых требуют создания программы для работы с файловой базой данных с одной таблицей. Формат записи данных в файл разрабатывается самостоятельно. Использование библиотек для работы с БД в форматах sqlite, xls и др допускается только после предварительного согласования с преподавателем. Программа должна предоставлять возможность просматривать, добавлять, удалять, копировать, редактировать, искать (по одному или нескольким полям), сортировать и хранить данные. Интерфейс пользователя — текстовый или графический.

Варианты курсовых работ

Вариант 1

Создать программу обслуживания телефонных абонентов, в которой создается база данных, содержащая сведения о номере абонента, Ф.И.О., лицевом счете.

Вариант 2

Создать программу обслуживания клиентов банка, в которой создается база данных, содержащая код клиента, лицевой счет (величина вклада, проценты по вкладу, даты и суммы вложений и изъятий в течении года).

Вариант 3

Создать программу обслуживания клиентов заправочной станции, в которой создается база данных, содержащая марку и номер машины, дату заправки, номер и количество отпускаемого бензина, суммы оплаты.

Вариант 4

Создать программу регистрации нарушителей ПДД, в которой создается база данных, содержащая марку и номер машины, дату нарушения, Ф.И.О., вид нарушения, номер квитанции, суммы оплаты.

Вариант 5

Создать программу регистрации подписчиков журнала, в которой создается база данных, содержащая Ф.И.О., название журнала, номер квитанции, сумму оплаты, срок подписки.

Вариант 6

Создать программу регистрации посетителей ресторана, в которой создается база данных, содержащая номер заказа, Ф.И.О. официанта, перечень заказанных блюд, сумму оплаты.

Вариант 7

Создать программу регистрации сотрудников, в которой создается база данных, содержащая Ф.И.О. сотрудника предприятия, номер трудовой книжки, дата трудоустройства, дата увольнения, причина увольнения, а также поощрения и наказания.

Вариант 8

Создать программу регистрации посетителей поликлиники, в которой создается база данных, содержащая Ф.И.О. больного, принимающий специалист, время приема, номер полиса.

Вариант 9

Создать программу учета материалов, в которой создается база данных, содержащая перечень расходуемых материалов на стройке. В перечень входят: наименование материала, количество, Ф.И.О. отпустившего, Ф.И.О. получившего.

Вариант 10

Создать программу, в которой создается база данных, содержащая информацию об экзопланетах: название, масса и размеры планеты; радиус орбиты; название звезды вокруг которой обращается планета.

Защита курсовой работы включает демонстрацию работы программы и исходного кода. Также во время защиты обязательно вносятся корректировки в программу в соответствии с замечаниями преподавателя для выяснения степени самостоятельности выполнения курсовой работы, студент объясняет некоторые участки программного кода. Студент отвечает на вопросы преподавателя о бизнес-логике программы, её реализации, исходном коде программы и процессе разработки. Код и программа должны демонстрировать высокий уровень и культуру владения языком программирования высокого уровня, а также правильный подход к разработке.

Критерии оценивания контрольных работ распространяются и на курсовую работу.

Экзамен (первый и второй семестр)

До экзамена во втором семестре допускаются студенты, сдавшие и защитившие контрольную работу (первый семестр) и защитившие курсовую работу (второй семестр).

Экзаменационный билет состоит из трёх вопросов: двух теоретических, одной задачи. Задачи выбираются подобными тем, что решались в контрольных работах или разбирались на занятиях.

Наличие сертификата онлайн-курса (соответствующего содержанию дисциплины) может быть учтено на экзамене.

Перечень примерных вопросов для подготовки к экзамену.

1. Типы данных.
2. Переменные.
3. Числовые типы данных.
4. Операции над числовыми типами данных.
5. Строки.
7. Вывод данных.
8. Ввод данных.
9. Форматированный ввод/вывод.
10. Списки.
11. Выражения в списках.
12. Оператор del.
13. Использование списков, как стеков.
14. Использование списков, как очередей.
15. Сравнение списков.
16. Диапазоны.
17. Кортежи. Отличие кортежей от словарей
18. Словари.
19. Оператор if.
20. Операторы цикла. Оператор for. Оператор while. Завершение цикла.
21. Продолжение цикла. Оператор pass.
22. Определение функции.
23. Пространство имен функции.
24. Передача параметров. Ключи.
25. Передача в функцию переменного числа аргументов.
26. Элементы функционального программирования.
27. Использование лямбда функций.
28. Функции работы со структурами данных.
29. Функция map(). Примеры применения
30. Функция filter(). Примеры применения
31. Функция reduce(). Примеры применения
32. Документирование функций.
33. Создание модулей.
34. Указание кодировки.
35. Поиск модулей.
36. Компиляция модулей на Python.
37. Стандартные модули Python.
38. Использование функции `__dir__`.
39. Структурирование модулей в пакеты.
40. Импорт модулей и их составляющих из пакета.
41. Ссылки в пакетах.
42. Пакеты и файловая система.
43. Класс File.
44. Открытие файла.
45. Методы класса для File ввода-вывода.
46. Взаимодействие с файловой системой.
47. Модуль path

Критерии формирования оценок экзамена

Экзамен проводится в устной форме: обсуждается теоретический материал и приводится решение практических заданий с объяснением.

Оценки "*отлично*" заслуживает студент, обнаруживший всестороннее, систематическое и глубокое знание учебно-программного материала, умение свободно выполнять задания, предусмотренные программой.

Оценки "*хорошо*" заслуживает студент обнаруживший полное знание учебно-программного материала, успешно выполняющий предусмотренные в программе задания. Ответ содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

Оценки "*удовлетворительно*" заслуживает студент выполнивший и защитивший практическое задание. Продемонстрированы знания основного учебно-программного материала в объёме, необходимом для дальнейшей учёбы, предстоящей работы по специальности. Эта оценка выставляется студентам, допустившим погрешности в ответе на экзамене и при выполнении экзаменационных заданий, но обладающим необходимыми знаниями для их устранения под руководством преподавателя.

Оценка "*неудовлетворительно*" выставляется студенту, обнаружившему пробелы в знаниях основного учебно-программного материала, допустившему принципиальные ошибки в выполнении предусмотренных программой заданий.

Учебно-методическое и информационное обеспечение дисциплины

Основная литература

1. Федоров, Д. Ю. Программирование на языке высокого уровня python : учебное пособие для прикладного бакалавриата / Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 126 с. — (Серия : Бакалавр. Прикладной курс). — ISBN 978-5-534-04479-9. — Режим доступа : biblio-online.ru/book/1EE056CF-F11A-4C18-8D33-40B703D49AC5
2. Гниденко, И. Г. Технологии и методы программирования : учебное пособие для прикладного бакалавриата / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — М. : Издательство Юрайт, 2017. — 235 с. — (Серия : Бакалавр. Прикладной курс). — ISBN 978-5-534-02816-4. — Режим доступа : biblio-online.ru/book/E0A213EF-E61B-4F8B-A4E5-D75FD4E72E10.
3. Python на практике [Электронный ресурс] / Марк Саммерфилд - М. : ДМК Пресс, 2014. - studentlibrary.ru/book/ISBN9785970600955.html

Дополнительная литература

1. *Лутц М. Изучаем Python. 2010. - 1280 с.*
2. Лутц М. Программирование на Python. 2011. - 992 с.
3. O`Conor T.J. Violent Python: A Cookbook for Hackers, Forensic Analysts, Penetration Testers and Security Engineers. 2012 — 288 p.
4. Введение в систему контроля версий Git. URL: youtube.com/watch?v=5EIfStQNDD4 (дата обращения: 21.06.2017)

Базы данных, информационно-справочные и поисковые системы

1. ЭБС Юрайт. – Режим доступа: biblio-online.ru
2. ЭБС "КОНСУЛЬТАНТ СТУДЕНТА". – Режим доступа: studentlibrary.ru
3. Библиотека ЗабГУ. – Режим доступа: library.zabgu.ru.
4. github.com/VetrovSV/Programming — Репозиторий с материалами по языку Python.

5. Табличное сравнение языков со схожим синтаксисом (шпаргалка по синтаксису). – Режим доступа: hyperpolyglot.org
6. Краткое описание языков программирования. – Режим доступа: learnxinyminutes.com
7. Система вопросов и ответов о программировании . – Режим доступа: stackoverflow.com

Ведущий преподаватель:

старший преподаватель кафедры ИВТ и ПМ

Ветров Сергей Владимирович.

vetrov-zabgu@outlook.com

Заведующий кафедрой ИВТ и ПМ

к. ф.-м. н., доцент О.В. Валова.

Приложение. Пример оформления контрольной работы

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра Информатики, вычислительной техники и прикладной
математики

КОНТРОЛЬНАЯ РАБОТА

по дисциплине: «**Программирование**»
** семестр

Вариант № **

Выполнил ст. гр. ИВТз-42
(группа)

Константинов К. К.
(фамилия, инициалы)

Проверил

ст. преп. Ветров С. В.
(должность, ученая степень, звание,
фамилия, инициалы)

Чита

20**

Задача 1

Описание условия задачи...

Например:

Найдите 10 наиболее частых слов на web странице

Для исходного кода программы используется моноширинный шрифт (как в среде программирования), например JetBrains Mono, PT Mono или Courier New

```
from urllib2 import urlopen      # подключение модуля urlopen из
библиотеки urllib2
u = urlopen("http://python.org") # загружается страница ...
words = {}                       # словарь для подсчёта числа слов

for line in u:                   # цикл по строкам из страницы
# очистка строки от начальных от пробелов и знаков переноса строки
    line = line.strip(" \n")
# разделение строки по пробелам, цикл с перебором полученных «слов»
    for word in line.split(" "):
# защищённый блок кода, в котором может возникнуть исключение
        try:
            words[word] += 1      # обращение к существующему элементу
словаря по ключу
        except KeyError:
            words[word] = 1      # создание нового элемента словаря

pairs = words.items()           # преобразование словаря в список из пар

pairs.sort(key=lambda x: x[1], reverse=True) # сортировка списка по
втором элементам

for p in pairs[:10]:           # цикл с перебором 10 первых элементов списка
    print(p[0], p[1])         # печать слова и его частоты
```

Отчёт о тестировании программы.

3 набора входных и выходных данных.