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 Т. А.
подпись
«31» мая 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 C/C++ составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.02 «Информационные системы и технологии».

Программу составил(и):
В. Н. Значко, старший преподаватель
кафедры теоретической физики и
компьютерных технологий


подпись

Рабочая программа дисциплины Технологии программирования на C/C++ утверждена на заседании кафедры теоретической физики и компьютерных технологий
протокол № 9 «29» марта 2018 г.
Заведующий кафедрой (разработчика) Исаев В.А.


подпись

Рабочая программа обсуждена на заседании кафедры теоретической физики и компьютерных технологий
протокол № 9 «29» марта 2018 г.
Заведующий кафедрой (выпускающей) Исаев В.А.

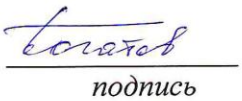

подпись

Утверждена на заседании учебно-методической комиссии физико-технического факультета
протокол № 10 «12» апреля 2018г.
Председатель УМК факультета Богатов Н.М.


подпись

Рецензенты:

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


подпись

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


подпись

1 Цели и задачи изучения дисциплины

1.1 Цель 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 информацион-ных систем; автоматизированная сборка информационных систем.

1.3 Место дисциплины в структуре образовательной программы

Дисциплина «Технологии программирования на C/C++» 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, соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих профессиональных компетенций:

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	**уметь**	**владеть**
1.	ОПК-4	пониманием сущности и значения информации в развитии современного информационного общества, соблюдение основных требований к информационной безопасности, в том числе защите государственной тайны	основные методы, способы и средства получения, хранения, переработки информации; основные требования к информационной безопасности	разрабатывать стратегию обеспечения информационной безопасности с использованием современных средств защиты; работать с традиционными носителями информации, распределенным и базами знаний; работать с информацией в глобальных компьютерных сетях;	навыками работы с компьютером как средством управления информацией
2.	ОПК-5	способностью использовать современные компьютерные технологии поиска	современные компьютерные технологии поиска информации для решения	проводить анализ информации и обосновывать принятые идеи и подходы к	современными компьютерными технологиями поиска информации для решения

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	**уметь**	**владеть**
		информации для решения поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению	поставленной задачи	решению задач с использованием современных компьютерных технологий	поставленной задачи, критического анализа этой информации и обоснования принятых идей и подходов к решению
3.	ПК-12	способностью разрабатывать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные)	принципы, базовые концепции технологий программирования, основные этапы и принципы создания программного продукта	применять современные технологии проектирования программных средств, использовать архитектурные и детализированные решения при проектировании программных средств	методами и средствами представления данных и знаний о предметной области, языками процедурного и объектно-ориентированного программирования

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

Вид учебной работы	Всего часов	Семестры (часы)			
		4			
Контактная работа, в том числе:	**82,3**	**82,3**			
Аудиторные занятия (всего):	**80**	**80**			
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	10	10	-	-	-
Подготовка к текущему контролю	10	10	-	-	-
Самоподготовка	6	6			
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 дисциплины, изучаемые во 4 семестре

№ раздела	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Организация программного обеспечения	16	4	8		4
2.	Технологические методы и средства разработки ПО	44	6	12	16	10
3.	Отладка, тестирование и сопровождение программных продуктов	46	6	12	16	12
	Итого по дисциплине:	106	16	32	32	26

2.3 Содержание разделов дисциплины

2.3.1 Занятия лекционного типа.

№	Наименование раздела	Содержание раздела	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.	ЛР

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

2.3.3 Лабораторные занятия.

№	Наименование раздела	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
7.	Организация программного обеспечения		Отчет по ЛР
8.	Технологические методы и средства разработки ПО	Модульное программирование Структурное программирование Объектно-ориентированное программирование	Отчет по ЛР

		Контейнерные типы данных Стиль программирования	
9.	Отладка, тестирование и сопровождение программных продуктов	Оптимизация программ Тестирование ПО Методика отладки программ Разработка документации к программе	Отчет по ЛР

Защита лабораторной работы (ЛР), выполнение курсового проекта (КП), курсовой работы (КР), расчетно-графического задания (РГЗ), написание реферата (Р), эссе (Э), коллоквиум (К), тестирование (Т) и т.д.

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.

Основные педагогические технологии

1. Традиционное обучение
2. Феноменологический подход
3. Интерактивные подходы
4. Эвристическое обучение
5. Программированное обучение
6. Контекстное обучение
7. Активное обучение
8. Дидактическая эвристика
9. Авторские педагогические технологии
10. Эмоционально-смысловой подход
11. Компьютерные технологии обучения
12. Разноуровневое обучение
13. Метод проектов
14. Учение через обучение
15. Технология парного обучения
16. Конструктивное обучение (конструктивистское обучение)
17. Нооген
18. Пренатальное обучение

Интерактивные подходы

Костяком интерактивных подходов являются интерактивные упражнения и задания, которые выполняются учащимися. Основное отличие интерактивных упражнений и заданий от обычных заключается в том, что они направлены не только и не столько на закрепление уже изученного материала, сколько на изучение нового. Современная педагогика богата целым арсеналом интерактивных подходов, среди которых можно выделить следующие:

- Творческие задания
- Работа в малых группах
- Обучающие игры (ролевые игры, имитации, деловые игры и образовательные игры)
- Использование общественных ресурсов (приглашение специалиста, экскурсии)
- Социальные проекты и другие внеаудиторные методы обучения (социальные проекты, соревнования, радио и газеты, фильмы, спектакли, выставки, представления, песни и сказки)
- Разминки
- Изучение и закрепление нового материала (интерактивная лекция, работа с наглядными пособиями, видео- и аудиоматериалами, «ученик в роли учителя», «каждый учит каждого», мозаика (ажурная пила), использование вопросов, Сократический диалог)
- Обсуждение сложных и дискуссионных вопросов и проблем («Займи позицию (шкала мнений)», ПОПС-формула, проективные техники, «Один — вдвоем — все вместе»,

«Смени позицию», «Карусель», «Дискуссия в стиле телевизионного ток-шоу», дебаты, симпозиум)

- Разрешение проблем («Дерево решений», «Мозговой штурм», «Анализ казусов», «Переговоры и медиация», «Лестницы и змейки»)

Творческие задания

Под творческими заданиями мы будем понимать такие учебные задания, которые требуют от учащихся не простого воспроизводства информации, а творчества, поскольку задания содержат больший или меньший элемент неизвестности и имеют, как правило, несколько подходов. Творческое задание составляет содержание, основу любого интерактивного метода. Творческое задание (особенно практическое и близкое к жизни обучающегося) придает смысл обучению, мотивирует учащихся. Неизвестность ответа и возможность найти свое собственное «правильное» решение, основанное на своем персональном опыте и опыте своего коллеги, друга, позволяют создать фундамент для сотрудничества, сообучения, общения всех участников образовательного процесса, включая педагога. Выбор творческого задания сам по себе является творческим заданием для педагога, поскольку требуется найти такое задание, которое отвечало бы следующим критериям:

- не имеет однозначного и односложного ответа или решения
- является практическим и полезным для учащихся
- связано с жизнью учащихся
- вызывает интерес у учащихся
- максимально служит целям обучения

Если учащиеся не привыкли работать творчески, то следует постепенно вводить сначала простые упражнения, а затем все более сложные задания.

Работа в малых группах

Работа в малых группах — это одна из самых популярных стратегий, так как она дает всем учащимся (в том числе и стеснительным) возможность участвовать в работе, практиковать навыки сотрудничества, межличностного общения (в частности, умение активно слушать, вырабатывать общее мнение, разрешать возникающие разногласия). Все это часто бывает невозможно в большом коллективе. Работа в малой группе — неотъемлемая часть многих интерактивных методов, например таких, как мозаика, дебаты, общественные слушания, почти все виды имитаций и др.

При организации групповой работы, следует обращать внимание на следующие ее аспекты. Нужно убедиться, что учащиеся обладают знаниями и умениями, необходимыми для выполнения группового задания. Нехватка знаний очень скоро даст о себе знать — учащиеся не станут прилагать усилий для выполнения задания. Надо стараться сделать свои инструкции максимально четкими. Маловероятно, что группа сможет воспринять более одной или двух, даже очень четких, инструкций за один раз, поэтому надо записывать инструкции на доске и (или) карточках. Надо предоставлять группе достаточно времени на выполнение задания.

Критическое мышление

Идея развития критического мышления является достаточно новой для российской дидактики. Заговорили о целостной технологии развития критического мышления лишь в середине 90-х годов. Но уже сегодня сторонников развития критического мышления учащихся достаточно много.

Критическое мышление означает не негативность суждений или критику, а разумное рассмотрение разнообразия подходов с тем, чтобы выносить обоснованные суждения и решения. Ориентация на критическое мышление предполагает вежливый скептицизм (ничто не принимается на веру), сомнение в общепринятых истинах, означает выработку точки зрения по определенному вопросу и способность отстоять эту точку зрения логическими доводами. Критическое мышление не является отдельным навыком, оно сочетает в себе следующие умения:

- выражать свои мысли (устно и письменно) ясно, уверенно и корректно по отношению к окружающим;

- аргументировать свою точку зрения и учитывать точки зрения других;
- брать на себя ответственность;
- работать с увеличивающимся и постоянно обновляющимся информационным потоком;
- задавать вопросы, самостоятельно формулировать гипотезу;
- решать проблемы;
- вырабатывать собственное мнение на основе осмысления различного опыта, идей и представлений;
- участвовать в совместном принятии решения;
- выстраивать конструктивные взаимоотношения с другими людьми.

Метод проектов

Основной его тезис: я знаю, для чего мне надо то, что я познаю, где и как я могу эти знания применить. Каждый обучаемый, принимая участие в проектировании, находит себе дело с учетом уровня своего интеллектуального развития, уровня подготовки по данной проблеме, своих способностей и задатков. Для того чтобы проект получился, надо верить в обучаемого. Мое твердое убеждение – нет плохих учеников. Они все яркие, талантливые, неповторимые индивидуальности.

Основные требования к использованию метода проектов:

1. Наличие значимой в исследовательском творческом плане проблемы / задачи, требующей интегрированного знания, исследовательского поиска для ее решения (например, исследование демографической проблемы в разных регионах мира; создание серии репортажей из разных концов земного шара одной проблеме и т.п.).
2. Практическая, теоретическая, познавательная значимость предполагаемых результатов. Например, доклад о демографическом состоянии данного региона, факторах, влияющих на это состояние, тенденциях, прослеживающихся в развитии данной проблемы; выпуск газеты, план мероприятий и т.п.
3. Самостоятельная (индивидуальная, парная, групповая) деятельность учащихся.
4. Использование исследовательских методов:
   - определение проблемы и вытекающих из нее задач исследования;
   - выдвижение гипотезы их решения;
   - обсуждение методов исследования;
   - обсуждение способов оформления конечных результатов (презентаций, защиты, творческих отчетов и т.п.);
   - сбор, систематизация и анализ полученных данных;
   - подведение итогов, оформление результатов, их презентация;
   - выводы, выдвижение новых проблем исследования.

Таким образом, метод проектов является одной из самых результативных и прогрессивных педагогических технологий. Он позволяет развивать познавательные навыки учащихся, критическое мышление, умение самостоятельно конструировать свои знания, ориентироваться в информационном пространстве.

Метод «мозгового штурма»

Существуют разные формы «мозгового штурма»: групповая прямая (совместный поиск возможных решений имеющейся задачи); групповая обратная (определение недостатков в имеющейся проблеме); индивидуальная (каждый участник за короткий промежуток времени должен сформулировать не менее одной оригинальной идеи).

Перед началом «мозгового штурма» необходимо создать у обучающихся доброжелательный настрой, добиться раскованности. При проведении «мозгового штурма» возможны лишь уточняющие вопросы, абсолютно неприемлемы критические замечания и промежуточные оценки, а поощрение и поддержка партнеров приветствуется. Участники должны формулировать суждения и идеи кратко и четко, действовать по принципу «чем больше идей, решительнее атака, тем ближе достижение цели штурма».

Дискуссия

Она является одной из важнейших форм образовательной деятельности, стимулирующей инициативность учащихся. Учебный материал в ходе дискуссии усваивается за счет:

- обмена информацией между участниками;
- разных подходов к одному и тому же предмету;
- сосуществования различных, вплоть до взаимоисключающих, точек зрения;
- возможности критиковать и даже отвергать любое мнение;
- поиска группового соглашения в виде общего мнения или решения.

Задача дискуссии – коллективно, с разных точек зрения, под разными углами обсудить и исследовать спорные моменты. Основные правила ведения дискуссии:

- нельзя критиковать людей, только их идеи;
- цель дискуссии не в определении победителя, а в консенсусе;
- все участники должны быть вовлечены в дискуссию;
- выступления должны проходить организованно, с разрешения ведущего, перепалка недопустима;
- каждый участник должен иметь право и возможность высказаться;
- обсуждению подлежат все позиции; – в процессе дискуссии участники могут изменить свою позицию;
- строить аргументацию необходимо на бесспорных фактах;
- в заключение всегда должны подводиться итоги.

По ходу дискуссии преподаватель должен следить, чтобы слишком эмоциональные и разговорчивые учащиеся не подменили тему, и чтобы критика позиций друг друга была обоснованной. Соединение работы в группах с решением проблемной ситуации создает наиболее эффективные условия для обмена знаниями, идеями и мнениями, обеспечивает всесторонний анализ и обоснованный выбор решения той или иной темы. Студенты овладевают ораторскими умениями, искусством ведения полемики, что само по себе вносит важный вклад в их личностное развитие.

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

4 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Вопросы для проведения компьютерного тестирования:

[qdiv]Строка, которая начинается с <B>#include <stdio.h></B> 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 <B>#include <myfile.h></B> от <B>#include "myfile.h"</B>
[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 <B>#include "myfile.h"</B> от <B>#include <myfile.h></B>
[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]техничное 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 сиволов
[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 сиволов
[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 сравнеия 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 результатае 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 результатае 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 результатае 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 результатае 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 результатае 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 результатае 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 результатае 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 результатае 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 результатае 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 изначает
[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 С++ значение переменной до инициализации = 0[ediv]-1
[ediv]В языке Си статические переменные инициализируются нулем[ediv]2
[ediv]В языке С++ статические переменные не инициализируются[ediv]-1

4.2 Фонд оценочных средств для проведения промежуточной аттестации.

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

Студенты обязаны сдать зачет в соответствии с расписанием и учебным планом. Зачет является формой контроля усвоения студентом учебной программы по дисциплине или ее части, выполнения практических, контрольных, реферативных работ.

Результат сдачи зачета по прослушанному курсу должны оцениваться как итог деятельности студента в семестре, а именно - по посещаемости лекций, результатам работы на практических занятиях, выполнения самостоятельной работы. При этом допускается на очной форме обучения пропуск не более 20% занятий, с обязательной отработкой пропущенных семинаров. Студенты у которых количество пропусков, превышает установленную норму, не выполнившие все виды работ и неудовлетворительно работавшие в течение семестра, проходят собеседование с преподавателем, который опрашивает студента на предмет выявления знания основных положений дисциплины.

Критерии оценки:

- оценка «зачтено»: студент владеет теоретическими знаниями по данному разделу, знает основную теорию дисциплины, допускает незначительные ошибки; студент умеет правильно объяснять материал, иллюстрируя его примерами.

- оценка «не зачтено»: материал не усвоен или усвоен частично, студент затрудняется привести примеры, довольно ограниченный объем знаний программного материала

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.

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

5.1 Основная литература:

1) Значко В.Н. Технология программирования: учебное пособие / В.Н. Значко; М-во образования и науки Рос. Федерации, Кубанский гос. ун-т. - Краснодар: [Кубанский государственный университет], 2012. - 48 с.

2) 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

3) 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

4) 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\С++. Структурное программирование: Практикум. - СПб.:Питер,2007.-238с.
14. Д. Ван Тассел. Стиль, 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.: ACADEMIA. – 2011. – 352с.

5.3. Периодические издания:

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», в том числе современные профессиональные базы данных и информационные справочные системы, необходимые для освоения дисциплины (модуля).

1. БД Web of Science - главный ресурс для исследователей по поиску и анализу научной литературы, охватывающей около 18000 научных журналов со всего мира. База данных международных индексов научного цитирования http://webofscience.com/
2. zbMATH - полная математическая база данных. Охватывает материалы с конца 19 века. zbMATH содержит около 4000000 документов из более 3000 журналов и 170000 книг по математике, статистике, информатике. https://zbmath.org/
3. БД Kaggle - это платформа для сбора и обработки данных. Является он-лайн площадкой для научного моделирования. https://www.kaggle.com/
4. База данных Научной электронный библиотеки eLIBRARY.RU https://elibrary.ru/
5. База данных Всероссийского института научной и технической информации (ВИНИТИ) РАН http://www2.viniti.ru/
6. «ЭЛЕКТРОННАЯ БИБЛИОТЕКА ДИССЕРТАЦИЙ» Российской Государственной Библиотеки (РГБ) – в настоящее время ЭБД содержит более 800 000 полных текстов диссертаций. https://dvs.rsl.ru
7. Портал открытых данных Российской Федерерации https://data.gov.ru

8. База открытых данных Министерства труда и социальной защиты РФ https://rosmintrud.ru/opendata
9. Федеральный портал единое окно доступа к информационным ресурсам - http://window.edu.ru/
10. Российский фонд фундаментальных исследований предоставляет доступ к информационным наукометрическим базам данных и полнотекстовым научным ресурсами издательств Springer Nature и Elsevier - http://www.rfbr.ru/rffi/ru
11. Федеральный портал "Информационно-коммуникационные технологии в образовании" - http://www.ict.edu.ru/
12. «Лекториум ТВ» – видеолекции ведущих лекторов России. Лекториум – on-line – библиотека, где ВУЗы и известные лектории России презентуют своих лучших лекторов. Доступ к материалам свободный и бесплатный - http://www.lektorium.tv.

7. 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 «Технология разработки C#» 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.

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

8.1 Перечень информационных технологий.

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

8.2 Перечень необходимого лицензионного программного обеспечения.

Электронный калькулятор Операционная система Windows

1) Архиваторы WinRAR, WinZip Windows
2) Антивирусные программы Kaspersky Anti-Virus, Dr.Web Windows
3) Растровый графический редактор Paint Операционная система Windows

4) Программа разработки презентаций Microsoft PowerPoint Дистрибутив Microsoft Office
5) Электронные таблицы Microsoft Excel Дистрибутив Microsoft Office
6) Текстовый процессор Microsoft Word Дистрибутив Microsoft Office
7) Браузер Internet Explorer Операционная система Windows
8) Браузер Mozilla Windows
9) Компиляторы Basic, Pascal, C++
10) Система математических вычислений MathCAD
11) Система математических вычислений MatLAB

9. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине (модулю).

№	Вид работ	Материально-техническое обеспечение дисциплины (модуля) и оснащенность
1.	*Лекционные занятия*	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.	*Семинарские занятия*	Аудитория для проведения семинарских занятий, оснащенная магнитно-маркерной доской, комплектом учебной мебели и презентационной техникой. 142, 114, 227, 209, 201 корп. С.
3.	*Лабораторные занятия*	Лаборатория, укомплектованная специализированной мебелью и техническими средствами обучения. 207, 212, 213 корп. С.
4.	*Курсовое проектирование*	Не предусмотрено
5.	*Групповые (индивидуальные) консультации*	Аудитория для проведения групповых (индивидуальных) занятий, оснащенная доской и комплектом учебной мебели. 212, 213, 207 корп. С.
6.	*Текущий контроль, промежуточная аттестация*	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.	*Самостоятельная работа*	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.