

МИНПРОСВЕЩЕНИЯ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
"Тульский государственный педагогический университет им. Л.Н. Толстого"
(ФГБОУ ВО "ТГПУ им. Л.Н. Толстого")

Практикум по программированию

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	институт передовых информационных технологий
ОПОП	Направление 09.03.03 Прикладная информатика направленность (профиль) Прикладная информатика в здравоохранении
Квалификация	Бакалавр
Год начала подготовки	2022
Форма обучения	очная
Общая трудоемкость	3 з.е.

Виды контроля по семестрам:
зачет 3

Семестр(Курс.Номер семестра на курсе)	3(2.1)		Итого	
	УП	РПД	УП	РПД
Лекции	18	18	18	18
Лабораторные	34	34	34	34
Итого ауд.	52	52	52	52
КСР	2	2	2	2
Контактная работа	54	54	54	54
Сам. работа	54	54	54	54
Часы на контроль	0	0	0	0
Практическая подготовка	0	0	0	0
Семинары	0	0	0	0
Консультации	0	0	0	0
Итого трудоемкость в часах	108	108	108	108

Программу составил(и):

к.п.н., доцент, Даниленко С.В.

Рабочая программа дисциплины

Практикум по программированию

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования - бакалавриат по направлению подготовки 09.03.03 Прикладная информатика (приказ Минобрнауки России от 19.09.2017 г. № 922)

составлена на основании учебного плана:

Направление 09.03.03 Прикладная информатика
направленность (профиль) Прикладная информатика в здравоохранении
утвержденного Учёным советом вуза от 28.02.2022 протокол № 3.

РПД утверждена Учёным советом университета
протокол от 28.2.2022 г. № 3

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Формирование комплекса знаний, умений и навыков в области применения современных алгоритмических языков программирования для
разработки программного обеспечения
Изучение дисциплины направлено на:
- развитие у обучающихся алгоритмического мышления,
- формирование навыков реализации различных алгоритмов на высокоуровневом языке программирования,
- применение знаний к решению прикладных задач

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.В
2.1	Требования к предварительной подготовке обучающегося:
1.	Объектно-ориентированное программирование
2.	ознакомительная практика
3.	Программирование
4.	Алгоритмизация и программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Методы и технологии программирования
2.	Основы проектирования веб-интерфейсов
3.	Практикум по веб-программированию
4.	технологическая (проектно-технологическая) практика
5.	Математические методы машинного обучения
6.	Программирование на языке Kotlin
7.	эксплуатационная практика
8.	научно-исследовательская работа
9.	Рекомендательные системы

3. СООТНЕСЕНИЕ РЕЗУЛЬТАТОВ ОБУЧЕНИЯ ПО ДИСЦИПЛИНЕ (МОДУЛЮ) С ИНДИКАТОРАМИ ДОСТИЖЕНИЯ КОМПЕТЕНЦИЙ

3.1 Компетенции обучающегося и индикаторы их достижения:	
ОПК-7: Способен разрабатывать алгоритмы и программы, пригодные для практического применения	
ОПК-7.1	Знает основные языки программирования и работы с базами данных, операционные системы и оболочки, современные программные среды разработки информационных систем и технологий
	Парадигмы, архитектурные черты, семантику и синтаксис языка программирования Python; Назначение, устройство и свойства основных структур данных и конструкций языка Python
ОПК-7.2	Умеет применять языки программирования и работы с базами данных, современные программные среды разработки информационных систем и технологий для автоматизации бизнес-процессов, решения прикладных задач различных классов, ведения баз данных и информационных хранилищ
	Разрабатывать математические методы и алгоритмы решения различных задач; Использовать для разработки и отладки программ интегрированные среды разработки
ОПК-7.3	Владеет навыками программирования, отладки и тестирования прототипов программно-технических комплексов задач
	Навыками чтения, написания, отладки и тестирования программ на высокоуровневом языке программирования в интегрированной среде разработки
ПК-3: Способен осуществлять ведение базы данных и поддержку информационного обеспечения решения прикладных задач	
ПК-3.1	Знает основные принципы построения баз данных и работы с ними; требования к поддержанию в работоспособном состоянии базы данных
	Модули и пакеты для решения различных прикладных и научных задач и методику их тестирования
ПК-3.2	Умеет разрабатывать программное обеспечение для ведения баз данных; осуществлять поддержку информационного обеспечения решения прикладных задач
	Умеет применять методы тестирования разработанного программного продукта
ПК-3.3	Владеет навыками работы с базами данных, как в локальной версии, так и с организацией архитектуры клиент-сервер
	Навыками чтения, написания, отладки и тестирования программ на высокоуровневом языке программирования в интегрированной среде разработки

3.2 Результаты обучения по дисциплине:**В результате освоения дисциплины обучающийся должен:**

	Знать:
3.1	Парадигмы, архитектурные черты, семантику и синтаксис языка программирования Python;
3.2	Назначение, устройство и свойства основных структур данных и конструкций языка Python;
3.3	Модули и пакеты для решения различных прикладных и научных задач и методику их тестирования.
	Уметь:
У.1	Разрабатывать математические методы и алгоритмы решения различных задач;
У.2	Использовать для разработки и отладки программ интегрированные среды разработки;
У.3	Умеет применять методы тестирования разработанного программного продукта
	Владеть:
В.1	Навыками чтения, написания, отладки и тестирования программ на высокоуровневом языке программирования в интегрированной среде разработки

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Раздел 1. Введение в Python				
1.1	Краткая история Python, особенности и типы данных /Лек/	3	2	Л1.1Л2.1	Краткая история языка программирования Python, особенности языка Python, интернированные (intern) объекты в Python. Глобальная блокировка интерпретатора. Подходы к сборке мусора в Python (алгоритм подсчета ссылок, Garbage Collector (GC), слабые ссылки). Встроенные типы данных Python
1.2	Основы языка программирования Python /Лаб/	3	2	Л1.1Л2.1	Переменные в языке программирования Python. Функции в языке программирования Python. Программы в отдельном файле. Область видимости переменных. Применение функций. Схема передачи чисел в качестве аргументов функций
1.3	Основы языка программирования Python /Ср/	3	4	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 2. Синтаксис, операторы и управляющие конструкции				
2.1	Основные операторы в Python /Лек/	3	2	Л1.1Л2.1	Процедурные операторы; операторы функций; операторы классов; операторы исключений; операторы модулей; арифметические операторы; операторы сравнения; операторы присваивания; побитовые операторы; логические операторы; операторы членства; операторы тождественности. Использование отступов в Python Комментарии Правила именования переменных (имен)
2.2	Основные операторы и управляющие конструкции /Лек/	3	2	Л1.1Л2.1	Синтаксис условного оператора if. Логические выражения. Схема использования условной конструкции. Оператор (цикл) while в языке Python. Синтаксис. Цикл с конструкцией else. Работа цикла с операторами break, continue, pass. Цикл for. Синтаксис. Различные способы написания циклов

2.3	Условный оператор if /Лаб/	3	2	Л1.1Л2.1	Синтаксис условного оператора. Логические выражения. Схема использования условной конструкции. Решение задач.
2.4	Работа с циклами /Лаб/	3	2	Л1.1Л2.1	Различные способы написания циклов. Использование встроенной функции range. Использование встроенной функции zip. Использование встроенной функции enumerate
2.5	Итерации и включения /Лаб/	3	2	Л1.1Л2.1	Итератор и итерируемый объект. Пример итерации по итерируемому объекту, используемый циклом for. Итерация в цикле for по списку my_list. Списковые включения (list comprehension)
2.6	Основные операторы и управляющие конструкции /Ср/	3	4	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 3. Работа со строками в Python				
3.1	Строки и операции над строками /Лек/	3	2	Л1.1Л2.1	Операции над строками. Дополнительные возможности функции print. Ввод значений с клавиатуры
3.2	Работа со строками /Лаб/	3	2	Л1.1Л2.1	Решение задач со строками. Операции над строками.
3.3	Работа со строками в Python /Ср/	3	4	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 4. Функции в Python				
4.1	Функции в Python /Лек/	3	2	Л1.1Л2.1	Принципы объявления и использования функций, механизм замыканий и правило LEGB, принцип работы с рекурсивными функциями, отличие генератора от списковых включений; механизм декорирования функций.
4.2	Функции в Python /Лаб/	3	4	Л1.1Л2.1	Использование генераторных функций и выражений при работе с циклом for, передача аргументов в функцию и получение возвращаемого ею результата; написание приложений с использованием функций, использование декораторов и генераторов в процессе написания приложений.
4.3	Функции в Python /Ср/	3	4	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 5. Модули и пакеты				
5.1	Принципы организации модулей и пакетов /Лек/	3	2	Л1.1Л2.1	Принципы организации модулей и пакетов, существующие механизмы импортирования модулей и пакетов, а также их отдельных атрибутов; встроенные модули и пакеты; создание собственных модулей; автоматизированное тестирование функций.
5.2	Основы работы с модулями и пакетами /Лаб/	3	4	Л1.1Л2.1	Создание и импортирование собственных модулей, создание и импортирование собственных пакетов; написание приложений с использованием модулей и пакетов.
5.3	Модули и пакеты /Ср/	3	6	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 6. Списки и файлы				

6.1	Списки и файлы /Лек/	3	2	Л1.1Л2.1	Операции над списками. Псевдонимы и копирование списков. Методы списка. Преобразование типов. Схема передачи списков в качестве аргументов функции Работа с файлами. Байтовый тип данных. Регулярные выражения. Работа с форматом JSON. Работа с открытыми данными
6.2	Работа со списками /Лаб/	3	2	Л1.1Л2.1	Создание и обработка списков. Операции со списками. Копирование списков. Передача списков в качестве аргументов функций
6.3	Работа с файлами /Лаб/	3	2	Л1.1Л2.1	Разработка программ для работы с файлами в Python. Схема определения относительного пути. Работа с форматом JSON. Работа с открытыми данными
6.4	Списки и файлы /Ср/	3	6	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 7. Классы и объектно-ориентированное программирование				
7.1	Классы и объектно-ориентированное программирование /Лек/	3	2	Л1.1Л2.1	Принципы объектно-ориентированного программирования, механизм реализации классов в Python и работы с ними, способы работы с перечислениями (Enum), существующие отличия между переменными класса и переменными экземпляра класса
7.2	Работа с классами. Объектно-ориентированное программирование. /Лаб/	3	2	Л1.1Л2.1	Создание пользовательских классов; объявление (абстрактных) базовых классов и их производных классов, перегрузка методов класса; написание кода на языке программирования Python в объектно-ориентированной парадигме.
7.3	Классы и объектно-ориентированное программирование /Ср/	3	6	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 8. Исключения				
8.1	Исключения в Python /Лек/	3	2	Л1.1Л2.1	Механизм исключений в Python, основные способы генерации и обработки исключений
8.2	Исключения в Python /Лаб/	3	2	Л1.1Л2.1	Создание и генерирование пользовательских исключений, использование операторов try/except/finally и with/as; перехват и обработка исключений, возникающих в процессе работы приложений, основные принципы работы с исключениями.
8.3	Исключения /Ср/	3	6	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 9. Потoki, процессы и асинхронное программирование				
9.1	Работа с потоками в Python /Лаб/	3	2	Л1.1Л2.1	Особенности работы с модулями threading, multiprocessing и asyncio, разница между многопоточным, асинхронным и многопроцессорным программированием в Python. Создание классов, которые могут запускаться в отдельном потоке или процессе, организация доступа к разделяемым ресурсам.
9.2	Межпроцессорное и многопоточное взаимодействие /Лаб/	3	2	Л1.1Л2.1	Организация межпроцессорного и многопоточного взаимодействия; разработка и отладка многопоточных, многопроцессорных и асинхронных приложений.

9.3	Потоки, процессы и асинхронное программирование /Ср/	3	8	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
	Раздел 10. Разработка графического пользовательского интерфейса				
10.1	Принципы разработки графических пользовательских интерфейсов /Лаб/	3	2	Л1.1Л2.1	Принципы разработки графических пользовательских интерфейсов (GUI), различия между библиотекой PySide2 и PyQt5 для разработки приложений с графическим пользовательским интерфейсом; компоновка виджетов GUI создаваемого приложения
10.2	Разработка приложений с графическим пользовательским интерфейсом /Лаб/	3	2	Л1.1Л2.1	Механизм сигнал-слотового взаимодействия между классами в рамках фреймворка Qt; разработка приложений с графическим пользовательским интерфейсом.
10.3	Разработка графического пользовательского интерфейса /Ср/	3	6	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
10.4	Контроль самостоятельной работы /КСР/	3	2	Л1.1Л2.1	Контрольная работа

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

ПЕРЕЧЕНЬ ЛАБОРАТОРНЫХ РАБОТ (Описание работ представлено в LMS MOODLE):

1. Основы языка программирования Python
2. Условный оператор if
3. Работа с циклами
4. Итерации и включения
3. Функции в Python
4. Основы работы с модулями и пакетами
5. Работа со списками
6. Работа с файлами
7. Работа с классами. Объектно-ориентированное программирование
8. Исключения в Python
9. Работа с потоками в Python
10. Межпроцессорное и многопоточное взаимодействие
11. Принципы разработки графических пользовательских интерфейсов
12. Разработка приложений с графическим пользовательским интерфейсом

Примеры заданий для самостоятельной работы:

Раздел 1. Основы языка программирования Python

1. Напишите скрипт для чтения последних n строк файла.
2. Напишите скрипт, подсчитывающий количество строк в файле.
3. Напишите скрипт, позволяющий найти самое встречаемое слово в файле.
4. Умножьте каждый элемент списка на произвольное число.
5. Найдите максимальное и минимальное числа, хранящиеся в списке.
6. Напишите скрипт, удаляющий все повторяющиеся элементы из списка.
7. Скопируйте список двумя различными способами.

Раздел 2.

1. Выведите все символы из строки «Данная часть была посвящена больше синтаксису Python и вопросам документации кода», значения индексов которых делятся на 2.
2. Выведите все символы из строки «Данная часть была посвящена больше синтаксису Python и вопросам документации кода», значения индексов которых без остатка делятся на 3, но не делятся на 4.
3. Выведите все символы из строки «Данная часть была посвящена больше синтаксису Python и вопросам документации кода», значения индексов которых при делении на 6 дают остаток 2, 4, и 5.
4. Выведите числа из диапазона от 1 до 10, используя цикл for и while.
5. Выведите числа из диапазона от -20 до 20 с шагом 3, используя цикл for и while.
6. Посчитайте количество вхождений элемента со значением «3» в следующем списке: ГЗ 0 I 304334 56 6 1 31, используя цикл for, while и метод count.

Раздел 3.

1. Напишите программу для определения суммы и произведения трех целочисленных значений, введенных с клавиатуры.

Пример выполнения программы:

Введите первое число: 1

Введите второе число: 4

Введите третье число: 7

Сумма введенных чисел: 12

Произведение введенных чисел: 28

2. Напишите программу, которая по атомному номеру элемента в периодической таблице химических элементов определяет его название. В качестве входных значений необходимо использовать атомные номера Li, Mg, Hg, Cl.

Раздел 4.

1. Напишите функцию, вычисляющую максимальное из трех чисел.
2. Напишите функцию, которая возвращает сумму элементов списка.
3. Напишите функцию, которая возвращает произведение элементов списка.
4. Напишите функцию, которая возвращает инвертированную строку, подаваемую ей на вход.
5. Напишите функцию для вычисления факториала задаваемого числа.
6. Напишите функцию, которая проверяет, входит ли задаваемое значение в определенный диапазон или нет.
7. Напишите функцию, которая подсчитывает количество элементов в нижнем и верхнем регистрах у входной строки.

Раздел 5.

1. Напишите модуль, содержащий функции, которые выполняют следующие арифметические операции: сложение, вычитание, умножение.
2. Напишите модуль, содержащий функции, которые выполняют следующие операции: проверка наличия элемента в списке, подсчет частоты вхождения элемента в список.
3. Напишите модуль, содержащий функции, которые выполняют следующие операции: проверку, является ли строка палиндромом, подсчет длины строки, перевод всех символов в нижний регистр.
4. Напишите модуль, содержащий функции, которые выполняют следующие операции: подсчет площади круга, прямоугольника и треугольника.
5. Напишите модуль, содержащий функции, которые выполняют следующие операции: подсчет количества элементов в словаре, проверку на наличие ключа в словаре.

Раздел 6.

Задан список $L = [3, 6, 7, 4, -5, 4, 3, -1]$.

1. Определите сумму элементов списка L. Если сумма превышает значение 2, то необходимо вывести на экран число элементов списка.
 2. Определите разность между минимальным и максимальным элементами списка. Если абсолютное значение разности больше 10, то необходимо вывести на экран отсортированный по убыванию список, иначе вывести на экран фразу «Разность меньше 10».
2. Задан список $L = [3, 'hello', 7, 4, 'привет', 4, 3, -1]$. Определите наличие строки 'привет' в списке. Если такая строка в списке присутствует, то вывести ее на экран, повторив 10 раз.

Раздел 7.

1. Напишите класс, реализующий все арифметические операции над двумя значениями (a и b).
2. Напишите класс, описывающий такой объект, как автомобиль. Продумайте, какие методы и переменные он должен иметь.
3. Напишите класс, описывающий такой объект, как прямоугольник. Перегрузите у реализованного класса методы сравнения (сравнивать по площади), после чего создайте два экземпляра класса и проверьте, как работают перегруженные методы.
4. Напишите класс, описывающий такой объект, как автомобиль. У него может быть различное количество состояний, реализуемых посредством перечислений. Добавьте методы, позволяющие экземпляру класса менять свое текущее состояние (например: остановка, движение, поворот налево и т. д.).

Раздел 8.

1. Напишите класс, реализующий все арифметические операции над двумя значениями (a и b). В случае, если одно из значений при вызове операции равно нулю, генерируется исключение.
2. Напишите класс, реализующий такие арифметические действия, как деление и умножение. Если одно из значений при вызове операции равно нулю, генерируется исключение, значение ноль меняется на 1 и вычисление операции продолжается.
3. Напишите функцию, возводящую строку в верхний регистр. Добавьте проверку на то, что на вход функции подается не пустая строка.

Раздел 9.

1. Напишите программу, которая осуществляет чтение данных из файла посредством одного потока и запись этих данных в другом потоке в файл. Названия файлов должны отличаться.
2. Напишите программу, которая осуществляет чтение из файла посредством одного процесса и запись этих данных

в другом процессе в файл. Названия файлов должны отличаться.

3. Напишите программу, которая, используя механизм асинхронного программирования, осуществляет чтение данных из одного файла и их запись в другой файл.
4. Напишите программу, осуществляющую перемножение двух матриц с использованием потоков и процессов. Каждая строка новой матрицы должна высчитываться в отдельном потоке (процессе).

Раздел 10.

1. Напишите программу «Калькулятор».
2. Напишите программу «Список дел» с возможностью сохранения данных в файл и их загрузки для редактирования.
3. Напишите программу, выводящую содержимое задаваемого каталога.
4. Напишите программу для игры в «крестики-нолики».
5. Напишите программу для просмотра изображений в задаваемом каталоге.
6. Напишите программу для решения квадратных уравнений.
7. Напишите программу, позволяющую создавать новые текстовые файлы, заполнять их, сохранять, загружать и редактировать, а также осуществлять поиск по тексту.

5.2. Типовые задания для проведения промежуточной аттестации

Вопросы для собеседования на зачете

1. В чем разница между сильными и слабыми ссылками?
2. Что такое интернированные объекты?
3. Как можно интернировать строку, содержащую не ASCII-символы?
4. Что такое GIL? В чем его особенность?
5. Как работает алгоритм подсчета ссылок?
6. Опишите принцип работы Garbage Collector.
7. Зачем используется подход с поколениями при работе Garbage Collector?
8. Для чего можно использовать слабые ссылки?
9. Назовите четыре основных типа данных Python.
10. Что означает «неизменяемость» и какие три основных типа Python считаются неизменяемыми?
11. Как связаны циклы for и итерируемые объекты?
12. Как связаны циклы for и списковые включения?
13. Когда должны использоваться строки документации вместо комментариев #?
14. Назовите три способа, которыми можно просматривать строки документации.
15. Как можно получить список доступных атрибутов в объекте?
16. Как можно получить список доступных модулей на компьютере?
17. Какую книгу, посвященную Python, вы должны приобрести после этой?
18. В чем смысл написания функций?
19. В какой момент Python создает функцию?
20. Что функция возвращает, если в ней нет ни одного оператора return?
21. Как оператор from связан с оператором import?
22. Как функция reload связана с операциями импортирования?
23. Когда вы обязаны использовать import вместо from?
24. Назовите три потенциальных затруднения, связанных с оператором from.
25. Каким образом файл исходного кода модуля становится объектом модуля?
26. Затем вам может понадобиться установка переменной среды PYTHONPATH?
27. Назовите пять главных компонентов в пути поиска импортируемых модулей.
28. Назовите четыре типа файлов, которые Python может загрузить в ответ на операцию импорта.
29. Что такое пространство имен и что содержит пространство имен модуля?
30. Что важно знать о переменных на верхнем уровне модуля, чьи имена начинаются с одиночного подчеркивания?
31. Назовите ключевые отличия в принципах работы модулей threading, multiprocessing и asyncio.
32. В каких случаях лучше применять асинхронное программирование?
33. Почему при вычислениях, интенсивно использующих CPU, не следует использовать многопоточное программирование?
34. Назовите способы межпроцессорного взаимодействия.
35. За что отвечает класс Timer в модуле threading?
36. Для чего используется блокировка доступа к общим ресурсам?

5.3. Перечень видов оценочных средств

Задания для самостоятельной работы

Зачет

5.4. Процедура применения оценочных материалов

Промежуточная аттестация может проводиться с применением электронного обучения и (или) дистанционных образовательных технологий в соответствии с "Порядком проведения промежуточной аттестации с применением

электронного обучения и /или дистанционных образовательных технологий".

Представлены в Приложении файл ФОС_Практикум по программированию

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
6.1. Рекомендуемая литература				
6.1.1. Основная литература				
	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л1.1	Окулов С. М., Иванов К. С., Бондарева Л. В.	Задачи по программированию	Лаборатория знаний, 2017	http://biblioclub.ru/index.php?page=book&id=561640
6.1.2. Дополнительная литература				
	Авторы, составители	Заглавие	Издательство, год (кол-во экземпляров для печатных изданий)	Ссылка на электронное издание
Л2.1	Хиценко В. П.	Основы программирования: учебное пособие	НГТУ, 2015	http://biblioclub.ru/index.php?page=book&id=438365
6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"				
Э1	Портал Министерства науки и высшего образования и РФ			
Э2	Сайт ГОУ ВО ТПУ им. Л.Н. Толстого			
Э3	Интуит. Национальный открытый университет [Электронный ресурс]			
6.3. Информационные технологии				
6.3.1 Перечень лицензионного и свободно распространяемого программного обеспечения				
1.	Операционная система ROSA Enterprise Linux Desktop № RL00450-1-110518-01. RL00450-1-110518-17 от 11 мая 2018 г.			
2.	Операционная система Microsoft Windows XP Professional Russian. Лицензия № 16698685 от 08.08.2003 г.			
3.	Операционная система Microsoft Windows Professional 7 Russian. Лицензия №48497058 от 13.05.2011 г., договор № Пр/16/6 от 05 апреля 2016 г.			
4.	Операционная система Microsoft Windows 10 Professional Russian. Контракт № ПР/ФЕН/15/18 от 23.10.2015 г., договор № Пр/16/6 от 05 апреля 2016 г.			
5.	Программное обеспечение Microsoft Office Enterprise 2007 Russian. Лицензия №46138962 от 16.11.2009			
6.	Программное обеспечение Microsoft Office 2013 Professional. Контракт № 405535 от 2 ноября 2015 года, контракт № ПР/ФЕН/15/18 от 23.10.2015 г.			
7.	Программа для распознавания текста ABBYY FineReader 9.0 Corporate Edition. Лицензионный сертификат - код позиции AF90-3U1V25-102, ABBYY FineReader 9.0 Corporate Edition Volume License Concurrent от 28 июля 2009 г.			
8.	Электронный словарь ABBYY Lingvo X3 Европейская версия - Код позиции AL14-2U1V05-102, ABBYY Lingvo x3 Европейская версия. Именная лицензия Concurrent от 28 июля 2009 г.			
9.	Комплексная система антивирусной защиты Kaspersky Endpoint Security для бизнеса – стандартный Russian Edition. 500-999 Node 2 year Educational Renewal License. Лицензия № 13C8-190514-084943-783-1256 от 15.05.2019			
10.	Файловый архиватор 7z. Свободно распространяемое ПО			
11.	Браузеры Google Chrome, Mozilla, Opera. Свободно распространяемое ПО			
12.	Текстовый редактор NotePad++. Свободно распространяемое ПО			
13.	Пакет офисных приложений Apache OpenOffice 4.1.6. Свободно распространяемое ПО			
14.	Программа просмотра файлов формата RPD Adobe Acrobat Reader DC. Свободно распространяемое ПО			
15.	Файловый менеджер Far manager. Свободно распространяемое ПО			
6.3.2 Перечень информационных справочных систем и профессиональных баз данных				
1.	Официальный интернет-портал базы данных правовой информации (http://pravo.gov.ru)			
2.	Портал Федеральных государственных образовательных стандартов высшего образования (http://fgosvo.ru)			
3.	Компьютерная информационно-правовая система «Гарант»			
7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)				
Ауд.	Назначение	Оборудование и технические средства обучения		Вид

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-304	Лекционная с мультимедийным комплексом	доска учебная, проектор, стол преподавателя, столы учебные, стул преподавателя, экран	Лек
4-305	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, аудиоколонки учебные, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лаб
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Лаб
4-302	Учебная аудитория	доска учебная, проектор, стол преподавателя, столы учебные, стул преподавателя, экран	Лек
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	Лаб
4-303	Помещение для самостоятельной работы	аудиоколонки, кондиционер, маркерная доска, столы компьютерные, столы учебные, компьютерная техника с возможностью подключения сети "Интернет" и обеспечением доступа в электронную информационно-образовательную среду университета	Ср
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	КСР
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Зачёт

8. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)

Для успешного освоения дисциплины студент должен посещать лекции и лабораторные занятия, вовремя выполнять все задания и самостоятельные работы, пользоваться основной и дополнительной литературой, рекомендованной настоящей программой. Во время лекции рекомендуется вести краткий конспект.

Навыки программирования на языке Python студент приобретает на лабораторных занятиях. Для выполнения самостоятельных заданий каждому студенту необходимо:

- получить вариант задания у преподавателя;
- скачать документ с описанием заданий, размещенный в среде электронного обучения ТППУ им. Л.Н. Толстого.
<http://moodle.tsput.ru>

- внимательно прочитать задание. Вопросы, возникающие в процессе выполнения самостоятельных работ, необходимо грамотно сформулировать и обратиться за помощью к преподавателю на ближайшем занятии;

- разработать, отладить и оттестировать программы, решающие поставленные задачи.

Для каждого самостоятельного задания предусмотрена процедура защиты, в ходе которой преподаватель проводит устный опрос студентов для контроля понимания выполненных ими заданий.

При подготовке к зачету в дополнение к изучению конспектов лекций и учебной литературы, рекомендуется посещать консультации и пользоваться свободными Интернет-ресурсами.