

МИНОБРНАУКИ РОССИИ

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

Функциональное программирование

рабочая программа дисциплины (модуля)

Закреплена за кафедрой	информатики и информационных технологий
ОПОП	02.03.03 Математическое обеспечение и администрирование информационных систем направленность (профиль) Информационные системы и базы данных
Квалификация	Бакалавр
Год начала подготовки	2019
Форма обучения	очная
Общая трудоемкость	3 з.е.

Виды контроля по семестрам:
зачет 8

Семестр(Курс.Номер семестра на курсе)	8(4.2)		Итого	
	УП	РПД	УП	РПД
Лекции	18	18	18	18
Лабораторные	22	22	22	22
Итого ауд.	40	40	40	40
КСР	2	2	2	2
Контактная работа	42	42	42	42
Сам. работа	66	66	66	66
Часы на контроль	0	0	0	0
Итого трудоемкость в часах	108	108	108	108

Программу составил(и):

к.ф.-м.н., доцент, Ванькова В.С.; к.п.н., доцент, Мартынюк Ю.М.; ассистент, Гладких И.Ю.

Рабочая программа дисциплины

Функциональное программирование

разработана в соответствии с ФГОС:

Федеральный государственный образовательный стандарт высшего образования по направлению подготовки 02.03.03 Математическое обеспечение и администрирование информационных систем (уровень бакалавриата) (приказ Минобрнауки России от 23.08.2017 г. № 809)

составлена на основании учебного плана:

02.03.03 Математическое обеспечение и администрирование информационных систем
направленность (профиль) Информационные системы и базы данных
утвержденного Учёным советом вуза от 30.05.2019 протокол № 6.

Рабочая программа одобрена на заседании кафедры

информатики и информационных технологий

Зав. кафедрой Богатырева Ю.И.

РПД утверждена Учёным советом университета
протокол от 30.5.2019 г. № 6

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Формирование у студентов общих методологических основ и практических навыков разработки программных систем с использованием функционального подхода к программированию

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ООП

Цикл (раздел) ООП:	Б1.В
2.1	Требования к предварительной подготовке обучающегося:
1.	К началу изучения дисциплины студенты должны владеть знаниями в области представления информации, основных типов данных и алгоритмических конструкций языков программирования процедурной парадигмы, этапов решения задач с помощью компьютера, умениями разрабатывать алгоритмы и на их основе реализовывать программные коды на языках программирования процедурной парадигмы, иметь навыки отладки и тестирования программ в определенной среде программирования.
2.	практика по получению первичных навыков научно-исследовательской работы
3.	эксплуатационная практика
4.	Теория графов
5.	Теория и методика изучения информатических дисциплин в системе дополнительного образования
6.	Системы компьютерной математики
7.	Структуры и алгоритмы компьютерной обработки данных
8.	Технология визуализации данных
9.	Дискретная математика
10.	Программирование
11.	Системное и прикладное программное обеспечение
12.	технологическая практика
13.	Информатика
14.	Основы медиапсихологии
15.	Психологические основы программирования
2.2	Дисциплины и практики, для которых освоение данной дисциплины (модуля) необходимо как предшествующее:
1.	Дисциплина является основой для научно-исследовательской работы и прохождения производственной практики по получению профессиональных умений и опыта профессиональной деятельности и преддипломной практики.

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

3.1 Компетенции обучающегося и индикаторы их достижения:

ПК-1: Способен демонстрировать базовые знания математических и естественных наук, программирования и информационных технологий

ПК-1.1	Обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий
	знает основные особенности функционального подхода к разработке программного обеспечения; роль и место функциональных языков программирования в индустрии проектирования программных систем; особенности представления и интерпретации функциональных программ; соответствие между функциональными и императивными программами.
ПК-1.2	Умеет находить, формулировать и решать стандартные задачи в собственной научно-исследовательской деятельности в области программирования и информационных технологий
	умеет использовать функциональные методы при проектировании прикладного программного обеспечения вычислительной техники; решать задачи администрирования информационных систем с применением пакетов для создания функциональных программ.
ПК-1.3	Имеет практический опыт научно-исследовательской деятельности в области программирования и информационных технологий
	имеет навыки разработки функциональных программ на языке Haskell; использования инструментальных средств для создания ПО на базе функциональных методов программирования; преобразования императивных программ в функциональные с использованием языков C и Haskell; проведения сравнительного анализа императивных и функциональных программ.

3.2 Результаты обучения по дисциплине:

В результате освоения дисциплины обучающийся должен:

	Знать:
3.1	основные особенности функционального подхода к разработке программного обеспечения;
3.2	роль и место функциональных языков программирования в индустрии проектирования программных систем;

3.3	особенности представления и интерпретации функциональных программ;
3.4	соответствие между функциональными и императивными программами.
Уметь:	
У.1	использовать функциональные методы при проектировании прикладного программного обеспечения вычислительной техники;
У.2	решать задачи администрирования информационных систем с применением пакетов для создания функциональных программ.
Владеть:	
В.1	разработки функциональных программ на языке Haskell;
В.2	использования инструментальных средств для создания ПО на базе функциональных методов программирования;
В.3	преобразования императивных программ в функциональные с использованием языков С и Haskell;
В.4	проведения сравнительного анализа императивных и функциональных программ;

4. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Код занятия	Наименование разделов и тем /вид занятия/	Семестр / Курс	Часов	Литература	Содержание
	Теоретический основы функционального программирования				
1.1	Общие сведения о функциональном подходе к программированию. /Лек/	8	2	Л1.1Л2.1	Поколения языков программирования. Императивный, объектно-ориентированный, логический и функциональный подходы к программированию – достоинства, недостатки и основные характеристики. Понятие функции и функциональной программы. История развития языков функционального программирования. Программирование при помощи функций. Программирование при помощи процедур.
1.2	Строго функциональный язык /Лек/	8	4	Л1.1Л2.1	Символьные данные и S-выражения. Типизация в функциональных языках. Строгая типизация. Списки и деревья. Элементарные селекторы и конструкторы с примерами применения. Элементарные предикаты и арифметика с примерами применения. Рекурсивные функции. Примеры применения рекурсивных функций на различных задачах обработки списков. Проблема выбора подфункций. Проблема модульности функциональной программы. Возможность накапливающих параметров на примере инверсии списка. Локальные определения в функциональных программах. Лямбда-исчисление А. Черча – теоретические сведения и машинная реализация. Точечная запись выражений.

1.3	Основы функционального программирования на языке Haskell /Лек/	8	4	Л1.1Л2.1	Синтаксис и особенности языка функционального программирования Haskell 98. Стандартные классы Haskell и типы данных. Конструкторы типов и конструкторы данных. Полиморфные типы и полиморфные функции с примерами применения. Типы, определяемые пользователем. Понятие и принцип действия механизма pattern-matching (соответствия образцу). Функции как объекты первого класса (first-class objects). Сигнатура типа функции. Понятие каррированных и некаррированных функций. Инфиксные операторы. Синонимы типов как средство повышения читабельности функциональных программ. Включение списков и арифметические последовательности. Генераторы. Лямбда-абстракции в Haskell. Функции высших порядков в Haskell. Отложенные вычисления. Механизмы организации массивов в Haskell и простейшие примеры их обработки. Примеры организации более сложных программ.
1.4	Представление и интерпретация функциональных программ /Лек/	8	2	Л1.1Л2.1	Представление и интерпретация функциональных программ. Абстрактная и конкретная формы программ. Проблема связывания значений и переменных. Техника интерпретации функциональных программ.
1.5	Соответствие между функциональными и императивными программами. /Лек/	8	2	Л1.1Л2.1	Особенности интерпретирования императивных программ. Функциональные эквиваленты императивных программ. Преобразование императивных программ в функциональные. Аппаратное обеспечение функциональных программ.
1.6	Функциональный подход к параллелизму /Лек/	8	2	Л1.1Л2.1	Задержка вычислений. Замедленные вычисления. Сети связанных процессов.
1.7	Функции высших порядков /Лек/	8	2	Л1.1Л2.1	Типы функций и понятие высшего порядка функций. Композиция функций. Примеры применения функций высшего порядка.
1.8	Подготовка к зачету /Ср/	8	6	Л1.1Л2.1	Подготовка к зачету
	Практические основы функционального программирования				
2.1	Интерпретатор языка Haskell. Знакомство с компилятором функционального языка Haskell /Лаб/	8	4	Л1.1Л2.1	Основы работы с интерпретатором Hugs. Типы данных. Арифметика. Кортежи. Списки. Строки. Функции. Условные выражения. Функции многих переменных и порядок определения функций.
2.2	Рекурсивные функции /Лаб/	8	4	Л1.1Л2.1	Комментарии. Рекурсия. Операция выбора и правила выравнивания. Кусочное задание функций. Сопоставление с образцом. Построение списков. Некоторые полезные функции
2.3	Рекурсивные функции /Ср/	8	12	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
2.4	Преобразование императивных программ в функциональные /Лаб/	8	4	Л1.1Л2.1	let-связывание. Сигнализация об ошибках. Охраняющие условия. Полиморфные типы. Пользовательские типы
2.5	Преобразование императивных программ в функциональные /Ср/	8	12	Л1.1Л2.1	Выполнение заданий для самостоятельной работы

2.6	Разработка функциональных программ для простейшей обработки списков и деревьев с использованием рекурсии /Лаб/	8	4	Л1.1Л2.1	Определение операторов. Рекурсивные типы. Списки как рекурсивные типы. Синтаксические деревья.
2.7	Разработка функциональных программ для простейшей обработки списков и деревьев с использованием рекурсии. /Ср/	8	12	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
2.8	Исследование и применение лямбда-абстракций и функций высших порядков /Лаб/	8	4	Л1.1Л2.1	Функции высшего порядка. Лямбда-абстракции. Секции.
2.9	Исследование и применение лямбда-абстракций и функций высших порядков /Ср/	8	12	Л1.1Л2.1	Выполнение заданий для самостоятельной работы
2.10	Модульный принцип программирования /Лаб/	8	2	Л1.1Л2.1	Модули. Абстрактные типы данных. Синонимы типов. Операции ввода-вывода. Базовые операции ввода-вывода. Стандартные операции ввода-вывода. Создание исполняемых программ
2.11	Модульный принцип программирования /Ср/	8	12	Л1.1Л2.1	Выполнение заданий для самостоятельной работы

5. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ

5.1. Типовые задания для проведения текущего контроля

Примерные варианты заданий для самостоятельного выполнения:

Приведите пример нетривиальных выражений, принадлежащих следующему типу:

```
((Char,Integer), String, [Double])
[(Double,Bool,(String,Integer))]
([Integer],[Double],[Bool,Char])
[[[(Integer,Bool)]]]
(((Char,Char),Char),[String])
((([Double],[Bool]),[Integer])
```

Требование нетривиальности в данном случае означает, что встречающиеся в выражениях списки должны содержать больше одного элемента.

Определите следующие функции:

Функция `max3`, по трем целым возвращающая наибольшее из них.

Функция `min3`, по трем целым возвращающая наименьшее из них.

Функция `sort2`, по двум целым возвращающая пару, в которой наименьшее из них стоит на первом месте, а наибольшее - на втором.

Функция `bothTrue :: Bool -> Bool -> Bool`, которая возвращает `True` тогда и только тогда, когда оба ее аргумента будут равны `True`. Не используйте при определении функции стандартные логические операции (`&&`, `||` и т.п.).

Функция `solve2 :: Double -> Double -> (Bool, Double)`, которая по двум числам, представляющим собой коэффициенты линейного уравнения $ax + b = 0$, возвращает пару, первый элемент которой равен `True`, если решение существует и `False` в противном случае; при этом второй элемент равен либо значению корня, либо `0.0`.

Определите функцию, принимающую на вход целое число n и возвращающую список, содержащий n элементов, упорядоченных по возрастанию.

- 1) Список натуральных чисел.
- 2) Список нечетных натуральных чисел.
- 3) Список четных натуральных чисел.

3 n -е треугольное число tn равно количеству одинаковых монет, из которых можно построить равносторонний треугольник, на каждой стороне которого укладывается n монет. Нетрудно убедиться, что $t1 = 1$ и $tn = n + tn?1$

4 n -е пирамидальное число pn равно количеству одинаковых шаров, из которых можно построить правильную пирамиду с

треугольным основанием, на каждой стороне которой укладывается n шаров. Нетрудно убедиться, что $p_1 = 1$ и $p_n = t_n + p_{n-1}$

Определите следующие функции:

- 1) Функция, принимающая на входе список вещественных чисел и вычисляющую их арифметическое среднее. Постарайтесь, чтобы функция осуществляла только один проход по списку.
- 2) Функция вычленения n -го элемента из заданного списка.
- 3) Функция сложения элементов двух списков. Возвращает список, составленный из сумм элементов списков-параметров. Учтите, что переданные списки могут быть разной длины.
- 4) Функция перестановки местами соседних четных и нечетных элементов в заданном списке

В современных web-магазинах часто продают книги, видеокассеты и компакт-диски. База данных такого магазина для каждого типа товаров должна содержать следующие характеристики:

- Книги: название и автор
- Видеокассеты: название
- Компакт-диск: название, исполнитель и количество композиций

- 1) Разработайте тип данных `Product`, который может представлять эти виды товаров.
- 2) Определите функцию `getTitle`, возвращающую название товара.
- 3) На ее основе определите функцию `getTitles`, которая по списку товаров возвращает список их названий.
- 4) Определите функцию `bookAuthors`, которая по списку товаров возвращает список авторов книг.
- 5) Определите функцию `lookupTitle :: String -> [Product] -> Maybe Product` которая возвращает товар с заданным названием (обратите внимание на тип результата функции)
- 6) Определите функцию `lookupTitles :: [String] -> [Product] -> [Product]` Она принимает в качестве параметров список названий и список товаров и для каждого названия извлекает из второго списка соответствующие товары. Названия, которым не соответствует никакой товар, игнорируется. При определении функции обязательно используйте функцию `lookupTitle`

Функции работы с бинарными деревьями поиска. Определите тип данных, представляющий бинарные деревья поиска. В отличие от деревьев, представленных в методических указаниях, в деревьях поиска данные могут находиться не только в листьях, но и в промежуточных узлах дерева. Будем использовать деревья для представления ассоциативного массива, сопоставляющие значения ключей (представляемых как строки) целым числом. Для каждого узла с некоторым ключом в левом поддереве должны содержаться элементы с меньшими значениями ключа, а в правом — с большими. При поиске соответствия между строкой и числом необходимо учитывать эту информацию, поскольку она позволяет более эффективно извлекать информацию из дерева. Определите описанный тип данных и следующие функции:

- 1) `add`, добавляющую в дерево заданную пару ключа и значения.
- 2) `find`, возвращающую число, соответствующее заданной строке.
- 3) `exists`, проверяющую, что элемент с заданным ключом содержится в дереве.
- 4) `toList`, преобразующая заданное дерево поиска в список, упорядоченный по значениям ключей.

Напишите следующие программы:

- 1) Программа, считывающая два числа и возвращающая их сумму.
- 2) Программа, распечатывающая переданные в нее аргументы командной строки.
- 3) Программа, которая принимает в командной строке имя файла и распечатывает его на экране.

Реализуйте программы, выполняющие задания вашего варианта из первой лабораторной работы. Параметры функций должны считываться с клавиатуры.

5.2. Типовые задания для проведения промежуточной аттестации

Вопросы к зачету

Концепция функционального программирования.
 Языки функционального программирования.
 Технология работы с интерпретатором языка Haskell.
 Основные типы языка Haskell.
 Составные типы (кортежи).
 Синтаксис арифметических операций.
 Списки.
 Бинарные операторы.
 Строки.
 Функции.
 Условные выражения.
 Рекурсия.
 Определение функций нескольких переменных.
 Лямбда-абстракции.
 Модули.

5.3. Перечень видов оценочных средств

Лабораторные работы

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".
1. Показатели оценивания планируемых результатов обучения 2. Рейтинг по дисциплине (Представлены в Приложении - файл ФОС_ФП)

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	ИГТУ, 2015	http://biblioclub.ru/index.php?page=book&id=438365

6.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет"

Э1	Официальный сайт ФГБОУ ВО «Тульский государственный педагогический университет им. Л.Н. Толстого»
Э2	Среда электронного обучения LMS Moodle
Э3	Math.ru [Электронный ресурс]: портал математического образования / Отделение математических наук Российской Академии Наук ; Московский центр непрерывного математического образования. - М : [б. и.], 2011
Э4	Exponenta.ru [Электронный ресурс] : образовательный математический сайт / АХОФТ. - М : [б. и.], 2000

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.	Среда выполнения Adobe Flash Player. Свободно распространяемое ПО

16.	Файловый менеджер Far manager. Свободно распространяемое ПО
17.	Редактор диаграмм, схем, блок-схем, UML-схем Dia 0.97.2. Свободно распространяемое ПО
18.	Оболочка программирования Code: Blocks 17.12. Свободно распространяемое ПО
19.	Среда программирования и набор инструментов для программирования. MinGW 0.6.3 Свободно распространяемое ПО

6.3.2 Перечень информационных справочных систем и профессиональных баз данных

1.	Web of Science Core Collection – политематическая реферативно-библиографическая и наукометрическая (библиометрическая) база данных (http://webofscience.com)
2.	Полнотекстовый архив ведущих западных научных журналов на российской платформе Национального электронно-информационного консорциума (НЭИКОН) (http://neicon.ru)
3.	Базы данных издательства Springer (https://link.springer.com)

7. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Ауд.	Назначение	Оборудование и технические средства обучения	Вид
4-302	Учебная аудитория	доска учебная, проектор, стол преподавателя, столы учебные, стул преподавателя, экран	Лек
4-301	Лекционная	доска учебная, стол преподавателя, столы учебные	Лек
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
4-304	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
4-307	Компьютерный класс	аудиоколонки, компьютеры, кондиционер, маркерная доска, столы компьютерные, столы учебные, телевизор	Лаб
4-306	Компьютерный класс	аудиоколонки для проектора и интерактивной доски, интерактивная доска, компьютеры, кондиционер, маркерная доска, проектор, столы компьютерные, столы учебные	Зачёт
4-318	Компьютерный класс	компьютеры, маркерная доска, серверная стойка лаборатории МТС, стол преподавателя, столы компьютерные, столы учебный большой	Лаб

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

Приступая к изучению новой учебной дисциплины, студенты должны ознакомиться с учебной программой, учебной, научной и методической литературой, имеющейся в библиотеке университета, встретиться с преподавателем, ведущим дисциплину, получить в библиотеке рекомендованные учебники и учебно-методические пособия, осуществить запись на соответствующий курс в среде электронного обучения университета.

Глубина усвоения дисциплины зависит от активной и систематической работы студента на лекциях и практических занятиях, а также в ходе самостоятельной работы, по изучению рекомендованной литературы.

На лекциях важно сосредоточить внимание на ее содержании. Это поможет лучше воспринимать учебный материал и уяснить взаимосвязь проблем по всей дисциплине. Основное содержание лекции целесообразнее записывать в тетради в виде ключевых фраз, понятий, тезисов, обобщений, схем, опорных выводов. Необходимо обращать внимание на термины, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации. Желательно оставлять в конспектах поля, на которых делать пометки из рекомендованной литературы, дополняющей материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. С целью уяснения теоретических положений, разрешения спорных ситуаций необходимо задавать преподавателю уточняющие вопросы. Для закрепления содержания лекции в памяти, необходимо во время самостоятельной работы внимательно прочесть свой конспект и дополнить его записями из учебников и рекомендованной литературы. Конспектирование читаемых лекций и их последующая доработка способствует более глубокому усвоению знаний, и поэтому являющаяся важной формой учебной деятельности студентов.

Прочное усвоение и долговременное закрепление учебного материала невозможно без продуманной самостоятельной работы. Такая работа требует от студента значительных усилий, творчества и высокой организованности. В ходе самостоятельной работы студенты выполняют следующие задачи: дорабатывают лекции, изучают рекомендованную литературу, готовятся к практическим занятиям, к коллоквиуму, контрольным работам по отдельным темам дисциплины. При этом эффективность учебной деятельности студента во многом зависит от того, как он распорядился выделенным для

самостоятельной работы бюджетом времени.

Результатом самостоятельной работы является прочное усвоение материалов по предмету согласно программы дисциплины. В итоге этой работы формируются профессиональные умения и компетенции, развивается творческий подход к решению возникших в ходе учебной деятельности проблемных задач, появляется самостоятельности мышления. Целью практических занятий по данной дисциплине является закрепление теоретических знаний, полученных при изучении дисциплины.

При подготовке к практическому занятию целесообразно выполнить следующие рекомендации: изучить основную литературу; ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях: журналах, газетах и т. д.; при необходимости доработать конспект лекций. При этом учесть рекомендации преподавателя и требования учебной программы.

При выполнении практических занятий основным методом обучения является самостоятельная работа студента под управлением преподавателя. На них пополняются теоретические знания студентов, их умение творчески мыслить, анализировать, обобщать изученный материал, проверяется отношение студентов к будущей профессиональной деятельности.

Оценка выполненной работы осуществляется преподавателем комплексно: по результатам выполнения лабораторных заданий и заданий для самостоятельного выполнения. После подведения итогов занятия студент обязан устранить недостатки, отмеченные преподавателем при оценке его работы.