

МИНОБРНАУКИ РОССИИ

**Федеральное государственное бюджетное образовательное учреждение
высшего образования «Горно-Алтайский государственный университет»
(ФГБОУ ВО ГАГУ, Горно-Алтайский государственный университет)**

**Высокоуровневые методы информатики и
программирования**
рабочая программа дисциплины (модуля)

Закреплена за кафедрой	кафедра экономики, туризма и прикладной информатики		
Учебный план	09.03.03_2018_828.rlx 09.03.03 Прикладная информатика Прикладная информатика в экономике		
Квалификация	бакалавр		
Форма обучения	очная		
Общая трудоемкость	6 ЗЕТ		
Часов по учебному плану	216	Виды контроля	в семестрах:
в том числе:		экзамены	6
аудиторные занятия	100		
самостоятельная работа	78,5		
часов на контроль	34,75		

Распределение часов дисциплины по семестрам

Семестр (<Курс>.<Семестр на курсе>)	6 (3.2)		Итого	
	19 5/6			
Неделя				
Вид занятий	УП	РП	УП	РП
Лекции	30	30	30	30
Лабораторные	70	70	70	70
Консультации (для студента)	1,5	1,5	1,5	1,5
Контроль самостоятельной работы при проведении аттестации	0,25	0,25	0,25	0,25
Консультации перед экзаменом	1	1	1	1
В том числе инт.	20	20	20	20
Итого ауд.	100	100	100	100
Контактная работа	102,75	102,75	102,75	102,75
Сам. работа	78,5	78,5	78,5	78,5
Часы на контроль	34,75	34,75	34,75	34,75
Итого	216	216	216	216

Программу составил(и):

к.ф.м.н., доцент, Губкина Елена Владимировна



Рабочая программа дисциплины

Высокоуровневые методы информатики и программирования

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА (уровень бакалавриата) (приказ Минобрнауки России от 12.03.2015 г. № 207)

составлена на основании учебного плана:

09.03.03 Прикладная информатика

утвержденного учёным советом вуза от 02.11.2017 протокол № 11.

Рабочая программа утверждена на заседании кафедры

кафедра экономики, туризма и прикладной информатики

Протокол от 21.05.2018 протокол № 10

Зав. кафедрой Куттубаева Тосканай Айтмуқановна



Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2018-2019 учебном году на заседании кафедры кафедры экономики, туризма и прикладной информатики

Протокол от _____ 2018 г. № ____
Зав. кафедрой Куттубаева Тосканай Айтмуқановна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2019-2020 учебном году на заседании кафедры кафедры экономики, туризма и прикладной информатики

Протокол от _____ 2019 г. № ____
Зав. кафедрой Куттубаева Тосканай Айтмуқановна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2020-2021 учебном году на заседании кафедры кафедры экономики, туризма и прикладной информатики

Протокол от 21 мая _____ 2020 г. № 10
Зав. кафедрой Куттубаева Тосканай Айтмуқановна

Визирование РПД для исполнения в очередном учебном году

Рабочая программа пересмотрена, обсуждена и одобрена для исполнения в 2021-2022 учебном году на заседании кафедры кафедры экономики, туризма и прикладной информатики

Протокол от _____ 2021 г. № ____
Зав. кафедрой Куттубаева Тосканай Айтмуқановна

1. ЦЕЛИ И ЗАДАЧИ ОСВОЕНИЯ ДИСЦИПЛИНЫ	
1.1	<i>Цели:</i> формирование у будущих специалистов практических навыков по разработке программного обеспечения (ПО) для решения экономических и расчетных задач с применением современных методов и технологий программирования, обучение работе с научно-технической литературой и технической документацией по программному обеспечению ПЭВМ.
1.2	<i>Задачи:</i> изучения дисциплины является реализация требований, установленных в квалификационной характеристике, при подготовке бакалавров в области разработки программного обеспечения в системах машинной обработки экономической и другой информации, проектирования и разработки этих систем.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП	
Цикл (раздел) ООП:	Б1.Б
2.1	Требования к предварительной подготовке обучающегося:
2.1.1	Базы данных
2.1.2	Информатика и программирование
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
2.2.1	
2.2.2	
2.2.3	Программная инженерия

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)	
ОПК-1: способностью использовать нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий	
Знать:	
основные нормативно правовые документы, международные и отечественные стандарты в области информационных систем и технологий стандарты для языка Python	
Уметь:	
использовать нормативно-правовые документы использовать для программирования и настройки стандарты языка Python	
Владеть:	
навыками использования официального сайта языка Python	
ПК-2: способностью разрабатывать, внедрять и адаптировать прикладное программное обеспечение	
Знать:	
основы разработки программ на языке высокого уровня	
Уметь:	
разрабатывать программы на языке высокого уровня Python	
Владеть:	
навыками разработки и адаптации программ на Python	
ПК-8: способностью программировать приложения и создавать программные прототипы решения прикладных задач	
Знать:	
основы языка программирования Python	
Уметь:	
программировать на Python. Решать задачи прикладного характера	
Владеть:	
навыками разработки приложений на Python для решения прикладных задач	
ПК-12: способностью проводить тестирование компонентов программного обеспечения ИС	
Знать:	
основные компоненты ИС основы составления тестовых примеров и задач	
Уметь:	
составлять тестовые примеры для тестирования	

Владеть:
навыками тестирования программ

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)							
Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Компетенции	Литература	Инте ракт.	Примечание
	Раздел 1. Основы программирования Python						
1.1	Типы данных и составные объекты Python Ввод и вывод на Python. Форматированный ввод и вывод. Типизация Python. Специальные типы. Числа: целые, с плавающей точкой и комплексное. Строки. Последовательности. Отображения. Арифметические выражения. Составные объекты. Списки. Дополнительные возможности при конструировании списков. Кортежи. Множества. Словари. Файлы. Вспомогательные типы. Стеки. Очереди. Работа с файлами на диске Презентация /Лек/	6	2	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	1	
1.2	Типы данных и составные объекты Python Ввод и вывод на Python. Форматированный ввод и вывод. Типизация Python. Специальные типы. Числа: целые, с плавающей точкой и комплексное. Строки. Последовательности. Отображения. Арифметические выражения. Составные объекты. Списки. Дополнительные возможности при конструировании списков. Кортежи. Множества. Словари. Файлы. Вспомогательные типы. Стеки. Очереди. Работа с файлами на диске /Лаб/	6	4	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
1.3	Типы данных и составные объекты Python Ввод и вывод на Python. Форматированный ввод и вывод. Типизация Python. Специальные типы. Числа: целые, с плавающей точкой и комплексное. Строки. Последовательности. Отображения. Арифметические выражения. Составные объекты. Списки. Дополнительные возможности при конструировании списков. Кор-тежи. Множества. Словари. Файлы. Вспомогательные типы. Стеки. Очереди. Работа с файлами на диске Выполнение задач для индивидуальной работы /Ср/	6	7	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	

1.4	Установка и настройка среды программирования Python Знакомство с интерфейсом среды программирования Python. Запуск программ, написанных на Python через командную строку ОС. Стандартные модули Python. Загрузка модулей в среду программирования и выполнение функция модуля. /Лаб/	6	2	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
1.5	Управляющие конструкции ходом выполнения программ на Python Полное и неполное ветвление. Реализация многовариантного выбора через ветвления. Параметрический цикл с вариативной и без вариативной части. Цикл с предусловием. Команды прерывания цикла и ее продолжения со следующей итерации (break, continue). Выполнение команд внутри контекста (with). Обработка исключений в программах. Встроенные типы исключений и определение новых исключений. /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	1	
1.6	Управляющие конструкции ходом выполнения программ на Python Полное и неполное ветвление. Реализация многовариантного выбора через ветвления. Пара-метрический цикл с вариативной и без вариативной части. Цикл с предусловием. Команды прерывания цикла и ее продолжения со следующей итерации (break, continue). Выполнение команд внутри контекста (with). Обработка исключений в программах. Встроенные типы исключений и определение новых исключений. Метод проектов. Работа над индивидуальным заданием /Лаб/	6	4	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
1.7	Управляющие конструкции ходом выполнения программ на Python Полное и неполное ветвление. Реализация многовариантного выбора через ветвления. Пара-метрический цикл с вариативной и без вариативной части. Цикл с предусловием. Команды прерывания цикла и ее продолжения со следующей итерации (break, continue). Выполнение команд внутри контекста (with). Обработка исключений в программах. Встроенные типы исключений и определение новых исключений. Выполнение задач для индивидуальной работы /Ср/	6	4	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	

1.8	Пользовательские функции и основы функционального программирования Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. Функции, как объекты и замыкания. Декораторы. Итераторы. Генераторы и сопрограммы. Генераторы списков. Выражения-генераторы. /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
1.9	Пользовательские функции и основы функционального программирования Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. Функции, как объекты и замыкания. Декораторы. Итераторы. Генераторы и сопрограммы. Генераторы списков. Выражения-генераторы. Метод проектов. Работа над индивидуальным заданием /Лаб/	6	4	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
1.10	Пользовательские функции и основы функционального программирования Определение пользовательских функций. Передача параметров и возврат результатов. Значения аргументов функции по умолчанию. Произвольный набор аргументов. Именованные аргументы. Правила видимости. Функции, как объекты и замыкания. Декораторы. Итераторы. Генераторы и сопрограммы. Генераторы списков. Выражения-генераторы. Выполнение задач для индивидуальной работы /Ср/	6	4	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
Раздел 2. Применение языка Python в различных областях знаний							
2.1	Математические модули Модуль чисел с плавающей точкой Decimal. Модуль рациональных чисел Fractions. Модуль стандартных математических функций Math. Модуль абстрактных базовых классов Numbers. Модуль псевдослучайных чисел Random. Модуль для работы с комплексными числами CMath. Модуль для работы с массивами Array. Модуль сортировки списков Bisect. Презентация /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	1	

2.2	Математические модули Модуль чисел с плавающей точкой Decimal. Модуль рациональных чисел Fractions. Модуль стандартных математических функций Math. Модуль абстрактных базовых классов Numbers. Модуль псевдослучайных чисел Random. Модуль для работы с комплексными числами CMath. Модуль для работы с массивами Array. Модуль сортировки списков Bisect. Метод проектов. Работа над индивидуальным заданием /Лаб/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
2.3	Математические модули Модуль чисел с плавающей точкой Decimal. Модуль рациональных чисел Fractions. Модуль стандартных математических функций Math. Модуль абстрактных базовых классов Numbers. Модуль псевдослучайных чисел Random. Модуль для работы с комплексными числами CMath. Модуль для работы с массивами Array. Модуль сортировки списков Bisect. Выполнение задач для индивидуальной работы /Ср/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.4	Python для анализа данных Эффективные библиотеки работы с данными на Python: numpy, pandas. Визуализация данных на Python /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.5	Python для анализа данных Эффективные библиотеки работы с данными на Python: numpy, pandas. Визуализация данных на Python /Лаб/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.6	Python для анализа данных Эффективные библиотеки работы с данными на Python: numpy, pandas. Визуализация данных на Python Выполнение задач для индивидуальной работы /Ср/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.7	Python для экономико-математических методов и моделей Python в экономико-математических расчетах. Дискретной математике. Работа с графами Презентация /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	1	
2.8	Python для экономико-математических методов и моделей Python в экономико-математических расчетах. Дискретной математике. Работа с графами Метод проектов. Работа над индивидуальным заданием /Лаб/	6	10	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
2.9	Python для экономико-математических методов и моделей Python в экономико-математических расчетах. Дискретной математике. Работа с графами Выполнение задач для индивидуальной работы /Ср/	6	11,5	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.10	Python для Internet Web-программирование. Презентация /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	1	

2.11	Python для Internet Web-программирование.Метод проектов. Работа над индивидуальным заданием /Лаб/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
2.12	Python для Internet Web-программирование. Выполнение задач для индивидуальной работы /Ср/	6	10	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.13	Python и создание графических приложений Презентация /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	1	
2.14	Python и создание графических приложений. Метод проектов. Работа над индивидуальным заданием /Лаб/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
2.15	Python и создание графических приложений Выполнение задач для индивидуальной работы /Ср/	6	9	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.16	Работа с базами данных /Лек/	6	4	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.17	Работа с базами данных/ Метод проектов. Работа над индивидуальным заданием /Лаб/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	2	
2.18	Работа с базами данных Выполнение задач для индивидуальной работы /Ср/	6	9	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.19	Работа с текстовыми данными Кодировки символов на Python. Операции над строками. Стандартные модули обработки строк String и Codecs. Модуль приблизительного сравнения двух строк DiffLib. /Лек/	6	3	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.20	Работа с текстовыми данными Кодировки символов на Python. Операции над строками. Стандартные модули обработки строк String и Codecs. Модуль приблизительного сравнения двух строк DiffLib. /Лаб/	6	6	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
2.21	Работа с текстовыми данными Кодировки символов на Python. Операции над строками. Стандартные модули обработки строк String и Codecs. Модуль приблизительного сравнения двух строк DiffLib. Выполнение задач для индивидуальной работы /Ср/	6	8	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
Раздел 3. Консультации							
3.1	Консультация по дисциплине /Конс/	6	1,5	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
Раздел 4. Промежуточная аттестация (экзамен)							
4.1	Подготовка к экзамену /Экзамен/	6	34,75	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	

4.2	Контроль CP /КСРАтт/	6	0,25	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	
4.3	Контактная работа /КонсЭк/	6	1	ОПК-1 ПК-2 ПК-8 ПК-12	Л1.1Л2.1 Л2.2	0	

5. ФОНД ОЦЕНОЧНЫХ СРЕДСТВ

5.1. Контрольные вопросы и задания

Типы данных.
 Переменные.
 Числовые типы данных.
 Операции над числовыми типами данных.
 Строки.
 Строки unicod.
 Вывод данных.
 Ввод данных.
 Форматированный ввод/вывод.
 Списки.
 Выражения в списках.
 Оператор del.
 Использование списков, как стеков.
 Использование списков, как очередей
 Операции сравнения для списков
 Диапазоны.
 17.Кортежи. Отличие кортежей от словарей
 Словари.
 Оператор if. Особенности операторов сравнения.
 Операторы цикла. Оператор for. Оператор while. Завершение цикла.
 Продолжение цикла. Оператор pass.
 Определение функции.
 Пространство имен функции.
 Передача параметров.
 Передача в функцию переменного числа аргументов.
 Элементы функционального программирования.
 Стандартные модули Python.
 Стандартные модули обработки строк String и Codecs
 Модуль для работы с массивами Array
 Модуль сортировки списков Bisect
 Модуль для работы с Web
 Модуль для работы с БД

5.2. Темы письменных работ

Не предусмотрено

Фонд оценочных средств

Фосы хранятся отдельным документом

6. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

6.1. Рекомендуемая литература

6.1.1. Основная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л1.1	Шелудько В.М.	Основы программирования на языке высокого уровня Python: учебное пособие	Ростов-на-Дону: Издательство Южного федерального университета, 2017	http://www.iprbookshop.ru/87461.html

6.1.2. Дополнительная литература

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
--	---------------------	----------	-------------------	-----------

	Авторы, составители	Заглавие	Издательство, год	Эл. адрес
Л2.1	Маккинли Уэс, Слинкина А.	Python и анализ данных: научное издание	Саратов: Профобразование, 2019	http://www.iprbookshop.ru/88752.html
Л2.2	Шелудько В.М.	Язык программирования высокого уровня Python. Функции, структуры данных, дополнительные модули: учебное пособие	Ростов-на-Дону: Издательство Южного федерального университета, 2017	http://www.iprbookshop.ru/87530.html

6.3.1 Перечень программного обеспечения	
6.3.1.1	MS WINDOWS
6.3.1.2	Яндекс.Браузер
6.3.1.3	Python
6.3.1.4	Kaspersky Endpoint Security для бизнеса СТАНДАРТНЫЙ
6.3.1.5	MS Office
6.3.1.6	NVDA
6.3.2 Перечень информационных справочных систем	
6.3.2.1	База данных «Электронная библиотека Горно-Алтайского государственного университета»
6.3.2.2	Электронно-библиотечная система IPRbooks

7. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ	
	кейс-метод
	презентация

8. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)		
Номер аудитории	Назначение	Основное оснащение
319 А2	Компьютерный класс. Учебная аудитория для проведения занятий лекционного типа, занятий семинарского типа, курсового проектирования (выполнения курсовых работ), групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации. Помещение для самостоятельной работы	Рабочее место преподавателя. Посадочные места обучающихся (по количеству обучающихся). Компьютеры, интерактивная доска с проектором, подключение к сети интернет

9. МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО ОСВОЕНИЮ ДИСЦИПЛИНЫ (МОДУЛЯ)
<p>1. Планирование и организация времени, необходимого для изучения дисциплины. Рекомендуется следующим образом организовать время, необходимое для изучения дисциплины:</p> <p>Изучение конспекта лекции в тот же день, после лекции – 10-15 минут. Изучение конспекта лекции за день перед следующей лекцией – 10-15 минут. Изучение теоретического материала по учебнику и конспекту – 1 час в неделю. Подготовка к лабораторному занятию – 30 мин. Всего в неделю – 2 часа 55 минут.</p> <p>2. Описание последовательности действий студента («сценарий изучения дисциплины»).</p> <p>При изучении дисциплины очень полезно самостоятельно изучать материал, который еще не прочитан на лекции. Тогда лекция будет гораздо понятнее. Однако легче при изучении курса следовать изложению материала на лекции. Для понимания материала и качественного его усвоения рекомендуется такая последовательность действий:</p> <p>1. После прослушивания лекции и окончания учебных занятий, при подготовке к занятиям следующего дня, нужно сначала просмотреть и обдумать текст лекции, прослушанной</p>

сегодня (10-15 минут).

2. При подготовке к лекции следующего дня, нужно просмотреть текст предыдущей лекции, подумать о том, какая может быть тема следующей лекции (10-15 минут).

3. В течение недели выбрать время (1 час) для работы с литературой по алгоритмическим методам в библиотеке или изучить дополнительную литературу в электронной форме.

3. Методические рекомендации по подготовке семинарских и практических занятий.

По данному курсу предусмотрены лабораторные занятия в компьютерном классе. При подготовке к лабораторным занятиям следует изучить соответствующий теоретический материал по языку программирования Python.

Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучаются и книги по программированию.

Литературу по курсу «Язык программирования Python» рекомендуется изучать на сайтах <http://www.python.org/>, <http://www.enthought.com/products/epdlibraries.php>, которые содержат в свободном доступе все необходимые дистрибутивы, полную информацию и уроки по программированию на языке Python. Полезно использовать несколько учебников по курсу «Программирование на Python». Однако легче освоить курс придерживаясь одного учебника и конспекта. Рекомендуется, кроме «заучивания» материала, добиться состояния понимания изучаемой темы дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений по программированию на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?. При изучении теоретического материала всегда нужно рисовать схемы или графики.

Необходимо изучить лабораторную работу предыдущего занятия и выяснить те вопросы, которые показались непонятными. Полезно вначале попытаться написать программный код самостоятельно, а затем сравнить его с тем, что был рассмотрен на предыдущем занятии. Такой подход позволяет студентам быстрее освоить алгоритмические методы языка и сократить время на его изучение.

4. Рекомендации по работе с литературой. Теоретический материал курса становится более понятным, когда дополнительно к прослушиванию лекции и изучению конспекта, изучаются и книги по программированию. Литературу по курсу «Язык программирования Python» рекомендуется изучать на сайтах <http://www.python.org/>, <http://www.enthought.com/products/epdlibraries.php>, которые содержат в свободном доступе все необходимые дистрибутивы, полную информацию и уроки по программированию на языке Python. Полезно использовать несколько учебников по курсу «Программирование на Python». Однако легче освоить курс придерживаясь одного учебника и конспекта.

Рекомендуется, кроме «заучивания» материала, добиться состояния понимания изучаемой темы дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений по программированию на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?. При изучении теоретического материала всегда нужно рисовать схемы или графики.

5. Советы по подготовке к экзамену. Дополнительно к изучению конспектов лекции необходимо пользоваться учебником по программированию. Кроме «заучивания» материала зачета, очень важно добиться состояния понимания изучаемых тем дисциплины. С этой целью рекомендуется после изучения очередного параграфа выполнить несколько простых упражнений по программированию на данную тему. Кроме того, очень полезно мысленно задать себе следующие вопросы (и попробовать ответить на них): о чем этот параграф?, какие новые понятия введены, каков их смысл?. При изучении теоретического материала всегда нужно рисовать схемы или графики. В конце подготовки к зачету полезно самостоятельно написать программу для проверки усвоения материала.