

Министерство образования и науки Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Чувашский государственный университет имени И. Н. Ульянова»

Факультет информатики и вычислительной техники

Кафедра компьютерных технологий

«УТВЕРЖДАЮ»  
Проректор по учебной работе

И.Е. Поверинов

«31» августа 2017 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ  
**«ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ»**

Направление подготовки (специальность) 09.03.04 Программная инженерия

Квалификация (степень) выпускника – бакалавр

Профиль (направленность) Управление разработкой программных проектов

Прикладной бакалавриат

Рабочая программа основана на требованиях Федерального государственного образовательного стандарта высшего образования по направлению подготовки 09.03.04 «Программная инженерия», утвержденного приказом Министерства образования и науки Российской Федерации № 229 от 12.03.2015 г.

*СОСТАВИТЕЛЬ (СОСТАВИТЕЛИ):*

доцент, к.т.н.

 П.В. Желтов

старший преподаватель

 Е.А. Мытникова

*ОБСУЖДЕНО:*

на заседании кафедры компьютерных технологий «30» 08 2017г., протокол № 1

заведующий кафедрой

 Т.А. Лавина


*СОГЛАСОВАНО:*

Методическая комиссия факультета информатики и вычислительной техники  
«30» 08 2017г., протокол № 1

Декан факультета

 А.В. Цыпцова

Директор научной библиотеки

 Н. Д. Никитина

Начальник управления информатизации

 И. П. Пивоваров

Начальник учебно-методического управления

 В. И. Маколов

## Оглавление

<b>1. Цель и задачи обучения по дисциплине</b> .....	4
<b>2. Место дисциплины в структуре основной образовательной программы (ООП)</b> .....	4
<b>3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП</b> .....	4
<b>4. Структура и содержание дисциплины</b> .....	5
4.1. Содержание дисциплины.....	5
4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения .....	7
<b>5. Содержание разделов дисциплины</b> .....	7
5.1. Лекции .....	7
5.2. Лабораторные работы .....	10
5.3. Вопросы для самостоятельной работы студента в соответствии с содержанием разделов дисциплины .....	11
<b>6. Образовательные технологии</b> .....	11
<b>7. Формы аттестации и оценочные материалы</b> .....	12
7.1. Вопросы и задачи к зачету.....	12
7.2. Вопросы и задачи к экзамену .....	12
7.3. Выполнение и примерная тематика курсового проекта.....	15
7.4. Выполнение и примерные задания расчетно-графической работы .....	18
7.5. Выполнение и примерная тематика контрольной работы .....	18
<b>8. Учебно-методическое и информационное обеспечение дисциплины</b> .....	18
8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе) .....	18
8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе).....	18
8.3. Рекомендуемые методические разработки по дисциплине .....	18
8.4. Программное обеспечение, профессиональные базы данных, информационно-справочные системы. ....	19
8.5. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы .....	19
<b>9. Материально-техническое обеспечение дисциплины</b> .....	19
<b>10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями</b> .....	20
<b>11. Методические рекомендации по освоению дисциплины</b> .....	20

## 1. Цель и задачи обучения по дисциплине

Целью изучения дисциплины является обучение студентов методике разработки программных средств с использованием технологии объектно-ориентированного программирования (ООП).

Задачи изучения дисциплины:

- моделирование прикладных и информационных процессов, описание реализации информационного обеспечения прикладных задач;
- проектирование информационных систем в соответствии со спецификой профиля подготовки по видам обеспечения (программное, информационное, организационное, техническое);
- программирование приложений, создание прототипа информационной системы, документирование проектов информационной системы на стадиях жизненного цикла, использование функциональных и технологических стандартов;
- участие в техническом и рабочем проектировании компонентов информационных систем в соответствии со спецификой профиля подготовки;
- программирование в ходе разработки информационной системы.

## 2. Место дисциплины в структуре основной образовательной программы (ООП)

Блок дисциплин – Б1.Б Дисциплины (модули). Базовая часть

Изучение данной дисциплины базируется на курсах «Информатика и программирование», «Профессиональная практика в программной инженерии», «Учебная практика (практика по получению первичных профессиональных умений и навыков, в том числе первичных умений и навыков научно-исследовательской деятельности)».

Последующие дисциплины и практики, для успешного изучения которых необходима данная дисциплина: «Введение в профессиональную деятельность программиста», «Проектный практикум», «Программная инженерия», «Объектно-ориентированное программирование», «Архитектура ЭВМ», «Проектирование и конструирование программного обеспечения», «Процессы программной инженерии», «Языки программирования систем искусственного интеллекта», «Производственная практика (практика по получению профессиональных умений и опыта профессиональной деятельности)», «Преддипломная практика для выполнения выпускной квалификационной работы».

## 3. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ООП

Процесс обучения по дисциплине направлен на формирование следующих компетенций:

общепрофессиональных (ОПК):

- готовностью применять основы информатики и программирования к проектированию, конструированию и тестированию программных продуктов (ОПК-3);

В результате обучения по дисциплине, обучающийся должен (ЗУН):

знать:

- 31 - концепции объектно-ориентированного программирования;
- 32 – основные понятия объектно-ориентированного программирования;
- 33 – способы составления объектно-ориентированных программ на языке программирования C++;
- 34 - возможности интегрированных сред программирования на языке C++;

уметь:

- У1 – описывать классы;
- У2 – анализировать различные модели обработки исключительных ситуаций;
- У3 - пользоваться средствами генерации прикладных программ на C++;

владеть навыками:

- Н1 - объектно-ориентированного анализа;

- Н2 – проектирования и программирования объектно-ориентированных программ на языке С++;
- Н3 - отладки объектно-ориентированных программ на языке С++.

#### 4. Структура и содержание дисциплины

Образовательная деятельность по дисциплине проводится:

- в форме контактной работы обучающихся с педагогическими работниками организации и (или) лицами, привлекаемыми организацией к реализации образовательных программ на иных условиях (далее – контактная работа);
- в форме самостоятельной работы.

Контактная работа включает в себя занятия лекционного типа, занятия семинарского типа (лабораторные работы), групповые и (или) индивидуальные консультации, в том числе в электронной информационно-образовательной среде.

Обозначения:

Л – лекции, л/р – лабораторные работы, п/р – практические занятия, КСР – контроль самостоятельной работы, СРС – самостоятельная работа студента, ИФР – интерактивная форма работы, К – контроль.

##### 4.1. Содержание дисциплины

Содержание	Формируемые компетенции	Формируемые ЗУН
Раздел 1. Основные понятия дисциплины ООП 1.1. Введение в ООП 1.2. Основные понятия дисциплины ООП 1.3. Принципы объектно-ориентированного анализа 1.4. Описание классов и их взаимосвязей	ОПК-3	<b>знать:</b> <ul style="list-style-type: none"> <li>– 31 - концепции объектно-ориентированного программирования;</li> <li>– 32 – основные понятия объектно-ориентированного программирования;</li> </ul> <b>уметь:</b> <ul style="list-style-type: none"> <li>– У1 – описывать классы;</li> </ul> <b>владеть навыками:</b> <ul style="list-style-type: none"> <li>– Н1 - объектно-ориентированного анализа;</li> </ul>
Раздел 2. Разработка объектно-ориентированных программ на С++ 2.1. Инкапсуляция 2.2. Наследование 2.3. Множественное наследование 2.4. Иерархия классов 2.5. Виртуальные функции 2.6. Шаблоны функций 2.7. Шаблоны классов 2.8. Анализ различных моделей обработки исключительных ситуаций. 2.9. Система классов для описания исключительных ситуаций.	ОПК-3	<b>знать:</b> <ul style="list-style-type: none"> <li>– 33 – способы составления объектно-ориентированных программ на языке программирования С++;</li> </ul> <b>уметь:</b> <ul style="list-style-type: none"> <li>– У2 – анализировать различные модели обработки исключительных ситуаций;</li> </ul> <b>владеть навыками:</b> <ul style="list-style-type: none"> <li>– Н2 – проектирования и программирования объектно-ориентированных программ на языке С++;</li> </ul>
Раздел 3. Инструментальные программные средства для С++ 3.1. Библиотеки классов для С++ 3.2. Средства генерации прикладных программ на С++	ОПК-3	<b>знать:</b> <ul style="list-style-type: none"> <li>– 34 - возможности интегрированных сред программирования на языке С++;</li> </ul> <b>уметь:</b> <ul style="list-style-type: none"> <li>– У3 - пользоваться средствами генерации прикладных программ на С++;</li> </ul> <b>владеть навыками:</b> <ul style="list-style-type: none"> <li>– Н3 - отладки объектно-</li> </ul>

		ориентированных программ на языке C++.
Курсовой проект	ОПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 31 - концепции объектно-ориентированного программирования;</li> <li>- 32 – основные понятия объектно-ориентированного программирования;</li> <li>- 33 – способы составления объектно-ориентированных программ на языке программирования C++;</li> <li>- 34 - возможности интегрированных сред программирования на языке C++;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У1 – описывать классы;</li> <li>- У2 – анализировать различные модели обработки исключительных ситуаций;</li> <li>- У3 - пользоваться средствами генерации прикладных программ на C++;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н1 - объектно-ориентированного анализа;</li> <li>- Н2 – проектирования и программирования объектно-ориентированных программ на языке C++;</li> <li>- Н3 - отладки объектно-ориентированных программ на языке C++.</li> </ul>
Экзамен	ОПК-3	<p><b>знать:</b></p> <ul style="list-style-type: none"> <li>- 31 - концепции объектно-ориентированного программирования;</li> <li>- 32 – основные понятия объектно-ориентированного программирования;</li> <li>- 33 – способы составления объектно-ориентированных программ на языке программирования C++;</li> <li>- 34 - возможности интегрированных сред программирования на языке C++;</li> </ul> <p><b>уметь:</b></p> <ul style="list-style-type: none"> <li>- У1 – описывать классы;</li> <li>- У2 – анализировать различные модели обработки исключительных ситуаций;</li> <li>- У3 - пользоваться средствами генерации прикладных программ на C++;</li> </ul> <p><b>владеть навыками:</b></p> <ul style="list-style-type: none"> <li>- Н1 - объектно-ориентированного анализа;</li> <li>- Н2 – проектирования и программирования объектно-ориентированных программ на языке C++;</li> </ul>

		- НЗ - отладки объектно-ориентированных программ на языке С++.
--	--	--

#### 4.2. Объем дисциплины, виды учебной работы обучающихся по очной форме обучения

Содержание	Всего, час	Контактная работа, час				СРС, час	ИФР, час	К, час
		Л	л/р	п/р	КСР			
Раздел 1. Основные понятия дисциплины ООП								
1.1. Введение в ООП	5	2	3				3	
1.2. Основные понятия дисциплины ООП	5	2	3				3	
1.3. Принципы объектно-ориентированного анализа	7	2	3			2	3	
1.4. Описание классов и их взаимосвязей	7	2	3			2	3	
Раздел 2. Разработка объектно-ориентированных программ на С++								
2.1. Инкапсуляция	5	2	3				3	
2.2. Наследование	5	2	3				3	
2.3. Множественное наследование	5	2	3				3	
2.4. Иерархия классов	12	4	6			2	6	
2.5. Виртуальные функции	7	2	3			2	3	
2.6. Шаблоны функций	5	2	3				3	
2.7. Шаблоны классов	7	2	3			2	3	
2.8. Анализ различных моделей обработки исключительных ситуаций.	5	2	3				3	
2.9. Система классов для описания исключительных ситуаций.	7	2	3			2	3	
Раздел 3. Инструментальные программные средства для С++								
3.1. Библиотеки классов для С++	7	2	3			2	3	
3.2. Средства генерации прикладных программ на С++	7	2	3			2	3	
Курсовой проект	12				2	10		
Экзамен	36							36
<b>Итого</b>	<b>144</b>	<b>32</b>	<b>48</b>		<b>2</b>	<b>26</b>	<b>48</b>	<b>36</b>
<b>Зачетных единиц</b>	<b>4</b>							

Вид промежуточной аттестации: курсовой проект в 3 семестре, экзамен в 3 семестре.

### 5. Содержание разделов дисциплины

#### 5.1. Лекции

##### Раздел 1. Основные понятия дисциплины ООП

##### Тема 1.1. Введение в ООП.

1. Предмет дисциплины, ее структура и содержание.
2. Цели и содержание дисциплины для подготовки студентов по выбранному направлению.
3. Структура и план учебной деятельности студентов, рекомендуемая литература.
4. Связь дисциплины с предшествующими и последующими дисциплинами.
5. Краткие сведения об истории развития и становления проблематики дисциплины.
6. Стандарты, используемые в ООП.

##### Тема 1.2. Основные понятия дисциплины ООП.

1. Семантика и интерпретация языка программирования.

2. Основные положения объектного подхода к разработке программ.
3. Принципы объектного подхода, абстрагирование, ограничение доступа, модульность, иерархия, типизация параллелизм, устойчивость.
4. Понятие объекта и класса.
5. Инкапсуляция и спецификация правил доступности элементов класса.
6. Конструкторы и деструкторы.
7. Наследование.
8. Иерархия классов.
9. Одиночное и множественное наследование.
10. Способы реализации множественного наследования, их достоинства и недостатки.
11. Отношения между объектами и/или классами.
12. Полиморфизм.
13. Ранее и позднее связывание.
14. Управление последовательностью действий в объектно-ориентированной программе.
15. Объект и процесс.
16. Инициализация и взаимодействие объектов и процессов.
17. Сообщения.
18. Реализация механизмов посылки сообщений.
19. Примеры функционирования объектно-ориентированной программы.
20. Формальные модели объектов и классов: автоматная и алгебраическая модели объектов, исчисления типов.

*Тема 1.3. Принципы объектно-ориентированного анализа.*

1. Системный анализ.
2. Принципы объектно-ориентированного анализа и их обсуждение.
3. Основные определения: система, домен, подсистема, элемент, связи, среда.
4. Структура системы, декомпозиция, иерархия элементов.
5. Процессы в системе и потоки информации.
6. Исследование действий.
7. Построение моделей доменов и подсистем, связей и взаимодействия подсистем, взаимодействия объектов, событий, процессов, потоков данных, действий.

*Тема 1.4. Описание классов и их взаимосвязей.*

1. Описание классов и их взаимосвязей.
2. Динамика поведения объектов, диаграммы перехода состояний.
3. Диаграммы объектов.
4. Видимость и синхронизация объектов, временные диаграммы.
5. Диаграмма процессов.
6. Обработка исключительных ситуаций.

Раздел 2. Разработка объектно-ориентированных программ на C++

*Тема 2.1. Инкапсуляция.*

1. Способы реализации инкапсуляции.
2. Определение класса.
3. Личная и общая части определения класса.
4. Функции-элементы класса и функции-друзья.
5. Объекты класса.
6. Создание и уничтожение объектов класса.
7. Конструкторы и деструкторы.

*Тема 2.2. Наследование*

1. Базовый и производный классы.
2. Функции-элементы и функции-друзья.
3. Правила доступа к элементам производного класса.
4. Конструкторы и деструкторы.



## 5. Простое наследование

*Тема 2.3. Множественное наследование*

1. Базовый и производные классы.
2. Правила доступа к элементам производных класса.
3. Одиночное и множественное наследование.

*Тема 2.4. Иерархия классов.*

1. Иерархия классов.

*Тема 2.5. Виртуальные функции.*

1. Виртуальные базовые классы.
2. Особенности доступа при множественном наследовании.
3. Правила преобразования указателей.
4. Виртуальные функции.
5. Таблицы виртуальных функций.
6. Распределение таблиц виртуальных функций в многофайловых программах.
7. Чистые виртуальные функции и абстрактные базовые классы.
8. Полиморфные контейнерные классы, итераторы и аппликаторы.
9. Абстрактные и конкретные контейнерные классы.
10. Виды классов: конкретный тип, абстрактный тип, узловой класс, интерфейсный класс и другие.
11. Инициализация объектов.
12. Отличия инициализации от присваивания.
13. Инициализация баз и членов.
14. Полный объект конечного производного класса.
15. Примеры описания и использования классов с наследованием.

*Тема 2.6. Шаблоны функций.*

1. Шаблоны классов и функций.
2. Наследование шаблонных классов.
3. Правила отождествления параметров шаблона.

*Тема 2.7. Шаблоны классов.*

1. Применение шаблонных классов для создания контейнерных классов.
2. Анализ вариантов шаблонных классов на примере шаблона вектора с операцией сортировки.

*Тема 2.8. Анализ различных моделей обработки исключительных ситуаций.*

1. Анализ различных моделей обработки исключительных ситуаций.
2. Стандартные средства контроля подтверждений.
3. Проверка предусловий и постусловий, вычисления инвариантов.
4. Контроль асинхронных событий.
5. Реализация модели обработки синхронных ситуаций с завершением в C++.
6. Возбуждение ситуации.
7. Описание блоков с контролем и реакций на ситуации.

*Тема 2.9. Система классов для описания исключительных ситуаций.*

## Лекция 12. Система классов для описания исключительных ситуаций.

1. Система классов для описания исключительных ситуаций.
2. Обработка исключительных ситуаций при обработке исключительных ситуаций.
3. Примеры контроля ситуаций в контейнерных классах.

## Раздел 3. Инструментальные программные средства для C++

*Тема 3.1. Библиотеки классов для C++.*

1. Библиотеки классов для C++.

*Тема 3.2. Средства генерации прикладных программ на C++.*

1. Средства генерации прикладных программ на C++.

## 5.2. Лабораторные работы

Лабораторная работа №1. Создание простого класса. Определение методов класса в классе и вне класса.

Лабораторная работа № 2. Создать класс с указанными полями. Класс имеет конструктор. Функции-члены установки полей данных, функции-члены получения полей данных, а также две функции-члены печати по заданному шаблону. Функции-члены установки полей класса должны проверять корректность задаваемых параметров.

Лабораторная работа № 3. Создание класса с конструкторами. Использование общих и частных элементов класса. Использование оператора глобального разрешения для элементов класса.

Лабораторная работа № 4. Построить диаграмму заданного класса.

Лабораторная работа № 5. Создать класс. Определить указанные операторы, как функции-члены или как дружественные функции.

Лабораторная работа №6. Создать класс, используя иерархию или множественное наследование.

Лабораторная работа №7. Создать класс имеющий имя (указатель на строку) и два целочисленных параметра. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс, имеющий имя. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения имен.

Лабораторная работа №8. Создать класс - данные - абстрактный базовый класс. Создать производные классы - данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Лабораторная работа №9. Написать функцию-шаблон согласно своему варианту.

Лабораторная работа №10. Написать шаблон класса согласно своему варианту.

Лабораторная работа №11. Создать класс – тип ошибок памяти, работы с файлами и т.д. Перехватить исключительную ситуацию типа, созданного Вами.

Лабораторная работа №12. Создать параметризованный класс согласно своему варианту.

Лабораторная работа №13. Создать абстрактный тип данных - класс вектор, который имеет указатель, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, - размер вектора, - выделяет место и инициализирует номер в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного типа. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Лабораторная работа №14. Спроектировать класс согласно своему варианту.

Лабораторная работа №15. Создать свою библиотеку согласно своему варианту.

Лабораторная работа №16. Создать иерархию классов. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

5.3. Вопросы для самостоятельной работы студента в соответствии с содержанием разделов дисциплины

1. Основные положения объектного подхода к разработке программ.
2. Ранее и позднее связывание.
3. Объект и процесс.
4. Сообщения.
5. Реализация механизмов посылки сообщений.
6. Формальные модели объектов и классов: автоматная и алгебраическая модели объектов, исчисления типов.
7. Системный анализ.
8. Видимость и синхронизация объектов, временные диаграммы.
9. Вложенные и локальные классы.
10. Распределение таблиц виртуальных функций в многофайловых программах.
11. Полиморфные контейнерные классы, итераторы и аппликаторы.
12. Абстрактные и конкретные контейнерные классы.
13. Виды классов: конкретный тип, абстрактный тип, узловой класс, интерфейсный класс и другие.
14. Проверка предусловий и постусловий, вычисления инвариантов.
15. Контроль асинхронных событий.
16. Обработка исключительных ситуаций при обработке исключительных ситуаций.
17. Примеры контроля ситуаций в контейнерных классах.
18. Механизмы функционирования программ.
19. Стиль программирования.
20. Надежность программ на C++.
21. Средства генерации прикладных программ на C++.

## 6. Образовательные технологии

В соответствии со структурой образовательного процесса по дисциплине применяются следующие технологии:

- диагностики;
- целеполагания;
- управления процессом освоения учебной информации;
- применения знаний на практике, поиска новой учебной информации;
- организации совместной и самостоятельной деятельности обучающихся (учебно-познавательной, научно-исследовательской, частично-поисковой, репродуктивной, творческой и пр.);

В соответствии с требованиями ФГОС ВО для реализации компетентного подхода при обучении дисциплине предусмотрено широкое использование в учебном процессе активных и интерактивных методов проведения занятий:

Интерактивные технологии:

№ темы	Вид занятия	Используемые интерактивные технологии
1-15	Лабораторная работа	Метод проектов

При обучении дисциплине применяются следующие формы занятий:

- лекции, направленные на получение новых и углубление научно-теоретических знаний, в том числе вводная лекция, информационная лекция, обзорная лекция, лекция-консультация, проблемная лекция, лекции-дискуссии, лекции-беседы и др.;
- лабораторные занятия, проводимые под руководством преподавателя в учебной лаборатории с использованием компьютеров и учебного оборудования, направленные на закрепление и получение новых умений и навыков, применение знаний и умений, полученных на теоретических занятиях, при решении практических задач и др.

Все занятия обеспечены мультимедийными средствами для повышения качества восприятия изучаемого материала. В образовательном процессе широко используются информационно-коммуникационные технологии.

Самостоятельная работа студентов – это планируемая работа студентов, выполняемая по заданию при методическом руководстве преподавателя, но без его непосредственного участия. Формы самостоятельной работы студентов определяются содержанием учебной дисциплины, степенью подготовленности студентов. Они могут иметь учебный или учебно-исследовательский характер: подготовка к лабораторным работам, разработка проекта.

Формами контроля самостоятельной работы выступают: проверка письменных отчётов по результатам выполненных лабораторных работ; проверка курсового проекта. Результаты самостоятельной работы учитываются при оценке знаний на экзамене.

## **7. Формы аттестации и оценочные материалы**

Текущий контроль предназначен для проверки хода и качества усвоения учебного материала, стимулирования учебной работы обучающихся и совершенствования методики проведения занятий. Он проводится в ходе всех видов занятий в форме, избранной преподавателем.

Промежуточная аттестация имеет целью определить степень достижения учебных целей по учебной дисциплине и проводится в форме экзамена, защиты курсового проекта. Принимается экзамен преподавателем, читающим лекции по данной учебной дисциплине в соответствии с перечнем основных вопросов, выносимых для контроля знаний обучающихся:

7.1. Вопросы и задачи к зачету

Зачет не предусмотрен.

7.2. Вопросы и задачи к экзамену

*Вопросы:*

1. Стандарты, используемые в ООП.
2. Семантика и интерпретация языка программирования.
3. Принципы объектного подхода, абстрагирование, ограничение доступа, модульность, иерархия, типизация, параллелизм, устойчивость.
4. Понятие объекта и класса.
5. Инкапсуляция и спецификация правил доступности элементов класса.
6. Наследование.
7. Одиночное и множественное наследование.
8. Способы реализации множественного наследования, их достоинства и недостатки.
9. Отношения между объектами и/или классами.
10. Полиморфизм.
11. Управление последовательностью действий в объектно-ориентированной программе.
12. Инициализация и взаимодействие объектов и процессов.
13. Примеры функционирования объектно-ориентированной программы.
14. Принципы объектно-ориентированного анализа и их обсуждение.
15. Основные определения: система, домен, подсистема, элемент, связи, среда.
16. Структура системы, декомпозиция, иерархия элементов.
17. Процессы в системе и потоки информации.
18. Исследование действий.
19. Построение моделей доменов и подсистем, связей и взаимодействия подсистем, взаимодействия объектов, событий, процессов, потоков данных, действий.
20. Описание классов и их взаимосвязей.
21. Динамика поведения объектов, диаграммы перехода состояний.
22. Диаграммы объектов.

23. Диаграмма процессов.
24. Обработка исключительных ситуаций.
25. Способы реализации инкапсуляции.
26. Определение класса.
27. Личная и общая части определения класса.
28. Функции-элементы класса и функции-друзья.
29. Объекты класса.
30. Создание и уничтожение объектов класса.
31. Конструкторы и деструкторы.
32. Интерфейс и реализация контейнерных классов для моделирования структур данных.
33. Статические члены объектов класса.
34. Примеры описания и использования классов.
35. Правила доступа к элементам производного класса.
36. Конструкторы и деструкторы.
37. Иерархия классов.
38. Виртуальные базовые классы.
39. Особенности доступа при множественном наследовании.
40. Правила преобразования указателей.
41. Виртуальные функции.
42. Таблицы виртуальных функций.
43. Чистые виртуальные функции и абстрактные базовые классы.
44. Отличия инициализации от присваивания.
45. Инициализация баз и членов.
46. Полный объект конечного производного класса.
47. Примеры описания и использования классов с наследованием.
48. Шаблоны классов
49. Шаблоны функций.
50. Наследование шаблонных классов.
51. Правила отождествления параметров шаблона.
52. Применение шаблонных классов для создания контейнерных классов.
53. Анализ вариантов шаблонных классов на примере шаблона вектора с операцией сортировки.
54. Анализ различных моделей обработки исключительных ситуаций.
55. Стандартные средства контроля подтверждений.
56. Реализация модели обработки синхронных ситуаций с завершением в C++.
57. Возбуждение ситуации.
58. Описание блоков с контролем и реакций на ситуации.
59. Система классов для описания исключительных ситуаций.
60. Особенности разработки объектно-ориентированных программ на C++.
61. Проектирование классов.
62. Библиотеки классов для C++.
63. Библиотека ввода-вывода.
64. Строки в стиле C++.

*Темы задач:*

*Определение классов. Наследование. Виртуальные функции*

1. Создать класс «Треугольник», заданный координатами вершин. Определить производные классы «Равносторонний» и «Прямоугольный» треугольники. Предусмотреть виртуальную функцию для вычисления площади описанной около треугольника окружности, сравнить площади окружностей, описанных около равностороннего и прямоугольного треугольников, координаты которых введены с клавиатуры.

2. Создать класс «Окружность», заданный координатами центра и точкой на окружности. Определить производный класс «Описанная окружность». Предусмотреть виртуальную

функцию для вычисления периметра равностороннего треугольника, вписанного в окружность. Сравнить периметры 3 равносторонних треугольников, вписанных в окружности, данные о которых вводятся с клавиатуры.

3. Создать класс «Окружность», заданный координатами центра и точкой на окружности. Определить производный класс «Описанная окружность». Предусмотреть виртуальную функцию для вычисления площади равностороннего треугольника, вписанного в окружность. Сравнить площади 2 равносторонних треугольников, вписанных в окружности, данные о которых вводятся с клавиатуры.

4. Создать класс «Шар», заданный координатами центра и точкой на сфере. Определить производный класс «Окружность». Предусмотреть виртуальную функцию для вычисления площади круга, сравнить площади 2 кругов, данные о которых введены с клавиатуры.

5. Создать класс «Шар», заданный радиусом. Определить производный класс «Окружность». Предусмотреть виртуальную функцию для вычисления площади круга и длины окружности, сравнить площади и длины 2 окружностей, радиусы которых введены с клавиатуры.

*Потоковый ввод-вывод. Текстовые и бинарные потоки. Перегрузка ввода-вывода*

1. В одном файле хранятся сведения о книгах: код, автор, название, во 2 файле сведения об издании: код, издательство, год издания. Вывести на экран информацию о книгах автора, введенного с клавиатуры, изданных в 2013 году, введенного с клавиатуры (ФИО, должность, зарплата за год), выбрать сотрудников, получивших зарплату за год меньше средней по отделу.

2. В одном файле хранятся сведения о сотрудниках: ФИО, табельный номер, должность, отдел, во 2 файле сведения о работе за год: табельный номер, заработная плата по месяцам. Вывести на экран информацию о сотрудниках отдела, введенного с клавиатуры (ФИО, должность, зарплата за год), выбрать сотрудников, получивших зарплату за год меньше средней по отделу.

3. В одном файле хранятся сведения о товарах в магазине: код товара, название, цена, во 2 файле сведения о производстве: код товара, производитель, дата производства, срок годности. Вывести на экран информацию о товарах произведенных после 20.06.2012 (название, цена, производитель, дата производства), которые стоят менее 300 рублей, у которых истек срок годности.

4. В одном файле хранятся сведения о сотрудниках: ФИО, табельный номер, должность, отдел, во 2 файле сведения о работе за год: табельный номер, заработная плата по месяцам. Вывести на экран информацию о сотрудниках отдела, введенного с клавиатуры (ФИО, должность, зарплата за год), отсортированную по фамилиям в алфавитном порядке.

5. В одном файле хранятся сведения об учениках: ФИО, класс, во 2 файле сведения о результатах сдачи экзаменов: ФИО, название предмета, ФИО преподавателя, оценка. Вывести на экран фамилии, класс и оценки учеников, сдавших математику, но не сдавших русский язык.

Билет для проведения промежуточной аттестации в форме экзамена включает вопросы и задачи для проверки сформированности знаний, умений и навыков.

Общими критериями, определяющими оценку знаний, умений и навыков являются:

– для оценки «отлично» - наличие глубоких и исчерпывающих знаний в объеме пройденного программного материала правильные и уверенные действия по применению полученных знаний на практике, грамотное и логически стройное изложение материала при ответе, знание дополнительно рекомендованной литературы;

– для оценки «хорошо» - наличие твердых и достаточно полных знаний программного материала, незначительные ошибки при освещении заданных вопросов, правильны действия по применению знаний на практике, четкое изложение материала;

– для оценки «удовлетворительно» - наличие твердых знаний пройденного материала, изложение ответов с ошибками, уверенно исправляемыми после дополнительных вопросов, необходимость наводящих вопросов, правильные действия по применению знаний на практике;

– для оценки «неудовлетворительно» - наличие грубых ошибок в ответе, непонимание сущности излагаемого вопроса, неумение применять знания на практике, неуверенность и неточность ответов на дополнительные и наводящие вопросы.

### 7.3. Выполнение и примерная тематика курсового проекта

Курсовой проект выполняется в процессе изучения дисциплины. Общее руководство и контроль за ходом выполнения курсового проекта осуществляет преподаватель соответствующей дисциплины. Курсовой проект выполняется в соответствии с методическими указаниями для обучающихся.

Основными функциями руководителя курсового проекта являются:

- определение и формулирование темы курсового проекта совместно с обучающимся на основе примерной тематики или по предложенной обучающимся теме в рамках содержания дисциплины;
- консультирование по вопросам содержания и последовательности выполнения курсового проекта;
- оказание помощи студенту в подборе необходимой литературы;
- контроль хода выполнения курсового проекта.

Примерная тематика курсового проекта

Комплект тем №1:

Возможные варианты абстрактных типов данных:

Тема 1. Разработка класса для представления множества целых чисел на основе связанного списка. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 2. Разработка класса для представления многочлена от одной переменной на основе связанного списка. Операции: сложение многочленов, умножение многочленов, деление многочленов, дифференцирование, интегрирование.

Тема 3. Разработка класса для представления упорядоченного множества вещественных чисел на основе циклического связанного списка. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 4. Разработка класса для представления многочлена от одной переменной на основе двунаправленного связанного списка. Операции: сложение многочленов, умножение многочленов, деление многочленов, дифференцирование, интегрирование.

Тема 5. Разработка класса для представления упорядоченного множества строк на основе бинарного дерева. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 6. Разработка класса для представления многочлена от одной переменной на основе циклического связанного списка. Операции: сложение многочленов, умножение многочленов, деление многочленов, дифференцирование, интегрирование.

Тема 7. Разработка класса для представления множества целых чисел на основе хеширования со связанными цепочками. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 8. Разработка класса для представления многочлена от двух переменных на основе связанного списка. Операции: сложение многочленов, умножение многочленов.

Тема 9. Разработка класса для представления множества целых чисел на основе хеширования в таблице. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 10. Разработка класса для представления множества вещественных чисел на основе двунаправленного связанного списка. Операции – включение элемента, исключение элемента, поиск элемента, объединение множеств, пересечение множеств.

Тема 11. Создать шаблон класса массив с методом сортировки, отсортировать свой класс “Адрес” по его полям город, улица, индекс.

Тема 12. Создать шаблон циклической очереди. С помощью него обработать ввод с клавиатуры, заполнение информацией вашей памяти.

Тема 13. Создать шаблон приоритетной очереди. При добавлении элемента в такую очередь порядковый номер нового элемента определяется его приоритетом..

Тема 14. С помощью шаблона класса стек написать программу калькулятор с операциями сложения, вычитания, деления, умножения, возведения в степень.

Тема 15. Создать шаблон связный список с двойными ссылками, в котором просмотр списка в любом направлении, поиск в списке конкретного элемента, удаление элемента из списка, ввод нового элемента в список, получение указателя на начало и конец списка, ввод нового элемента в список.

Тема 16. Создать шаблон бинарное дерево, в котором можно проходить по дереву одним из трех способов: последовательно, нисходящим и восходящим. Добавлять элементы в дерево, поиск по дереву, удалять элементы из дерева.

Базовый класс будет определять природу объектов, формирующих дерево, а от него наследовать класс, определяющий операции на деревом.

Тема 17. Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования T-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

Тема 18. Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также последовательного доступа ко всем элементам.

Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

Тема 19. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол.

Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

Тема 20. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

#### Комплект тем №2:

Изучение возможностей работы с текстовыми и двоичными файлами. Использование стандартных потоковых классов.

Во всех заданиях составить две программы. Первая программа должна формировать двоичный файл. Вторая программа должна считать данные из двоичного файла, выполнить соответствующие вычисления и записать их результаты в текстовый файл.

Варианты заданий:

1. Создать двоичный файл, куда записать  $n$  целых чисел. Из исходного файла сформировать массивы четных и нечетных чисел. Определить наибольший чётный компонент файла и наименьший нечётный.
2. Создать двоичный файл, куда записать  $n$  целых чисел. На основе исходного файла создать массив удвоенных нечетных чисел. Упорядочить его по возрастанию элементов.
3. Создать двоичный файл, куда записать  $n$  целых чисел. Сформировать массив положительных чисел, делящихся на пять без остатка, используя элементы исходного файла. Упорядочить массив по убыванию элементов.
4. Создать двоичный файл, куда записать  $n$  вещественных чисел. Из компонентов исходного файла сформировать массивы положительных и отрицательных простых чисел. Вычислить количество нулевых компонентов файла.



5. Создать двоичный файл, куда записать  $n$  целых чисел. Из файла создать массив, элементы которого не являются простыми числами и расположены до минимального элемента.
6. Создать двоичный файл, куда записать  $n$  целых чисел. Из файла целых чисел сформировать массив, записав в него только ненулевые компоненты, находящиеся после максимального элемента.
7. Создать двоичный файл, куда записать  $n$  вещественных чисел. Сделать массив из элементов исходного файла, внося в него числа, превосходящие среднее значение среди положительных значений файла.
8. Создать двоичный файл, куда записать  $n$  целых чисел. Из исходного файла сформировать массив, записав в него числа, расположенные в файле до минимального элемента и после максимального.
9. Создать двоичный файл, куда записать  $n$  целых чисел. Найти среднее арифметическое среди простых чисел, расположенных после минимального значения.
10. Создать двоичный файл, куда записать  $n$  целых чисел. Поменять местами первое совершенное и последнее отрицательное число в файле.
11. Создать двоичный файл, куда записать  $n$  целых чисел. Все простые числа из исходного файла записать в массив, который упорядочить по убыванию.
12. Создать двоичный файл, куда записать  $n$  целых чисел. Последнюю группу совершенных чисел из исходного файла переписать в текстовый файл.
13. Создать двоичный файл, куда записать  $n$  целых чисел. Найти в нем группу положительных элементов наибольшей длины.
14. Создать двоичный файл, куда записать  $n$  целых чисел. Из исходного файла сформировать массивы простых и совершенных чисел. Определить наибольшее простое число в файле и наименьшее совершенное.
15. Создать двоичный файл, куда записать  $n$  целых чисел. Из файла создать массив, элементы которого являются простыми числами и расположены после максимального значения файла.
16. Создать двоичный файл, куда записать  $n$  целых чисел. Из файла целых чисел сформировать массив, записав в него только кратные 3 значения, находящиеся до минимального элемента файла.
17. Создать двоичный файл, куда записать  $n$  вещественных чисел. Сделать массив из элементов исходного файла, внося в него числа, превосходящие среднее значение среди четных значений файла.
18. Создать двоичный файл, куда записать  $n$  вещественных чисел. Поменять местами последнее простое число в файле с наименьшим совершенным.
19. Создать двоичный файл, куда записать  $n$  вещественных чисел. Найти сумму последней группы отрицательных элементов.
20. Создать двоичный файл, куда записать  $n$  целых чисел. Найти произведение первой группы, состоящей из простых чисел.

Оценивание курсового проекта осуществляется в соответствии с полнотой и качеством выполнения задания на курсовой проект, качеством защиты проекта (ответы на вопросы, презентация и др.). Оценка курсового проекта отражает уровень сформированности соответствующих компетенций:

– «отлично» - проект выполнен в соответствии с утвержденным планом и заданием, полностью раскрыто содержание каждого вопроса; студентом сформулированы собственные аргументированные выводы по теме работы; оформление работы соответствует предъявляемым требованиям; при защите работы обучающийся демонстрирует свободное владение материалом и верно отвечает на поставленные вопросы;

– «хорошо» - проект выполнен в соответствии с утвержденным планом и заданием; полностью раскрыто содержание каждого вопроса; имеются незначительные замечания к оформлению работы; при защите работы обучающийся демонстрирует владение материалом, но отвечает на ряд поставленных вопросов не в достаточно полном объеме;

– «удовлетворительно» - проект выполнен в соответствии с утвержденным планом и заданием, но не полностью раскрыто содержание каждого вопроса; обучающимся не сделаны собственные выводы по теме работы; допущены существенные недостатки в оформлении работы; при защите работы обучающийся демонстрирует владение материалом, но отвечает не на все поставленные вопросы, либо не в достаточно полном объеме;

– «неудовлетворительно» - если проект не выполнен в соответствии с утвержденным планом и заданием, не раскрыто содержание каждого вопроса; обучающимся не сделаны выводы по теме работы, имеются существенные недостатки в оформлении работы; при защите работы обучающийся не демонстрирует владение материалом, не отвечает на поставленные вопросы.

В случае оценивания работы на «неудовлетворительно» работа направляется на дальнейшую доработку.

#### 7.4. Выполнение и примерные задания расчетно-графической работы

Расчетно-графическая работа не предусмотрена.

#### 7.5. Выполнение и примерная тематика контрольной работы

Контрольная работа не предусмотрена.

## 8. Учебно-методическое и информационное обеспечение дисциплины

Электронный каталог и электронные информационные ресурсы, предоставляемые научной библиотекой ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова» доступны по ссылке <http://library.chuvsu.ru/>

8.1. Рекомендуемая основная литература (ежегодное обновление перечня и условия доступа представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Васильев А.Н. Объектно-ориентированное программирование на C++ [Электронный ресурс] / А.Н. Васильев. — Электрон. текстовые данные. — СПб. : Наука и Техника, 2016. — 544 с. - Режим доступа: <a href="http://www.iprbookshop.ru/60648.html">http://www.iprbookshop.ru/60648.html</a>
2.	Программирование на языке высокого уровня C/C++ [Электронный ресурс] : конспект лекций / . — Электрон. текстовые данные. — М. : Московский государственный строительный университет, Ай Пи Эр Медиа, ЭБС АСВ, 2016. — 140 с. - Режим доступа: <a href="http://www.iprbookshop.ru/48037.html">http://www.iprbookshop.ru/48037.html</a>
3.	Стенли Липпман Язык программирования C++ [Электронный ресурс] : полное руководство / Липпман Стенли, Лажоие Жози. — Электрон. текстовые данные. — Саратов: Профобразование, 2017. — 1104 с. - Режим доступа: <a href="http://www.iprbookshop.ru/63964.html">http://www.iprbookshop.ru/63964.html</a>

8.2. Рекомендуемая дополнительная литература (ежегодное обновление и условия доступа перечня представлены в Приложениях к рабочей программе)

№ п/п	Наименование
1.	Никишев, В. К Объектно-ориентированное программирование : учебное пособие / В. К. Никишев ; [отв. ред. В. П. Желтов] ; Чуваш. гос. ун-т им. И. Н. Ульянова. - Чебоксары : Изд-во Чуваш. ун-та, 2016. - 262с.
2.	Объектно-ориентированное программирование : методические указания к курсовой работе [для 3 курса направления "Информатика и вычислительная техника"] / Чуваш. гос. ун-т им. И. Н. Ульянова ; [сост.: И. А. Обломов ; отв. ред. А. Л. Симаков]. - Чебоксары : Изд-во Чуваш. ун-та, 2015. - 16с.
3.	Обломов, И. А. Объектно-ориентированное программирование : лабораторный практикум / И. А. Обломов ; [отв. ред. А. Л. Симаков] ; Чуваш. гос. ун-т им. И. Н. Ульянова. - Чебоксары : Изд-во Чуваш. ун-та, 2014. - 111с.

8.3. Рекомендуемые методические разработки по дисциплине

№ п/п	Наименование	Условия доступа
	Программирование: лабораторный практикум	<a href="http://moodle.chuvsu.ru/course/view.php?id=1206">URL: http://moodle.chuvsu.ru/course/view.php?id=1206</a>
2.	Методические рекомендации по выполнению курсового проекта по дисциплине «Программирование на языках высокого уровня»	<a href="http://moodle.chuvsu.ru/course/view.php?id=1206">URL: http://moodle.chuvsu.ru/course/view.php?id=1206</a>

8.4. Программное обеспечение, профессиональные базы данных, информационно-справочные системы.

Доступное программное лицензионное и свободно распространяемое программное обеспечение, предоставляемое студенту университетом возможно для загрузки и использования по URL: <http://ui.chuvsu.ru/> \*.

№ п/п	Наименование Рекомендуемого ПО	Условия доступа/скачивания
		свободное лицензионное соглашение:
1.	Microsoft Visual Studio	<a href="https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017">https://www.microsoft.com/ru-ru/SoftMicrosoft/vs2017</a>
2.	DevC++	<a href="https://sourceforge.net/projects/orwelldevcpp/">https://sourceforge.net/projects/orwelldevcpp/</a>
3.	Linux/ Ubuntu	<a href="http://ubuntu.ru/">http://ubuntu.ru/</a>
4.	LibreOffice	<a href="https://ru.libreoffice.org/">https://ru.libreoffice.org/</a>
1.	Microsoft Windows	из внутренней сети университета (договор)*
2.	Microsoft Office	
1.	Гарант	из внутренней сети университета (договор)*
2.	Консультант +	

#### 8.5. Рекомендуемые интернет-ресурсы и открытые он-лайн курсы

№ п/п	Наименование интернет ресурса	Режим доступа
1.	Открытое образование. Языки программирования. Генезис и практики	URL: <a href="https://openedu.ru/course/ITMOUniversity/PROLAN/">https://openedu.ru/course/ITMOUniversity/PROLAN/</a>
2.	Национальный открытый университет. Язык программирования C++	URL: <a href="http://www.intuit.ru/studies/courses/17/17/info">http://www.intuit.ru/studies/courses/17/17/info</a>
3.	Национальный открытый университет. Основы объектно-ориентированного программирования	URL: <a href="http://www.intuit.ru/studies/courses/71/71/info">http://www.intuit.ru/studies/courses/71/71/info</a>

### 9. Материально-техническое обеспечение дисциплины

Учебные аудитории для лекционных занятий по дисциплине оснащены автоматизированным рабочим местом (АРМ) преподавателя, обеспечивающим тематические иллюстрации и демонстрации, соответствующие программе дисциплины в составе:

- ПЭВМ с доступом в интернет (операционная система, офисные программы, антивирусные программы);
- мультимедийный проектор с дистанционным управлением;
- настенный экран;

Учебные аудитории для лабораторных и самостоятельных занятий по дисциплине, курсового проектирования оснащены АРМ преподавателя и пользовательскими АРМ по числу обучающихся, объединенных локальной сетью («компьютерный» класс), с возможностью подключения к сети Интернет и доступом к электронной информационно-образовательной среде ФГБОУ ВО «Чувашский государственный университет имени И.Н. Ульянова».

## **10. Средства адаптации преподавания дисциплины к потребностям лиц с ограниченными возможностями**

В случае необходимости, обучающимся из числа лиц с ограниченными возможностями здоровья (по заявлению обучающегося) могут предлагаться одни из следующих вариантов восприятия информации с учетом их индивидуальных психофизических особенностей, в том числе с применением электронного обучения и дистанционных технологий:

- для лиц с нарушениями зрения: в печатной форме увеличенным шрифтом; в форме электронного документа; в форме аудиофайла (перевод учебных материалов в аудиоформат); в печатной форме на языке Брайля; индивидуальные консультации с привлечением тифлосурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями слуха: в печатной форме; в форме электронного документа; видеоматериалы с субтитрами; индивидуальные консультации с привлечением сурдопереводчика; индивидуальные задания и консультации.

- для лиц с нарушениями опорно-двигательного аппарата: в печатной форме; в форме электронного документа; в форме аудиофайла; индивидуальные задания и консультации.

## **11. Методические рекомендации по освоению дисциплины**

В ходе лекционных занятий студенту рекомендуется вести конспектирование учебного материала. Следует обращать внимание на категории, формулировки, раскрывающие содержание тех или иных явлений и процессов, научные выводы и практические рекомендации, положительный опыт в ораторском искусстве. При составлении конспекта желательно оставлять в рабочих конспектах поля, на которых в дальнейшем можно делать пометки из рекомендованной литературы, дополняющие материал прослушанной лекции, а также подчеркивающие особую важность тех или иных теоретических положений. В ходе лекционных занятий рекомендуется задавать преподавателю уточняющие вопросы с целью уяснения теоретических положений, разрешения спорных ситуаций.

В ходе подготовки к лабораторным работам рекомендуется изучить основную литературу, ознакомиться с дополнительной литературой, новыми публикациями в периодических изданиях. Основой для выполнения лабораторной работы являются разработанные кафедрой методические указания. Рекомендуется дорабатывать свой конспект лекций, делая в нем соответствующие записи из литературы, рекомендованной преподавателем и предусмотренной рабочей программой дисциплины. В процессе подготовки студент может дополнить список использованной литературы современными источниками, не представленными в списке рекомендованной литературы, и в дальнейшем использовать собственные подготовленные учебные материалы при написании курсового проекта и при подготовке к экзамену.

Форма организации студентов на лабораторных работах индивидуальная. При индивидуальной форме организации занятий каждый студент выполняет индивидуальное задание.

Если в результате выполнения лабораторной работы запланирована подготовка письменного отчета, то отчет о выполненной работе необходимо оформлять в соответствии с требованиями методических указаний. Качество выполнения лабораторных работ является важной составляющей оценки текущей успеваемости обучающегося.