

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кубанский государственный университет»
Физико-технический факультет

УТВЕРЖДАЮ
Проректор по учебной работе,
качеству образования – первый
проректор
Хагуров А.А.

подпись

« 31 » _____ 2019 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1.В.05 ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ НА C/C++

Направление подготовки 09.03.02 Информационные системы и технологии

Направленность (профиль) Программное обеспечение информационных систем в цифровой экономике

Форма обучения очная

Квалификация выпускника бакалавр

Краснодар 2019

Рабочая программа дисциплины Технологии программирования на С++ составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.02 Информационные системы и технологии (Цифровые вычислительные комплексы и сети)

Программу составил(и):

В.Н Значко, ст.преподаватель кафедры теоретической физики и компьютерных технологий,


подпись

Рабочая программа дисциплины Технологии программирования на С++ утверждена на заседании кафедры теоретической физики и компьютерных технологий

протокол № 9 «21» мая 2019г.

Заведующий кафедрой (разработчика) В.А. Исаев


подпись

Рабочая программа обсуждена на заседании кафедры теоретической физики и компьютерных технологий

протокол № 9 «21» мая 2019г.

Заведующий кафедрой (выпускающей) В.А. Исаев


подпись

Утверждена на заседании учебно-методической комиссии физико-технического факультета

протокол № 11 «21» мая 2019г.

Председатель УМК факультета Богатов Н.М.


подпись

Рецензенты:

Н.М. Богатов, зав. кафедрой физики и информационных систем КубГУ, д. м.-ф. наук

Л.Р. Григорьян, ген. директор ООО НПФм «Мезон», к. м.-ф. наук

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Цель курса – изучение современных технологий разработки и сопровождения как отдельных программных компонентов, так и информационных систем в целом.

1.2 Задачи дисциплины

выработка у студентов практических навыков необходимых для решения следующих производственных вопросов: программирование на структурных и объектно-ориентированных языках; документирование и сопровождение информационных систем; автоматизированная сборка информационных систем.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Технологии программирования на С/С++» относится к базовой части Блока 1 "Дисциплины (модули)" учебного плана и является дисциплиной по выбору 4-го семестра обучения для подготовки бакалавров направления 09.03.02 "Информационные системы".

Для успешного изучения дисциплины необходимы знания курса "Информатика". Освоение дисциплины необходимо для изучения других дисциплин.

1.4 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся профессиональных компетенций (ПК)

№ п.п.	Код и наименование компетенции	Индикаторы достижения компетенции		
		знает	умеет	владеет
1.	ПК-2 Способность осуществлять администрирование процесса контроля производительности сетевых устройств и программного обеспечения, проводить регламентные работы на сетевых устройствах и программном обеспечении инфокоммуникационной системы, осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации	правила работы с различными информационными системами и базами данных	работать с различными информационными системами и базами данных; осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации	навыками сбора, анализа и обработки статистической информации с целью оценки качества предоставляемых услуг, соответствия требованиям технических регламентов телекоммуникационного оборудования
2.	ПК-8 Способность обеспечивать защиту информации в автоматизированных системах в процессе их эксплуатации	архитектуру, протоколы и общие принципы функционирования аппаратных, программных и	основные принципы, криптографические протоколы и программные средства обеспечения	применять программные, аппаратные и программно-аппаратные средства защиты сетевых устройств от

№ п.п.	Код и наименование компетенции	Индикаторы достижения компетенции		
		знает	умеет	владеет
		программно-аппаратных средств администрируемой сети	информационной безопасности сетевых устройств	несанкционированного доступа

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач. ед. (144 часов), их распределение по видам работ представлено в таблице
(для студентов ОФО)

Вид учебной работы	Всего часов	Семестры (часы)			
		4			
Контактная работа, в том числе:					
Аудиторные занятия (всего):					
Занятия лекционного типа	16	16			
Лабораторные занятия	32	32			
Занятия семинарского типа (семинары, практические занятия)	32	32			
Иная контактная работа:					
Контроль самостоятельной работы (КСР)	2	2			
Промежуточная аттестация (ИКР)	0,3	0,3			
Самостоятельная работа, в том числе:	26	26			
Проработка учебного (теоретического) материала	8	8			
Самоподготовка	8	8			
Подготовка к текущему контролю	10	10			
Контроль:					
Подготовка к экзамену	35,7	35,7			
Общая трудоемкость	час.	144	144		
	в том числе контактная работа	82,3	82,3		
	зач. ед	4	4		

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые в 4 семестре (очная форма)

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Организация программного обеспечения	38	6	12	12	8
2.	Технологические методы и средства разработки ПО	33	5	10	10	8

3.	Отладка, тестирование и сопровождение программных продуктов	35	5	10	10	10
	<i>ИТОГО по разделам дисциплины</i>	106	16	32	32	26
	Контроль самостоятельной работы (КСР)	2				
	Промежуточная аттестация (ИКР)	0,3				
	Подготовка к экзамену	36,7				
	Общая трудоемкость по дисциплине	144				

Примечание: Л – лекции, ПЗ – практические занятия / семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Организация программного обеспечения	Этапы решения задач на ЭВМ. Виды программ. Структура жизненного цикла программы. Стоимость программного обеспечения. Надежность программных продуктов. Требования к надежности на различных этапах разработки программного обеспечения. Критерии оценки качества программного обеспечения.	ЛР
2.	Технологические методы и средства разработки ПО	Формы записи алгоритмов. Языки программирования и стандартные библиотеки. Средства языка программирования для решения различных классов задач. Библиотеки стандартных подпрограмм. Теория и методы структурного программирования. Объектно-ориентированное программирование. Стиль программирования. Стандартизация стиля. Требования к стилю написания программы. Эффективность программ. Оптимизирующие компиляторы. Методы оптимизации.	ЛР
3.	Отладка, тестирование и сопровождение программных продуктов	Ошибки программного обеспечения и методы их обнаружения. Отладка. Типы отладки. Использование инструментальных средств отладки. Тестирование. Различие между тестированием и отладкой. Типы и методы тестирования. Инструментальные средства.	ЛР

2.3.2 Занятия семинарского типа

№	Наименование раздела (темы)	Тематика практических занятий (семинаров)	Форма текущего
---	-----------------------------	---	----------------

			контроля
1	2	3	4
1.	Организация программного обеспечения	Этапы решения задач на ЭВМ. Структура жизненного цикла программы. Надежность программных продуктов. Критерии оценки качества программного обеспечения.	ЛР
2.	Технологические методы и средства разработки ПО	Формы записи алгоритмов. Библиотеки <code>stdio</code> , <code>stdlib</code> , <code>stdcpp</code> . Библиотеки контейнерных классов. Объектно-ориентированное и порождающее программирование. Стиль программирования. Стандартизация стиля. Требования к стилю написания программы. Правила выбора имен идентификаторов.	ЛР
3.	Отладка, тестирование и сопровождение программных продуктов	Ошибки программного обеспечения и методы их обнаружения. Отладка. Средства отладки в средах разработки <code>qtcreator</code> и <code>visual studio</code> . Тестирование. Различие между тестированием и отладкой. Типы и методы тестирования. Инструментальные средства.	тестирование

2.3.3 Лабораторные занятия

№	Наименование лабораторных работ	Форма текущего контроля
1	3	4
1.	Модульное программирование	Отчет по лабораторной работе
2.	Стиль программирования	Отчет по лабораторной работе
3.	Тестирование ПО	Отчет по лабораторной работе
4.	Структурное программирование	Отчет по лабораторной работе
5.	Объектно-ориентированное программирование	Отчет по лабораторной работе
6.	Контейнерные типы данных	Отчет по лабораторной работе

		работе
7.	Оптимизация программ	Отчет по лабораторной работе
8.	Методика отладки программ	Отчет по лабораторной работе
9.	Разработка документации к программе	Отчет по лабораторной работе

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрено

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Проработка учебного (теоретического) материала	Методические указания по организации аудиторной и внеаудиторной самостоятельной работы, утвержденные кафедрой теоретической физики и компьютерных технологий, протокол № 9 от «14» марта 2017 г.
2	Подготовка к текущему контролю	Методические рекомендации для проведения практических, семинарских и лабораторных занятий, утвержденные кафедрой теоретической физики и компьютерных технологий, протокол № 9 от «14» марта 2017 г.
3	Самоподготовка	Методические указания по организации аудиторной и внеаудиторной самостоятельной работы, утвержденные кафедрой теоретической физики и компьютерных технологий, протокол № 9 от «14» марта 2017 г.

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В современных условиях развитие продуктивных технологий в сфере образования становится неотъемлемой частью процесса модернизации. Заканчиваются возможности экстенсивного пути развития образования, при котором повышение образованности и профессиональности связывалось с увеличением объема знаний, и начинается переход к интенсивному пути развития образования. Он требует становления принципиально новых образовательных подходов в противовес широко распространенным сегодня репродуктивным технологиям, основанным на простом воспроизводстве информации. Новые технологии должны базироваться на продуктивности, креативности, мобильности и опираться на научное мышление, формирование которого у обучающихся становится основной задачей образовательного процесса.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

10. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «название дисциплины».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме лабораторных работ и **промежуточной аттестации** в форме вопросов к экзамену.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на экзамене;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

- в печатной форме увеличенным шрифтом,
- в форме электронного документа.

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Организация программного обеспечения	ПК-2 (знать); ПК-8 (уметь)	Лабораторная работа	Вопросы к экзамену
2	Технологические методы и средства разработки ПО	ПК-2 (владеть); ПК-8 (владеть)	Лабораторная работа	Вопросы к экзамену
3	Отладка, тестирование и сопровождение программных продуктов	ПК-2 (уметь); ПК-8 (знать)	Лабораторная работа, тест	Вопросы к экзамену

Показатели, критерии и шкала оценки сформированных компетенций

Код и наименование компетенций	Соответствие уровней освоения компетенции планируемым результатам обучения и критериям их оценивания		
	пороговый	базовый	продвинутый
	Оценка		
	Удовлетворительно /зачтено	Хорошо/зачтено	Отлично /зачтено
ПК-2 Способность осуществлять администрирование процесса контроля производительности сетевых устройств и программного обеспечения, проводить регламентные работы на сетевых устройствах и программном обеспечении инфокоммуникационной системы, осуществлять администрирование сетевой подсистемы инфокоммуникационной системы	Знает не в полном объеме правила работы с различными информационными системами и базами данных	Знает в достаточном объеме правила работы с различными информационными системами и базами данных	Знает в полном объеме правила работы с различными информационными системами и базами данных
	Умеет в требуемых пределах работать с различными информационными системами и базами данных; осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации	Умеет квалифицированно работать с различными информационными системами и базами данных; осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации	Умеет на высоком научном уровне работать с различными информационными системами и базами данных; осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации
	Владеет	Владеет свободно и	Владеет свободно и

организации	некоторыми навыками сбора, анализа и обработки статистической информации с целью оценки качества предоставляемых услуг, соответствия требованиям технических регламентов телекоммуникационного оборудования	большинством навыками сбора, анализа и обработки статистической информации с целью оценки качества предоставляемых услуг, соответствия требованиям технических регламентов телекоммуникационного оборудования	всеми навыками сбора, анализа и обработки статистической информации с целью оценки качества предоставляемых услуг, соответствия требованиям технических регламентов телекоммуникационного оборудования
ПК-8 Способность обеспечивать защиту информации в автоматизированных системах в процессе их эксплуатации	Знает не в полном объеме архитектуру, протоколы и общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой сети	Знает в достаточном объеме архитектуру, протоколы и общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой сети	Знает в полном объеме архитектуру, протоколы и общие принципы функционирования аппаратных, программных и программно-аппаратных средств администрируемой сети
	Умеет в требуемых пределах основные принципы, криптографические протоколы и программные средства обеспечения информационной безопасности сетевых устройств	Умеет квалифицированно основные принципы, криптографические протоколы и программные средства обеспечения информационной безопасности сетевых устройств	Умеет на высоком научном уровне основные принципы, криптографические протоколы и программные средства обеспечения информационной безопасности сетевых устройств
	Владеет некоторыми применять программные, аппаратные и программно-аппаратные средства защиты сетевых устройств от несанкционированного доступа	Владеет свободно и большинством применять программные, аппаратные и программно-аппаратные средства защиты сетевых устройств от несанкционированного доступа	Владеет свободно и всеми применять программные, аппаратные и программно-аппаратные средства защиты сетевых устройств от несанкционированного доступа

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Вопросы для проведения компьютерного тестирования:

[qdiv]Строка, которая начинается с `#include <stdio.h>` в Си-программе означает ...

[ediv]директиву препроцессору подключить заголовочный файл `stdio.h`[ediv]2

[ediv]директиву компоновщику подключить файл `stdio.h`[ediv]-1

[ediv]директиву компилятору откомпилировать заголовочный файл `stdio.h`[ediv]-1

[ediv]директиву препроцессору провести синтаксический анализ файла `stdio.h`[ediv]-1

[qdiv]Чем отличается препроцессорная директива `#include <myfile.h>` от `#include "myfile.h"`

[ediv]ничем[ediv]-2

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в текущем каталоге[ediv]-1

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в каталоге `windows`[ediv]-1

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в каталоге `/usr`[ediv]-1

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в каталоге для заголовочных файлов[ediv]2

[qdiv]Чем отличается препроцессорная директива `#include "myfile.h"` от `#include <myfile.h>`

[ediv]ничем[ediv]-2

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в текущем каталоге[ediv]2

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в каталоге `windows`[ediv]-1

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в каталоге `/usr`[ediv]-1

[ediv]препроцессор в первую очередь ищет файл `myfile.h` в каталогах перечисленных в переменной `BIN`[ediv]-1

[qdiv]Директива препроцессора `#ifdef` предназначена для ...

[ediv]условной компиляции[ediv]2

[ediv]работы с макросами[ediv]0

[ediv]объявления переменной[ediv]-1

[ediv]уничтожения переменной[ediv]-1

[qdiv]Директива препроцессора `#ifndef` предназначена для ...

[ediv]условной компиляции[ediv]2

[ediv]работы с макросами[ediv]0

[ediv]объявления переменной[ediv]-1

[ediv]уничтожения переменной[ediv]-1

[qdiv]Директива препроцессора `#endif` предназначена для ...

[ediv]условной компиляции[ediv]2

[ediv]работы с макросами[ediv]0

[ediv]объявления переменной[ediv]-1

[ediv]уничтожения переменной[ediv]-1

[qdiv]Директива препроцессора `#undef` предназначена для ...

[ediv]условной компиляции[ediv]-1

[ediv]работы с макросами[ediv]0

[ediv]объявления переменной[ediv]-1

[ediv]уничтожения переменной[ediv]2

[qdiv]Директива препроцессора #define предназначена для ...

[ediv]условной компиляции[ediv]-1

[ediv]работы с макросами[ediv]2

[ediv]объявления переменной[ediv]0

[ediv]уничтожения переменной[ediv]-1

[qdiv]В приведенном фрагменте кода найдите значение переменной a
#define TWOK 1024+1024

...

a=10*TWOK;

[ediv]11264[ediv]2

[ediv]2048[ediv]-1

[ediv]20480[ediv]-1

[ediv]это неправильно[ediv]-1

[qdiv]Лексема #else в Си-программе означает

[ediv]макрос[ediv]-1

[ediv]ключевое слово языка Си, управляющее ветвлением[ediv]-1

[ediv]ключевое слово языка Си, управляющее циклом[ediv]-1

[ediv]директиву препроцессора, управляющую условной компиляцией[ediv]2

[qdiv]Выберите в списке технологии программирования

[ediv]объектно-ориентированное программирование[ediv]1

[ediv]структурное кодирование[ediv]0

[ediv]ленивое программирование[ediv]-1

[ediv]структурное программирование[ediv]1

[qdiv]Выберите в списке технологии программирования

[ediv]объектно-ориентированное программирование[ediv]1

[ediv]техническое программирование[ediv]-1

[ediv]ленивое программирование[ediv]-1

[ediv]порождающее программирование[ediv]1

[qdiv]Выберите в списке фазы жизненного цикла программы

[ediv]проектирование[ediv]1

[ediv]форматирование[ediv]-1

[ediv]отладка[ediv]1

[ediv]сопровождение[ediv]1

[qdiv]Выберите в списке фазы жизненного цикла программы

[ediv]кодирование[ediv]1

[ediv]форматирование[ediv]-1

[ediv]отладка[ediv]1

[ediv]установка[ediv]-1

[qdiv]Во время проектирования программы выполняются следующие действия

[ediv]выбирается язык программирования[ediv]1

[ediv]выбирается система программирования[ediv]1

[ediv]поиск синтаксических ошибок[ediv]-1

[ediv]поиск логических ошибок[ediv]-1

[ediv]отладка[ediv]-1

[qdiv]Во время проектирования программы выполняются следующие действия

[ediv]выполняется разбиение на модули[ediv]1

[ediv]выбирается система программирования[ediv]1

[ediv]поиск синтаксических ошибок[ediv]-1

[ediv]определяются требования к ресурсам вычислительной системы[ediv]1

[ediv]отладка[ediv]-1

[qdiv]Во время проектирования программы выполняются следующие действия

[ediv]анализируется задача[ediv]-1

[ediv]выбирается язык программирования[ediv]1

[ediv]поиск синтаксических ошибок[ediv]-1

[ediv]определяются требования к ресурсам вычислительной системы[ediv]1

[ediv]отладка[ediv]-1

[qdiv]Отладка программы нужна для

[ediv]устранения ошибок[ediv]2

[ediv]успокоения пользователей[ediv]0

[ediv]"тонкой настройки" программы[ediv]-1

[ediv]улучшения читаемости кода[ediv]-1

[qdiv]Поиск логических ошибок в программе выполняется на этапе

[ediv]определения спецификаций[ediv]-1

[ediv]проектирования[ediv]-1

[ediv]реализации[ediv]-1

[ediv]отладки[ediv]1

[qdiv]Поиск синтаксических ошибок в программе выполняется на этапе

[ediv]определения спецификаций[ediv]-1

[ediv]проектирования[ediv]-1

[ediv]реализации[ediv]1

[ediv]отладки[ediv]-1

[qdiv]На этапе реализации выявляются

[ediv]синтаксические ошибки кодирования[ediv]2

[ediv]ошибки проектирования[ediv]0

[ediv]логические ошибки кодирования[ediv]-1

[ediv]ошибки при выборе сотрудника на должность программиста[ediv]0

[qdiv]Дороже всего обходится исправление ошибки обнаруженной на этапе

[ediv]определения требований[ediv]-1

[ediv]проектирования[ediv]-1

[ediv]реализации[ediv]-1

[ediv]отладки[ediv]-1

[ediv]сопровождения[ediv]2

[qdiv]По статистике большая часть времени на разработку программы
затрачивается на фазу

[ediv]определения требований[ediv]-1

[ediv]проектирования[ediv]-1

[ediv]реализации[ediv]-1
[ediv]отладки[ediv]2
[ediv]сопровождения[ediv]-1

[qdiv]Выберите в списке составляющие структурного программирования
[ediv]проектирование сверху-вниз[ediv]2
[ediv]анализ предметной области[ediv]-1
[ediv]правильное документирование[ediv]-1
[ediv]модульное программирование[ediv]2

[qdiv]Выберите в списке составляющие структурного программирования
[ediv]проектирование сверху-вниз[ediv]2
[ediv]структурное кодирование[ediv]2
[ediv]правильное документирование[ediv]-1
[ediv]своевременное исправление ошибок[ediv]-1

[qdiv]Выберите в списке три принципа структурного программирования
[ediv]проектирование сверху-вниз[ediv]2
[ediv]инкапсуляция данных[ediv]-2
[ediv]модульное программирование[ediv]2
[ediv]структурное кодирование[ediv]2
[ediv]наследование[ediv]-2
[ediv]полиморфизм[ediv]-2

[qdiv]Выберите в списке синтаксически корректные операторы цикла,
используемые в языке Си
[ediv]for(;;)[ediv]2
[ediv]for a in \$()[ediv]-2
[ediv]while(1);[ediv]2
[ediv]for(,)[ediv]-2

[qdiv]Выберите в списке синтаксически корректные операторы цикла,
используемые в языке Си
[ediv]for(;1;)[ediv]2
[ediv]for \$a in array[ediv]-2
[ediv]while(0);[ediv]2
[ediv]do{ }while[ediv]-1

[qdiv]Выберите в списке синтаксически правильные определения массива символов
[ediv]char a[]="any text";[ediv]1
[ediv]char a[100]="any text";[ediv]1
[ediv]char a='a';[ediv]-1
[ediv]char *a=&b;[ediv]-1
[ediv]char *a=0;[ediv]-1

[qdiv]Выберите в списке синтаксически правильные определения массива символов
[ediv]char a[]="any text";[ediv]1
[ediv]char a[100]=malloc(100);[ediv]-1
[ediv]char a=&b;[ediv]-1
[ediv]char *a="any text";[ediv]1
[ediv]char *a=(char*)malloc(100);[ediv]1

[qdiv]Прототип функции strcpy стандартной библиотеки Си находится в заголовочном файле ...

[ediv]stdio.h[ediv]-1
[ediv]string.h[ediv]2
[ediv]stdlib.h[ediv]-1
[ediv]conio.h[ediv]-1

[qdiv]Прототип функции printf стандартной библиотеки Си находится в заголовочном файле ...

[ediv]stdio.h[ediv]2
[ediv]string.h[ediv]-1
[ediv]stdlib.h[ediv]-1
[ediv]conio.h[ediv]-1

[qdiv]Прототип функции exit стандартной библиотеки Си находится в заголовочном файле ...

[ediv]stdio.h[ediv]-1
[ediv]string.h[ediv]-1
[ediv]stdlib.h[ediv]2
[ediv]conio.h[ediv]-1

[qdiv]Прототип функции strdup стандартной библиотеки Си находится в заголовочном файле ...

[ediv]stdio.h[ediv]-1
[ediv]string.h[ediv]2
[ediv]stdlib.h[ediv]-1
[ediv]conio.h[ediv]-1

[qdiv]Прототип функции strcmp стандартной библиотеки Си находится в заголовочном файле ...

[ediv]stdio.h[ediv]-1
[ediv]string.h[ediv]2
[ediv]stdlib.h[ediv]-1
[ediv]conio.h[ediv]-1

[qdiv]Для сравнения строк можно использовать функции ... стандартной библиотеки Си

[ediv]strcpy[ediv]-1
[ediv]strcmp[ediv]2
[ediv]strdup[ediv]-1
[ediv]strncpy[ediv]-1

[qdiv]Для копирования строк можно использовать функцию ... стандартной библиотеки Си

[ediv]strcpy[ediv]1
[ediv]strcmp[ediv]-1
[ediv]strncpy[ediv]1
[ediv]strdup[ediv]1
[ediv]strrev[ediv]-1

[qdiv]Для определения длины строки можно использовать функцию ... стандартной библиотеки Си

[ediv]strcpy[ediv]-1
[ediv]strcmp[ediv]-1
[ediv]strncpy[ediv]-1
[ediv]strlen[ediv]1
[ediv]strrev[ediv]-1

[qdiv]В результате работы этого фрагмента кода float f=123.456; printf("%2.3f", f); на экране будет напечатано

[ediv]123.456[ediv]2
[ediv]2.3[ediv]-1
[ediv]2.3f[ediv]-1
[ediv]123.45[ediv]-1

[qdiv]В результате работы этого фрагмента кода float f=123.456; printf("%8.1f", f); на экране будет напечатано

[ediv]123.456[ediv]-1
[ediv]2.3[ediv]-1
[ediv]2.3f[ediv]-1
[ediv]123.5[ediv]2
[ediv]123.45[ediv]-1

[qdiv]В результате работы этого фрагмента кода int f=123; printf("%04d", f); на экране будет напечатано

[ediv]123.4[ediv]-1
[ediv]1.230e+002[ediv]-1
[ediv]0123[ediv]2
[ediv]123[ediv]-1

[qdiv]В результате работы этого фрагмента кода int f=123; printf("%X", f); на экране будет напечатано

[ediv]0123[ediv]-1
[ediv]12[ediv]-1
[ediv]7B[ediv]2
[ediv]7b[ediv]-1

[qdiv]В результате работы этого фрагмента кода char *a="abcd"; printf("%2s", a); на экране будет напечатано

[ediv]2abcd[ediv]-1
[ediv]2ab[ediv]-1
[ediv]abcd[ediv]2
[ediv]2[ediv]-1

[qdiv]В результате работы этого фрагмента кода char *a="abcd"; printf("2%s", a); на экране будет напечатано

[ediv]2abcd[ediv]2
[ediv]2ab[ediv]-1
[ediv]abcd[ediv]-1
[ediv]2[ediv]-1

[qdiv]В результате работы этого фрагмента кода char *a="abcd"; printf("%%2%s", a); на экране будет напечатано

[ediv]%2abcd[ediv]2

[ediv]2abcd[ediv]-1
[ediv]%abcd[ediv]-1
[ediv]2%abcd[ediv]-1

[qdiv]В результате работы этого фрагмента кода double a, b=2e20+1;
a = b - 2.0e20;
printf(" %f ", a);

на экране будет напечатано

[ediv]0.000000[ediv]2
[ediv]0.000001[ediv]-1
[ediv]1.000000[ediv]-1
[ediv]-1.000000[ediv]-1

[qdiv]В результате работы этого фрагмента кода printf(" \\\"", a); на экране будет напечатано

[ediv]\\"[ediv]2
[ediv]\\"[ediv]-1
[ediv]\\\""[ediv]-1
[ediv]\\\""[ediv]-1

[qdiv]Одной из основ объектно-ориентированного программирования является ...

[ediv]свертывание[ediv]0
[ediv]инкапсуляция[ediv]2
[ediv]прозрачность[ediv]-1

[qdiv]Чем в языке C++ различается действие ключевых слов **class** и **struct** на описываемый класс?

[ediv]ничем[ediv]-1
[ediv]доступом к полям по умолчанию[ediv]2
[ediv]видом наследования данных[ediv]0
[ediv]видом наследования функций[ediv]0

[qdiv]Инкапсуляция в ООП означает ...

[ediv]сокрытие данных и функций класса[ediv]2
[ediv]сокрытие глобальных данных и функций[ediv]0
[ediv]сокрытие данных и функций объемлющей области видимости[ediv]-1
[ediv]сокрытие локальных данных и функций[ediv]-1

[qdiv]Чем отличаются полиморфизм и перегрузка операций в C++?

[ediv]при полиморфизме одинаковые имена имеют функции разных классов иерархии, а при перегрузке - одного класса[ediv]2
[ediv]при перегрузке одинаковые имена имеют функции разных классов иерархии, а при полиморфизме - одного класса[ediv]0

[qdiv]под термином "сцепление" в программировании подразумевают ...

[ediv]написание функций программы в одной строке[ediv]0
[ediv]написание функций программы без отступа между ними[ediv]0
[ediv]связь между программами или объектами пользовательского интерфейса[ediv]2

[qdiv]чтобы ограничить использование функции в языке C++ надо ...

[ediv]использовать спецификатор доступа private в описании класса[ediv]2

[ediv]использовать спецификатор доступа public в описании класса[ediv]0
[ediv]объявить функцию с ключевым словом const[ediv]0
[ediv]объявить функцию с ключевым словом static[ediv]0

[qdiv]почему нежелательно размещать 2 оператора на одной строке Си-программы?
[ediv]это ухудшает читаемость программы[ediv]2
[ediv]это запрещено синтаксисом языка Си[ediv]-2

[qdiv]Что такое lint?
[ediv]программа анализирующая исходный код программ[ediv]2
[ediv]тип данных long int в языке Си[ediv]0
[ediv]тип данных long int в языке Си++[ediv]0
[ediv]тип данных long int в языке Паскаль[ediv]0

[qdiv]препроцессор это ...
[ediv]программа, выполняющая синтаксическую проверку исходного кода программы[ediv]-1
[ediv]устройство, выполняющее предварительную обработку потока команд[ediv]0
[ediv]программа, выполняющая обработку исходного текста программы перед компиляцией[ediv]2
[ediv]устройство, выполняющее разделение команд на микрооперации[ediv]0

[qdiv]выберите в списке правильные утверждения
[ediv]Си-функция, объявленная с ключевым словом **inline** выполняется быстрее[ediv]-1
[ediv]Си-функция, объявленная с ключевым словом **inline** встраивается в точку вызова[ediv]2
[ediv]Си-функция, объявленная с ключевым словом **inline** увеличивает размер бинарного файла программы[ediv]2
[ediv]Си-функция, объявленная с ключевым словом **inline** не увеличивает размер бинарного файла программы[ediv]0

[qdiv]Почему вместо макросов с параметрами лучше использовать встраиваемые (inline) функции?
[ediv]встраиваемые функции выполняются быстрее чем макросы с параметрами[ediv]0
[ediv]в макросах труднее обнаруживать ошибки, т.к. они не выполняются в пошаговом режиме[ediv]2
[ediv]не знаю[ediv]-1

[qdiv]Вычислите значение переменной a: a=3; a=(++a)?a+1:a-1;
[ediv]3[ediv]0
[ediv]4[ediv]0
[ediv]5[ediv]2
[ediv]6[ediv]0

[qdiv]Операция **<<** в языке Си означает
[ediv]вывод на печать[ediv]-1
[ediv]сравнение[ediv]-1
[ediv]побитовый сдвиг влево[ediv]2
[ediv]возведение в степень[ediv]-1

[qdiv]Вычислите значение переменной c. int a[]={1, 2, 3, 4, 5}; int*b=a; int c=*(b++);
[ediv]1[ediv]0
[ediv]2[ediv]2
[ediv]3[ediv]0
[ediv]4[ediv]0
[ediv]5[ediv]0

[qdiv]Операция **::** в языке C++ означает ...
[ediv]уточнение области видимости[ediv]2
[ediv]поэлементное деление массива чисел[ediv]0
[ediv]взятие адреса объекта[ediv]0
[ediv]операцию разадресации[ediv]0

[qdiv]Операция разадресации указателя в языках Си и C++ обозначается символом ...
[ediv]->[ediv]0
[ediv].[ediv]0
[ediv]*[ediv]2
[ediv][][ediv]0

[qdiv]Выберите в списке правильные утверждения для абстрактного класса
[ediv]нельзя создать объект абстрактного класса[ediv]2
[ediv]можно создать объект абстрактного класса[ediv]0
[ediv]абстрактный класс содержит чистую виртуальную функцию[ediv]2
[ediv]абстрактный класс не может содержать виртуальных функций[ediv]-1
[ediv]абстрактный класс не должен иметь деструктор[ediv]0

[qdiv]Деструктор класса в языке C++ предназначен для ...
[ediv]уничтожения объекта[ediv]2
[ediv]инициализации объекта[ediv]0
[ediv]создания копии объекта[ediv]0
[ediv]загадочных действий[ediv]-1

[qdiv]Может ли деструктор класса в языке C++ быть виртуальным?
[ediv]да[ediv]2
[ediv]нет[ediv]0

[qdiv]Можно ли перегружать операции new и delete как члены классов?
[ediv]да[ediv]2
[ediv]нет[ediv]0
[ediv]только new[ediv]0
[ediv]только delete[ediv]0

[qdiv]Можно ли перегружать глобальные операции new и delete?
[ediv]да[ediv]2
[ediv]нет[ediv]0
[ediv]только new[ediv]0
[ediv]только delete[ediv]0

[qdiv]Ключевое слово **throw** в языке C++ предназначено для ...
[ediv]освобождения памяти[ediv]-2
[ediv]попытки выполнения кода[ediv]0

[ediv]генерации исключения[ediv]2
[ediv]перехвата исключения[ediv]0

[qdiv]Ключевое слово `catch` в языке C++ предназначено для ...
[ediv]освобождения памяти[ediv]-2
[ediv]попытки выполнения кода[ediv]0
[ediv]генерации исключения[ediv]0
[ediv]перехвата исключения[ediv]2

[qdiv]Ключевое слово `try` в языке C++ предназначено для ...
[ediv]освобождения памяти[ediv]-2
[ediv]попытки выполнения кода[ediv]2
[ediv]генерации исключения[ediv]0
[ediv]перехвата исключения[ediv]0

[qdiv]Какие ключевые слова из нижеперечисленных недопустимы в языке Си?
[ediv]try[ediv]2
[ediv]catch[ediv]2
[ediv]if[ediv]-1
[ediv]else[ediv]-1
[ediv]for[ediv]0

[qdiv]Какие ключевые слова из нижеперечисленных предназначены для выполнения ветвления в программе?
[ediv]for[ediv]-1
[ediv]switch[ediv]2
[ediv]while[ediv]-1
[ediv]if[ediv]2

[qdiv]Чему равно значение переменной a после выполнения приведенного фрагмента кода?
for(int a=0, b=10; b; a++, b--)
{
a++;
}
[ediv]0[ediv]0
[ediv]5[ediv]0
[ediv]10[ediv]0
[ediv]15[ediv]0
[ediv]20[ediv]2
[ediv]все ответы неправильные[ediv]0

[qdiv]Чему равно значение переменной a после выполнения приведенного фрагмента кода?
int a=0, b=10;
do{
++a++;
}while(b--)
[ediv]19[ediv]0
[ediv]20[ediv]0
[ediv]21[ediv]0
[ediv]22[ediv]2

[ediv]23[ediv]0

[ediv]все ответы неправильные[ediv]0

[qdiv]Выберите в списке правильное утверждение/утверждения

[ediv]В языке Си истиной считается любое ненулевое значение переменной[ediv]2

[ediv]В языке Си истиной считается любое неотрицательное значение переменной[ediv]-1

[ediv]В языке Си есть тип данных bool[ediv]-1

[ediv]В языке C++ есть тип данных bool[ediv]2

[qdiv]Выберите в списке правильное утверждение/утверждения

[ediv]В языке Си значение переменной до инициализации не определено[ediv]2

[ediv]В языке C++ значение переменной до инициализации = 0[ediv]-1

[ediv]В языке Си статические переменные инициализируются нулем[ediv]2

[ediv]В языке C++ статические переменные не инициализируются[ediv]-1

Список лабораторных работ

1. Модульное программирование
2. Структурное программирование
3. Объектно-ориентированное программирование
4. Контейнерные типы данных
5. Стиль программирования
6. Оптимизация программ
7. Тестирование ПО
8. Методика отладки программ
9. Разработка документации к программе

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Вопросы к экзамену

1. Предмет технологии разработки программного обеспечения.
2. Жизненный цикл программы
3. Критерии оценки качества ПО на различных этапах жизненного цикла программного обеспечения
4. Критерии оценки надежности программного обеспечения
5. Единая система программной документации. Программные документы, эксплуатационные документы. Общие требования к программным документам.
6. Формы записи алгоритмов. Блок-схема, структурограмма, псевдокод, естественный язык, PDL, UML.
7. Языки программирования. Классификация, область применения.
8. Использование библиотек функций и классов. MFC, OWL, Windows Forms, Turbo Vision, stdC, Qt.
9. Структурное программирование. Модульное программирование.
10. Структурное программирование. Структурное кодирование.
11. Структурное программирование. Проектирование сверху вниз.
12. ООП. Инкапсуляция. Права доступа. Примеры.
13. ООП. Наследование. Множественное наследование. Примеры.
14. ООП. Полиморфизм. Примеры.
15. ООП. Отличие от традиционного программирования

16. Линейное и событийное программирование. Примеры.
17. Порождающее программирование. Библиотека STL. Шаблоны.
18. Стиль программирования. Оформление исходного текста (отступы, пробелы, переносы, скобки)
19. Стиль программирования. Выбор имен идентификаторов. Правила Джексона.
20. Стиль программирования. Комментарии.
21. Эффективность программ. Оптимизация.
22. Методы оптимизации. Подпрограммы. Память.
23. Методы оптимизации. Арифметические операции и операции с плавающей точкой.
24. Методы оптимизации. Циклы. Работа с файлами.
25. Методы оптимизации. Оптимизация условных выражений.
26. Ошибки программного обеспечения: причины, источники и классификация
27. Методы обнаружения ошибок. Характерные ошибки
28. Отладка. Общая характеристика процесса.
29. Средства отладки
30. Методы тестирования. Этапы тестирования

Перечень компетенций (части компетенции), проверяемых оценочным средством

ПК-2 Способность осуществлять администрирование процесса контроля производительности сетевых устройств и программного обеспечения, проводить регламентные работы на сетевых устройствах и программном обеспечении инфокоммуникационной системы, осуществлять администрирование сетевой подсистемы инфокоммуникационной системы организации.

ПК-8 Способность обеспечивать защиту информации в автоматизированных системах в процессе их эксплуатации.

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации определяющие процедуры оценивания на экзамене

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является экзамен. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом

ФОС промежуточной аттестации состоит из вопросов и задач к экзамену по дисциплине.

Экзамен по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения экзамена: устно (письменно).

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заносится преподавателем в экзаменационную ведомость и зачетную книжку.

Критерии оценки:

оценка «отлично»: глубокие исчерпывающие знания всего программного материала, логически последовательные, полные, грамматически правильные и конкретные ответы на вопросы экзаменационного билета и дополнительные вопросы членов комиссии; использование в необходимой мере в ответах языкового материала, представленного в рекомендуемых учебных пособиях и дополнительной литературе.;

оценка «хорошо»: твёрдые и достаточно полные знания всего программного материала, последовательные, правильные, конкретные ответы на поставленные вопросы при свободном реагировании на замечания по отдельным вопросам;

оценка «удовлетворительно»: знание и понимание основных вопросов программы, наличие ошибок при недостаточной способности их корректировки, наличие определенного количества (не более 50%) ошибок в освещении отдельных вопросов билета;

оценка «неудовлетворительно»: непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы экзаменаторов.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1. Царев Р.Ю. Программирование на языке Си: учебное пособие [Электронный ресурс]. / Р.Ю. Царев; Министерство образования и науки Российской Федерации, Сибирский Федеральный университет. - Красноярск: Сибирский федеральный университет, 2014. - 108 с. - URL: <http://biblioclub.ru/index.php?page=book&id=364601>

2. Белоцерковская И.Е. Алгоритмизация. Введение в язык программирования С++ [Электронный ресурс]. / И.Е. Белоцерковская, Н.В. Галина, Л.Ю. Катаева. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 197 с. - URL: <http://biblioclub.ru/index.php?page=book&id=428935>

3. Программирование на языке С++ в среде Qt Creator [Электронный ресурс]. / Г.Г. Злобин, Д.А. Костюк, А.С. Чмыхало и др. - 2-е изд., испр. - М.: Национальный Открытый Университет «ИНТУИТ», 2016. - 716 с. - URL: <http://biblioclub.ru/index.php?page=book&id=428929>

Для освоения дисциплины инвалидами и лицами с ограниченными возможностями здоровья имеются издания в электронном виде в электронно-библиотечных системах «Лань» и «Юрайт».

5.2 Дополнительная литература:

1. Боэм Б., Браун Дж., Каспар Х. Характеристики качества программного обеспечения. М.: Мир, 1981. – 208с.

2. Брукс Ф.П. Как проектируются и создаются программные комплексы (Мифический человеко-месяц). М.: Наука, 1979. – 152с.

3. Гласс Р., Нуазо Р. Сопровождение программного обеспечения. М.: Мир, 1983. – 156с.

4. Зелковиц М., Шоу А. Принципы разработки ПО - М.: Мир, 1982.

5. Вельбицкий И.В. Технология программирования - Киев, 1984.

6. Майерс Г. Надежность программного обеспечения - М.: Мир, 1980.

7. Шураков В.В. Надежность программного обеспечения - М. Финансы и статистик, 1987.

8. Бутаков Е.А. Методы создания качественного программного обеспечения ЭВМ - М.: Энергоатомиздат, 1984.

9. Р. Гласс. Руководство по надежному программированию. М.: Финансы и статистика, 1982. – 256с.

10. Лингер Р., Миллс Х., Уитт Б. Теория и практика структурного программирования. М.: Мир, 1982. – 406с.

11. Липаев В.В. Качество программного обеспечения. М.: Финансы и статистика, 1983. – 263с.

12. Павловская Т.А. С++. Объектно-ориентированное программирование. Практикум: Учеб. пособие.-СПб.:Питер,2006.-264с.
13. Павловская Т.А. С\C++. Структурное программирование: Практикум. - СПб.:Питер,2007.-238с.
14. Д. Ван Тассел. Стил, разработка, эффективность, отладка и испытание программ. М.: Мир, 1985. – 332с.
15. А. Голуб. С и С++. Правила программирования. М.: БИНОМ, 1996. – 272с
16. У. Дал, Э. Дейкстра, К. Хоор. Структурное программирование. М.: Мир., 1975. – 247с.
17. Баженова И.Ю. Языки программирования. Учеб. Пособие. М.: АCADEMIA. – 2011. – 352с.

5.3. Периодические издания:

Периодические издания - не предусмотрены.

6. Методические указания для обучающихся по освоению дисциплины (модуля)

Рекомендации по оцениванию лабораторных работ

В целях закрепления практического материала и углубления теоретических знаний по разделам дисциплины «Технологии программирования на С/С++» предполагается выполнение лабораторных работ, что позволяет углубить процесс познания, раскрыть понимание прикладной значимости осваиваемой дисциплины. Комплект заданий репродуктивного уровня для выполнения на лабораторных занятиях, позволяющих оценивать и диагностировать знание фактического материала (базовые понятия, алгоритмы, факты) и умение правильно использовать специальные термины и понятия, распознавание объектов изучения в рамках определенного раздела дисциплины.

Критерии оценки лабораторных работ

Оценка	Критерии оценивания
5 баллов	Задание выполнено полностью, в представленном отчете обоснованно получено правильное выполненное задание.
4 балла	Задание выполнено полностью, но нет достаточного обоснования или при верном решении допущена незначительная ошибка, не влияющая на правильную последовательность рассуждений.
3 балла	Задания выполнены частично.
2 балла	Задание не выполнено.

В освоении дисциплины инвалидами и лицами с ограниченными возможностями здоровья большое значение имеет индивидуальная учебная работа (консультации) – дополнительное разъяснение учебного материала.

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта

между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

7.1 Перечень информационно-коммуникационных технологий

1. Использование электронных презентаций при проведении лекций.
2. Подготовка к тестированию и консультирование посредством электронной почты.
3. Выполнение лабораторных работ.

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

Программное обеспечение в рамках программы компании Microsoft “Enrollment for Education Solutions” DsktpEdu ALNG LicSAPk MVL

Дог. №67-АЭФ/223-ФЗ/2018 от 2018 Desktop Education ALNG LicSAPk MVL Pre2017EES A Faculty EES

Дог. №344/145 от 28.06.2018 Предоставление неисключительных имущественных прав на использование программного обеспечения «Антиплагиат» на один год

Контракт №59-АЭФ/223-ФЗ_2018 от 07.09.2018 Антивирусная защита физических рабочих станций и серверов:

Kaspersky Endpoint Security для бизнеса – Стандартный Russian Edition. 1500-2499 Node 1 year Educational Renewal License

Microsoft Windows 10;

Microsoft Office Professional Plus (№73–АЭФ/223-ФЗ/2018 Соглашение Microsoft ESS 72569510);

Microsoft Windows 10;

Microsoft Office Professional Plus (№73–АЭФ/223-ФЗ/2018 Соглашение Microsoft ESS 72569510);

Oracle Virtual Box 5.2 (бесплатное ПО)

7.3 Перечень современных профессиональных баз данных и информационных справочных систем

1. Справочно-правовая система «Консультант Плюс» (<http://www.consultant.ru>)
2. Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru/>)

8. Материально-техническое обеспечение по дисциплине (модулю)

№	Вид работ	Материально-техническое обеспечение дисциплины (модуля) и оснащенность
1.	<i>Лекционные занятия</i>	Лекционная аудитория, оснащенная презентационной техникой (проектор, экран, компьютер/ноутбук) и соответствующим программным обеспечением (ПО) для воспроизведения файлов формата jpg и avi, достаточным количеством посадочных мест. 300, 114, 209, 201 корп. С.
2.	<i>Семинарские занятия</i>	Аудитория для проведения семинарских занятий, оснащенная магнитно-маркерной доской, комплектом учебной мебели и презентационной техникой. 142, 114, 227, 209, 201 корп. С.
3.	<i>Лабораторные занятия</i>	Лаборатория, укомплектованная специализированной мебелью и техническими средствами обучения. 207, 212,

		213 корп. С.
4.	<i>Курсовое проектирование</i>	Не предусмотрено
5.	<i>Групповые (индивидуальные) консультации</i>	Аудитория для проведения групповых (индивидуальных) занятий, оснащенная доской и комплектом учебной мебели. 212, 213, 207 корп. С.
6.	<i>Текущий контроль, промежуточная аттестация</i>	Аудитория для текущего контроля и промежуточной аттестации студентов, оснащенная компьютерной техникой с возможностью подключения к сети "Интернет", с соответствующим программным обеспечением в режиме подключения к терминальному серверу, с программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета. 114, 212, 230 корп. С.
7.	<i>Самостоятельная работа</i>	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета. 208 корп. С.