

Министерство науки и высшего образования Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кубанский государственный университет»
факультет математики и компьютерных наук

УТВЕРЖДАЮ
Проректор по учебной работе,
качеству образования – первый
проректор



Хайруров Т.А.
«29» мая 2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Б1.В.ДВ.01.01 Объектно-ориентированное программирование и компьютерный инжиниринг

Направление подготовки: 01.05.01 Фундаментальные математика и механика

Специализация: Вычислительная механика и компьютерный инжиниринг

Форма обучения: очная

Квалификация: Математик. Механик. Преподаватель

Краснодар 2020

Рабочая программа дисциплины «Объектно-ориентированное программирование и компьютерный инжиниринг» составлена в соответствии с Федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.05.01 Фундаментальные математика и механика

Программу составил:

С.И. Фоменко, к.ф.-м.н., доцент

подпись

Рабочая программа утверждена на заседании кафедры теории функций, протокол № 8 «17» марта 2020 г.

Заведующий кафедрой теории функций Голуб М.В. _____

Рабочая программа обсуждена на заседании кафедры теории функций протокол № 8 «17» марта 2020 г.

Заведующий кафедрой теории функций Голуб М.В. _____

Утверждена на заседании учебно-методической комиссии факультета математики и компьютерных наук протокол № 2 «30» апреля 2020 г.

Председатель УМК факультета Шмалько С.П. _____

Рецензенты:

Гусаков Валерий Александрович, канд. физ. – мат. наук, директор ООО «Просвещение – Юг»

Засядко Ольга Владимировна, доцент кафедры информационных образовательных технологий, канд. физ. - мат. наук, доцент

1 Цели и задачи изучения дисциплины .

1.1 Цель освоения дисциплины.

Целью изучения дисциплины «Введение в метод конечных элементов» является освоение современных технологий объектно-ориентированного программирования для компьютерного моделирования и разработки пользовательских приложений для решения естественно-научных и инженерных задач, развитие профессиональных компетентностей и приобретение практических навыков решения программистских и инженерных задач современными численными методами и приемами программирования. Цели дисциплины соответствуют формируемой компетенции ПК-5 и ПК-6 и позволяют подготовить обучающихся к успешной работе в различных сферах, применяющих математические методы, математическое и компьютерное моделирование, а также информационные технологии и развить способности самостоятельно приобретать и применять новые знания и умения. Полученные в результате освоения дисциплины навыки и умения позволяют поднять общий уровень исследовательской, математической и программистской культуры обучающихся.

1.2 Задачи дисциплины

- Знакомство с концепцией объектно-ориентированного программирования, формами ее реализации в высокоуровневых языках программирования.
- Формирование профессиональных компетенций для разработки сложных программных комплексов на основе методов объектно-ориентированного программирования.
- Развитие навыков компьютерного моделирования в естественных и инженерных науках.
- Поднять общий уровень исследовательской, математической и программистской культуры обучающихся.
- Выработать умения использовать справочные материалы и пособия в своей профессиональной деятельности.

1.3 Место дисциплины в структуре образовательной программы.

Дисциплина «Объектно-ориентированное программирование и компьютерный инжиниринг» относится к вариативной части блока 1 «Дисциплины (модули)» учебного плана и является дисциплиной по выбору.

Для освоения дисциплины студент должен владеть знаниями, умениями и навыками по дисциплинам «Технология программирования и работа на электронно-вычислительной машине (ЭВМ)» (Б1.Б.06), «Численные методы» (Б1.Б.05), «Линейная алгебра» (Б1.Б.12), «Дифференциальные уравнения»(Б1.Б.15).

1.4 Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения образовательной программы.

Программа определяет общий объем знаний, позволяющий сформировать у студента культуру решения как простых вспомогательных задач, так и умения программировать, владение общими принципами построения вычислительных алгоритмов, написания и отладки компьютерных программ.

Компетенции обучающегося, формируемые в результате освоения дисциплины «Объектно-ориентированное программирование и компьютерный инжиниринг»:

№ п.п.	Индекс компетенции	Содержание компетенции (или её части)	В результате изучения учебной дисциплины обучающиеся должны		
			знать	уметь	владеть
1.	ПК-5	способность использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	основные приемы программирования на основе объектно-ориентированных технологий для компьютерного моделирования при решении прикладных задач	реализовать компьютерные алгоритмы иерархией объектов для реализации алгоритмов решения численных задач математической физики и инженерии	навыками программирования и разработки эффективных иерархических алгоритмов
2.	ПК-6	способностью к творческому применению современных специализированных программных комплексов, включению в них собственных моделей, методов и алгоритмов	основные концепции разработки вычислительных программ и приложения для компьютерного моделирования	самостоятельно разрабатывать математические модели, алгоритмы и компьютерные программы, проводить анализ производительности программ и анализ результатов расчета	навыками анализа структуры вычислительных алгоритмов, связями между задачами и промежуточными методами компьютерного проекта

2. Структура и содержание дисциплины.

2.1 Распределение трудоёмкости дисциплины по видам работ.

Общая трудоёмкость дисциплины составляет 2 зач.ед. (72 часа), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)
		7 семестр
Контактная работа (всего), в том числе:		
Аудиторные занятия (всего)	40	40
В том числе:		
Занятия лекционного типа	10	10

Лабораторные занятия		26	26
Иная контактная работа			
В том числе:			
Контроль самостоятельной работы (КСР)		2	2
Промежуточная аттестация (ИКР)		0,2	0,2
Самостоятельная работа, в том числе:			
Проработка учебного материала		5	5
Выполнение индивидуальных заданий		20	20
Подготовка к текущему контролю		6,8	6,8
Контроль			
Общая трудоемкость	час.	72	72
	в том числе контактная работа	42,2	42,2
	зач. ед	2	2

2.2 Структура дисциплины:

Распределение видов учебной работы и их трудоемкости по разделам дисциплины, разделы дисциплины, изучаемые в 7 семестре (*очная форма*)

№	Наименование разделов	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Концепция объектно-ориентированного программирования (ООП).	12	4		4	4
2.	Сравнительный анализ реализации ООП в языках C++, C#, Python и Fortran	29	5		14	10
3.	Разработка компьютерных моделей на основе ООП для решения прикладных задач	22,8	1		6	15,8
4.	Анализ эффективности ООП и вычисленный эксперимент	4	0		2	2
	<i>Итого по дисциплине:</i>	67,8	10		26	31,8

Примечание: Л – лекции, ЛР – лабораторные занятия, СРС – самостоятельная работа студента; ИКР – иная контактная работа

2.3 Содержание разделов дисциплины:

2.3.1 Занятия лекционного типа.

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	2	3	4
1	Концепция объектно-ориентированного программирования (ООП).	Методологии процедурного, структурного и объектно-ориентированного программирования, принципы ООП, абстракция, инкапсуляция, наследование и полиморфизм. Класс и объект, поля, методы и свойства.	Устный опрос
2	Сравнительный анализ реализации ООП в языках C++, C#, Python и Fortran	Реализации ООП в C++, C#, Python и Fortran, разработка классов и решения проблем наследования и полиморфизма.	Устный опрос
3	Разработка компьютерных моделей на основе ООП для решения прикладных задач	Математическое и компьютерное моделирование, алгоритмизация, дизайн компьютерного проекта, анализ кода.	Устный опрос
4	Анализ эффективности ООП и вычисленный эксперимент	Методика оценки эффективности программ.	Устный опрос

2.3.2 Занятия семинарского типа – не предусмотрены.

2.3.3 Лабораторные занятия.

№	№ раздела дисциплины	Наименование лабораторных работ	Форма текущего контроля
1		3	4
1.	1	Методологии процедурного, структурного и объектно-ориентированного программирования	Опрос по теоретическому материалу . Выполнение практических индивидуальных заданий.
2.	1	Основы ООП: класс, поля и методы	Опрос по теоретическому материалу . Выполнение практических индивидуальных заданий.
3.	1	Основы ООП: наследование	Опрос по теоретическому материалу . Выполнение практических индивидуальных заданий.
4.	2	Основы программирования в С++, ООП в С++	Опрос по теоретическому материалу . Выполнение практических индивидуальных заданий.
5.	2	Основы программирования в С#, ООП в С#	Проверка выполнения практических индивидуальных заданий.
6.	2	Основы программирования в Fortran, ООП в Fortran 2003	Проверка выполнения практических индивидуальных заданий.
7.	2	Основы программирования в Python, ООП в Python	Проверка выполнения практических индивидуальных заданий.
8.	3	Разработка математической модели и построение алгоритма	Проверка выполнения практических индивидуальных заданий.

9.	3	Разработка структуры программы на основе концепции ООП и реализация компьютерной модели.	Проверка выполнения практических индивидуальных заданий.
10	4	Анализ эффективности ООП и вычисленный эксперимент	Проверка выполнения практических индивидуальных заданий.

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

№	Вид самостоятельной работы	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Проработка и повторение лекционного материала, материала учебной и научной литературы	Методические указания для подготовки к лекционным и семинарским занятиям, утвержденные на заседании кафедры
2	Подготовка к лабораторным занятиям	Методические указания по выполнению лабораторных работ, утвержденные на заседании кафедры
3	Подготовка к решению задач	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры
4	Подготовка к текущему контролю	Методические указания по выполнению самостоятельной работы, утвержденные на заседании кафедры

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.

С точки зрения применяемых методов используются как традиционные информа-

ционно-объяснительные *лекции*, так и интерактивная подача материала с мультимедийной системой и др. Такое сочетание позволяет оптимально использовать отведенное время и раскрывать логику и содержание дисциплины.

Лекции представляют собой систематический обзор понятий и методов Комплексного анализа с подачей материала в форме презентаций и с использованием других интерактивных технологий: проблемное обучение, моделирование, дискуссия.

Занятия, проводимые с использованием интерактивных технологий

№	Наименование разделов (тем)	Количество часов	
		всего ауд. часов	интерактивные часы
1	2	3	4
1.	Концепция объектно-ориентированного программирования (ООП).	8	6
2.	Сравнительный анализ реализации ООП в языках C++, C#, Python и Fortran	19	8
3.	Разработка компьютерных моделей на основе ООП для решения прикладных задач	7	1
4.	Анализ эффективности ООП и вычисленный эксперимент	2	0
	<i>Итого по дисциплине:</i>	39	15

Лабораторное занятие позволяет научить студента применять теоретические знания при решении и исследовании конкретных задач, развить математическую интуицию и творческое мышление. Разбор конкретных ситуаций, математическое моделирование задач, встречающихся на практике (проблемное обучение), командная работа, визуализация и обсуждение результатов анализа широко используется при проведении лабораторных, а также самостоятельных работ.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием возможностей средств удаленного доступа (электронная почта, видеоконференция).

4. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации.

Текущий контроль знаний студентов представляет собой:

- выполнение расчетно-графических заданий;

4.1 Фонд оценочных средств для проведения текущего контроля.

Фонд оценочных средств дисциплины состоит из средств текущего контроля выполнения заданий, лабораторных работ, средств для промежуточной аттестации (зачета)

Оценка успеваемости осуществляется по результатам:

- выполнения лабораторных работ;
- оценки, выставяемой при сдаче индивидуальных заданий;

Зачет выставляется по результатам выполненных индивидуальных заданий и текущей работы на лабораторных занятиях.

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	Концепция объектно-ориентированного программирования (ООП).	ПК-5, ПК-6	Отчет по лабораторной работе №1-3	Вопросы к зачету 1-5
2	Сравнительный анализ реализации ООП в языках C++, C#, Python и Fortran	ПК-6	Отчеты по лабораторным работам №4-7	Индивидуальное задание № 1 Вопросы к зачету 6-20
3	Разработка компьютерных моделей на основе ООП для решения прикладных задач	ПК-2, ПК-3	Отчеты по лабораторным работам №8-9	Индивидуальное задание № 2
4	Анализ эффективности ООП и вычисленный эксперимент	ПК-5	Отчеты по лабораторным работам №10	Индивидуальное задание № 2

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-5: способен использовать методы математического и алгоритмического моделирования при решении теоретических и прикладных задач	<i>Знает</i> – базовые понятия и методы дисциплины, возможные сферы их связи и приложения в других областях математического знания и дисциплинах естественно-научного и профессионального цикла;	<i>Знает</i> – основные методы и понятия, изучаемые в дисциплине, возможные сферы их связи и приложения в других областях математического знания и дисциплинах естественно-научного и профессионального цикла;	<i>Знает</i> - основные методы, понятия, изучаемые в дисциплине, сферы их связи и приложения в других областях математического знания и дисциплинах естественнонаучного и профессионального цикла;
	<i>Умеет</i> – решать базовые задачи, встречающиеся в практике профессиональной деятельности, сформулированные в терминах данной математической дисциплины, применять основные методы решения; использовать графические и текстовые редакторы, а также составлять научно-технические отчеты, инструкции и описание разработанных программ по ГОСТ	<i>Умеет</i> - решать основные задачи, встречающиеся в практике профессиональной деятельности, сформулированные в терминах данной математической дисциплины, применять основные и продвинутые методы решения; использовать графические и текстовые редакторы, а также визуализировать полученные результаты, составлять научно-технические отчеты, инструкции и описание разработанных программ по ГОСТ	<i>Умеет</i> – формулировать в терминах данной математической дисциплины задачи, встречающиеся в практике профессиональной деятельности, решать их с помощью основных и продвинутых методов; использовать графические и текстовые редакторы, а также визуализировать полученные результаты, составлять на высоком уровне научно-технические отчеты, инструкции и описание разработанных программ по ГОСТ
	<i>Владеет</i> отдельными методами применения математических методов, рассмотренных в рамках дисциплины, для решения профессиональных задач; навыками составления технических описаний и инструкций	<i>Владеет</i> основными методами применения математических методов, рассмотренных в рамках дисциплины, для решения профессиональных задач; навыками составления научно-технических описаний и инструкций	<i>Владеет</i> методами применения полученных знаний в постановке и решении прикладных задач; навыками составления научно-технических описаний и инструкций

<p>ПК-6: Способен к творческому применению современных специализированных программных комплексов, включению в них собственных моделей, методов и алгоритмов</p>	<p><i>Знает</i> – базовые методы, понятия, определения и свойства математических объектов, изучаемых в дисциплине, формулировки и утверждений; методы алгоритмизации математической модели</p>	<p><i>Знает</i> – основные методы, основные понятия, определения и свойства математических объектов, изучаемых в дисциплине; методы алгоритмизации математической модели, разработки соответствующих компьютерных программ на высоком уровне</p>	<p><i>Знает</i> - основные методы, основные понятия, определения и свойства математических объектов, изучаемых в дисциплине, формулировки утверждений, методы их доказательства; знаком с нестандартными подходами к решению задач; методы алгоритмизации математической модели, разработки соответствующих компьютерных программ на высоком уровне</p>
	<p><i>Умеет</i> – формулировать базовые утверждения, решать базовые задачи дисциплины; программировать разработанные алгоритмы на языках высокого уровня, проводить расчеты</p>	<p><i>Умеет</i> - формулировать и доказывать базовые утверждения дисциплины, решать основные задачи дисциплины, применять полученные навыки в других областях и дисциплинах естественно-научного цикла; программировать разработанные алгоритмы на языках высокого уровня, проводить расчеты.</p>	<p><i>Умеет</i> - формулировать и доказывать основные утверждения дисциплины, решать основные и продвинутые задачи, применять полученные навыки в других областях и дисциплинах естественнонаучного цикла; проводить доказательства нестандартным путем; программировать разработанные алгоритмы на языках высокого уровня, проводить расчеты и анализировать полученные результаты</p>
	<p><i>Владеет</i> – методами решения базовых задач и базовыми понятиями, рассматриваемые в дисциплине, а также методами разработки программных проектов. Способен работать в команде разработчиков.</p>	<p><i>Владеет</i> - методами решения задач, а также методами разработки программных комплексов, навыками применения этого в других областях и дисциплинах естественно-научного цикла. Способен работать в команде разработчиков.</p>	<p><i>Владеет</i> - методами решения задач, а также методами разработки программных комплексов, навыками применения этого в других областях и дисциплинах естественнонаучного цикла. Способен работать в команде разработчиков в качестве лидера. Демонстрирует дополнительные знания и эрудицию.</p>

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

4.1.1. Методические рекомендации к выполнению лабораторных работ

Для успешного выполнения лабораторной работы обучающемуся следует ознакомиться с теоретической частью дисциплины по теме лабораторной работы, изложенной в лекциях. Для углубленного понимания теоретического материала могут быть использованы источники, указанные в списке основной литературы [1-4], дополнительной [5-7].

Критерием должной подготовки студентов к выполнению лабораторных работ являются приобретенные знания, позволяющие безошибочно ответить на вопросы, сформулированные по каждой теме лабораторных работ. Для приобретения должных навыков к решению задач предполагается решение задач на лабораторных занятиях в учебных аудиториях под руководством преподавателя. Закрепление приобретенных навыков осуществляется внеаудиторным самостоятельным решением студентом задач. Номера задач для решения в аудитории и дома указаны к каждой лабораторной работе.

4.2 Фонд оценочных средств для проведения промежуточной аттестации.

Промежуточный контроль осуществляется в конце семестра в форме зачета.

Образцы расчетно-графических заданий к лабораторным работам

1-6

Задание 1. Создать класс Point, содержащий следующие члены класса:

1. Поля:
 - `float x, y;`
2. Конструкторы, позволяющие создать экземпляр класса:
 - с нулевыми координатами;
 - с заданными координатами.
3. Методы, позволяющие:
 - рассчитать расстояние от начала координат до точки;
 - переместить точку на плоскости на вектор (a, b).
4. Свойство:
 - позволяющее получить-установить координаты точки (доступное для чтения и записи);
5. Индексатор, позволяющий по индексу 0 обращаться к полю x, по индексу 1 – к полю y, при других значениях индекса выдается сообщение об ошибке.
6. Перегрузку:
 - Метода ToString() для форматного представления на экране: "(x,y)"
 - операции + со скаляром: одновременно добавляет к полям x и y значение скаляра.
 - операции * на скаляр: одновременно умножает поля x и y на значение скаляра.
 - Операции сравнения == и != двух точек.

Продемонстрировать работу класса.

Задание 2. Создать класс Triangle, содержащий следующие члены класса:

1. Поля:
 - `float a, b, c;` (длины сторон)
2. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
3. Методы, позволяющие:
 - рассчитать периметр треугольника p;
 - рассчитать площадь треугольника .
4. Свойство:
 - позволяющее получить-установить длины сторон треугольника (доступное для чтения и записи);
 - позволяющее установить, существует ли треугольник с данными длинами сторон (доступное только для чтения).
5. Индексатор, позволяющий по индексу 0 обращаться к полю a, по индексу 1 – к полю b, по индексу 2 – к полю c, при других значениях индекса выдается сообщение об ошибке.
6. Перегрузку:
 - Метода ToString() для форматного представления на экране: "(a,b,c)"
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей a, b и c на 1;
 - констант true и false: обращение к экземпляру класса дает значение true, если треугольник с заданными длинами сторон существует, иначе false;

- операции *: одновременно домножает поля a, b и c на скаляр.
- операции равенства == и !=: треугольники считаются равными, если равны их площади.

Продемонстрировать работу класса.

Задание 3. Создать класс Rectangle, содержащий следующие члены класса:

1. Поля:
 - float a, b; (длины сторон)
2. Конструктор, позволяющий создать экземпляр класса с заданными длинами сторон.
3. Методы, позволяющие:
 - вывести длины сторон прямоугольника на экран;
 - рассчитать периметр прямоугольника;
 - рассчитать площадь прямоугольника.
4. Свойство:
 - позволяющее получить-установить длины сторон прямоугольника (доступное для чтения и записи);
 - позволяющее установить, является ли данный прямоугольник квадратом (доступное только для чтения).
5. Индексатор, позволяющий по индексу 0 обращаться к полю a, по индексу 1 – к полю b, при других значениях индекса выдается сообщение об ошибке.
6. Перегрузку:
 - Метода ToString() для форматного представления на экране: “[a,b]”
 - операции ++ (--): одновременно увеличивает (уменьшает) значение полей a и b на 1;
 - констант true и false: обращение к экземпляру класса дает значение true, если прямоугольник с заданными длинами сторон является квадратом, иначе false;
 - операции *: одновременно домножает поля a и b на скаляр.
 - операции равенства == и !=: прямоугольники считаются равными, если равны их площади.

Продемонстрировать работу класса.

Задание 4.

Создать класс Point, содержащий следующие члены класса:

1. Поля:
 - float x, y;
2. Конструктор, позволяющий создать экземпляр класса:
 - с заданными координатами.
3. Методы:
 - Статический метод, позволяющий рассчитать расстояние от двух точек;
4. Свойство:
 - позволяющее получить-установить координаты точки (доступное для чтения и записи);
5. Перегрузку:
 - Метода ToString() для форматного представления на экране: “(x,y)”

Создать класс Triangle, содержащий следующие члены класса:

1. Поля:
 - Point A, B, C; (точки треугольника)
2. Конструкторы, позволяющие создать экземпляр класса с заданными точками.
 - С параметрами типа Point;
 - С 6 параметрами типа float, одна пара соответствует точке;
3. Методы, позволяющие:
 - рассчитать площадь треугольника.
4. Индексатор, позволяющий по индексу 0 обращаться к полю A, по индексу 1 – к полю B, по индексу 2 – к полю C, при других значениях индекса выдается сообщение об ошибке.
7. Перегрузку: Метода ToString() для форматного представления на экране: “{A;B;C}”

Продемонстрировать работу классов.

Задание 5.

Создать класс Point, содержащий следующие члены класса:

1. Поля:
 - float x, y;
2. Конструктор, позволяющий создать экземпляр класса:
 - с заданными координатами.
3. Методы:
 - Статический метод, позволяющий рассчитать расстояние от двух точек;
4. Свойство:
 - позволяющее получить-установить координаты точки (доступное для чтения и записи);
5. Перегрузку:
 - Метода ToString() для форматного представления на экране: '(x,y)'

Создать класс Rectangle, содержащий следующие члены класса:

1. Поля:
 - Point A, B; (левая верхняя и правая нижняя точки)
 2. Конструкторы, позволяющие создать экземпляр класса с заданными точками.
 - С параметрами типа Point;
 - С 4 параметрами типа float, одна пара соответствует точке;
 3. Методы, позволяющие:
 - рассчитать периметр прямоугольника.
 - рассчитать площадь прямоугольника.
 4. Индексатор, позволяющий по индексу 0 обращаться к полю A, по индексу 1 – к полю B, при других значениях индекса выдается сообщение об ошибке.
 5. Перегрузку метода ToString() для форматного представления на экране: '[A;B]'
- Продемонстрировать работу классов.

Задание 6. Создать класс Vector, содержащий следующие члены класса:

1. Поля:
 - float[] v;
2. Конструкторы, позволяющий создать экземпляр класса:
 - с заданным размером и нулевыми координатами.
 - с фактически переданным одномерным массивом.
3. Методы:
 - Скалярного умножения векторов (статический метод);
 - Проекции одного вектора на другой; $pr_a b = a \cdot b / |a|$
 - Определение угла между векторами: $\psi = \arccos\left(\frac{a \cdot b}{|a| |b|}\right)$ (статический метод)
4. Свойство:
 - длина вектора;
 - индексатор, позволяющее получить-установить координату вектора, если индекс меньше длины, иначе выдается сообщение об ошибке;
5. Перегрузку:
 - Метода ToString для форматного представления на экране: '{x1,x2,...}'
 - Операции сложения (+) векторов
 - Операции умножения (*) вектора на число
 - Операции скалярного умножения (*) векторов

- Операции \wedge -- нахождение угла между векторами

Создать класс Vector3 трехмерных векторов, являющийся наследником класса Vector.

Написать метод векторного произведения и перегрузить для него операцию $\&$:

$$\vec{a} \& \vec{b} \equiv [\vec{a} \times \vec{b}] = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix}; \vec{i} = \{1,0,0\}; \vec{j} = \{0,1,0\}; \vec{k} = \{0,0,1\}$$

Продемонстрировать работу классов.

7)

Разработать алгоритм и компьютерную вычислительную программу для решения краевой задачи для гармонических колебаний струны методом конечных элементов с кусочно-линейной аппроксимацией; найти точное аналитическое решение, сравнить его с приближенным, исследовать сходимость решения в зависимости от частоты колебаний ω :

- В 1. $u''(x) + k^2u(x) = x^2, \quad u(0) = 1, \quad u'(1) = 0, \quad k = \omega/2$
- В 2. $u''(x) + k^2u(x) = 2x, \quad u'(0) = 1, \quad u(1) = 0, \quad k = \omega/3$
- В 3. $u''(x) + k^2u(x) = 2x-1, \quad u(0) = 2, \quad u'(1) = 0, \quad k = \omega/3$
- В 4. $u''(x) + k^2u(x) = \frac{x^2}{2}, \quad u'(0) = 1, \quad u(1) = 0, \quad k = \omega/4$
- В 5. $u''(x) + k^2u(x) = e^x, \quad u(0) = 1, \quad u(1) = 0, \quad k = \omega$
- В 6. $u''(x) + k^2u(x) = e^{-x}, \quad u'(0) = 1, \quad u(1) = 0, \quad k = \omega/2$
- В 7. $u''(x) + k^2u(x) = x(1-x), \quad u'(0) = 1, \quad u(1) = 0, \quad k = \omega/2$
- В 8. $u''(x) + k^2u(x) = x(1-x^2), \quad u(0) = 1, \quad u'(1) = 0, \quad k = \omega$
- В 9. $u''(x) + k^2u(x) = x^3, \quad u'(0) = 2, \quad u'(1) = 0, \quad k = \omega$
- В 10. $u''(x) + k^2u(x) = \cos x, \quad u(0) = 1, \quad u'(1) = 0, \quad k = \omega$

По результатам работы необходимо подготовить итоговый отчет, включающий в себя описания постановки задачи, метода решения, а также и анализа численного анализа и выводы по результатам выполнения лабораторной работы

Примерные перечень вопросов к Зачету по предмету

1. Понятие объектно-ориентированное программирование (ООП). Отличия ООП от процедурного программирования. Понятие класса и экземпляра.
2. Принципы объектно-ориентированного программирования: инкапсуляция, полиморфизм, наследование.
3. Платформа .NET Framework, ее назначение и структура. Обзор технологий .NET.
4. Структура языка C#. Понятие класса и объекта. Класс object. Классификация типов. Переменные и константы.
5. Понятие инкапсуляция и механизмы ее реализации в C#.
6. Понятие полиморфизм и механизмы его реализации в Python.
7. Понятие наследование и механизмы его реализации в C++..
8. Проектирование класса в C#: данные, конструкторы, деструкторы и методы. Создание и инициализация экземпляра класса (объекта). Вызов методов класса.
9. Проектирование класса: свойства, индексаторы.

10. Проектирование класса: перегрузка методов класса и операторов (математических операторов, операторов сравнения и преобразования типов).
11. Проектирование класса: делегаты и события.
12. Операции языка Fortran. Приоритеты операций. Выражения. Приведение типов в выражениях.
13. Операторы языка: следования, ветвления, цикла, безусловного перехода. Вложение операторов.
14. Массив как объект. Одномерные, многомерные и «рваные» массивы и их сравнительная характеристика.
15. Строка как объект. Изменяемые и неизменяемые строки и их сравнительная характеристика.
16. Анонимные методы, лямбда-выражения, замыкание и каррирование.
17. Понятие «коллекция» в C#. Интерфейсные коллекции. Обзор коллекций Stack, Queue, ArrayList.
18. Интерфейсы: основные понятия и использование интерфейсов в построении иерархии классов. Стандартные интерфейсы .Net. Интерфейс IComparable.
19. Организация консольного ввода-вывода данных. Форматированный вывод.
20. Иерархия потоков: байтовые и символьные потоки. Организация файлового ввода вывода данных.

Критерии выставления оценок.

Оценка «отлично»:

- систематизированные, глубокие и полные знания по всем разделам дисциплины, а также по основным вопросам, выходящим за пределы учебной программы;
- точное использование научной терминологии систематически грамотное и логически правильное изложение ответа на вопросы;
- безупречное владение инструментарием учебной дисциплины, умение его эффективно использовать в постановке научных и практических задач;
- выраженная способность самостоятельно и творчески решать сложные проблемы и нестандартные ситуации;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной учебной программой по дисциплине;
- умение ориентироваться в теориях, концепциях и направлениях дисциплины и давать им критическую оценку, используя научные достижения других дисциплин;
- творческая самостоятельная работа на практических/семинарских/лабораторных занятиях, активное участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;
- высокий уровень сформированности заявленных в рабочей программе компетенций.

Оценка «хорошо»:

- достаточно полные и систематизированные знания по дисциплине;
- умение ориентироваться в основном теориях, концепциях и направлениях дисциплины и давать им критическую оценку;
- использование научной терминологии, лингвистически и логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- владение инструментарием по дисциплине, умение его использовать в постановке и решении научных и профессиональных задач;
- усвоение основной и дополнительной литературы, рекомендованной учебной программой по дисциплине;

- самостоятельная работа на практических занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий;
- средний уровень сформированности заявленных в рабочей программе компетенций.

Оценка «удовлетворительно»:

- достаточный минимальный объем знаний по дисциплине;
- усвоение основной литературы, рекомендованной учебной программой;
- умение ориентироваться в основных теориях, концепциях и направлениях по дисциплине и давать им оценку;
- использование научной терминологии, стилистическое и логическое изложение ответа на вопросы, умение делать выводы без существенных ошибок;
- владение инструментарием учебной дисциплины, умение его использовать в решении типовых задач;
- умение под руководством преподавателя решать стандартные задачи;
- работа под руководством преподавателя на практических занятиях, допустимый уровень культуры исполнения заданий;
- достаточный минимальный уровень сформированности заявленных в рабочей программе компетенций.

Оценка «неудовлетворительно»:

- фрагментарные знания по дисциплине;
- отказ от ответа (выполнения письменной работы);
- знание отдельных источников, рекомендованных учебной программой по дисциплине;
- неумение использовать научную терминологию;
- наличие грубых ошибок;
- низкий уровень культуры исполнения заданий;
- низкий уровень сформированности заявленных в рабочей программе компетенций.

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 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.

5.1 Основная литература:

1. Барков, И. А. Объектно-ориентированное программирование : учебник / И. А. Барков. — Санкт-Петербург : Лань, 2019. — 700 с. — ISBN 978-5-8114-3586-9. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/119661>

2. Оксюта, О. В. Объектно-ориентированное программирование : учебное пособие / О. В. Оксюта, В. В. Лавлинский. — Воронеж : ВГЛУ, 2018. — 96 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/118684>

3 Васильев, А. Н. Объектно-ориентированное программирование на C++ / А. Н. Васильев. — Санкт-Петербург : Наука и Техника, 2016. — 544 с. — ISBN 978-5-94387-984-5. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/90227>

4. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка C# : учебное пособие / Л. А. Залогова. — 2-е изд., стер. — Санкт-Петербург : Лань, 2020. — 192 с. — ISBN 978-5-8114-4757-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/126160>

5. Зализняк, В. Е. Численные методы. Основы научных вычислений : учебник и практикум для академического бакалавриата / В. Е. Зализняк. — 2-е изд., перераб. и доп. — М. : Издательство Юрайт, 2018. — 356 с. — (Серия : Бакалавр. Академический курс). — ISBN 978-5-534-02714-3. — URL: <https://biblio-online.ru/book/9D9516CB-A065-4497-9062-5D8C77D8E644/chislennye-metody-osnovy-nauchnyh-vychisleniy>

6. Компьютерное моделирование в авиакосмической промышленности / под редакцией И. Б. Аббасова. — Москва : ДМК Пресс, 2019. — 300 с. — ISBN 978-5-97060-634-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131726>

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

а. Дополнительная литература:

6. Голуб М. В., Фоменко С. И., Шпак А. Н. Метод спектральных конечных элементов в задачах математической физики: учеб. пособие. Краснодар: Кубанский гос. ун-т, 2015.
7. В. А. Биллиг. Основы программирования на C# . - М. : БИНОМ. Лаб. знаний, 2009.
8. А. Чакраборти, Ю. Кранти, Р. Дж. Сандху. Microsoft.NET Framework: разработка профессиональных проектов . - СПб. : БХВ-Петербург, 2003.
9. Просиз, Джеф. Программирование для Microsoft. NET - М. : Рус. Ред., 2003.

5.3. Периодические издания:

1. Журнал "Вычислительная механика сплошных сред"
<http://www2.icmm.ru/journal/>

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. <https://e.lanbook.com/> (Электронно-библиотечная система)
2. Российская система прочностного анализа на основе метода спектральных конечных элементов Fidesys <http://www.cae-fidesys.com/ru/about/info>

7. Методические указания для обучающихся по освоению дисциплины.

По курсу предусмотрено проведение лекционных занятий, на которых дается основной теоретический материал, лабораторных занятий, позволяющих студентам в полной мере ознакомиться с понятиями и методами Комплексного анализа и навыками их применением в решении практических задач.

Важнейшим этапом является самостоятельная работа по дисциплине. Целью самостоятельной работы магистрская является углубление знаний, полученных в результате аудиторных занятий. Вырабатываются навыки самостоятельной работы. Закрепляются опыт и знания, полученные во время лабораторных занятий.

Самостоятельная работа студентов в ходе изучения дисциплины состоит в выполнении индивидуальных заданий, задаваемых преподавателем, ведущим лабораторные занятия, подготовки теоретического материала к лабораторным занятиям, на основе конспектов лекций и учебной литературы, согласно календарному плану и подготовки теоретического материала к тестовому опросу, зачету и экзамену, согласно вопросам к экзамену.

Указания по оформлению работ:

Отчет по выполнению лабораторной работы и индивидуальных расчетно-графических заданий должен быть подготовлен в соответствии с ГОСТ 7.32-2001 и содержать:

- титульный лист;
- введение;
- постановку задачи;
- краткое описание последовательного алгоритма;
- подробное описание параллельного алгоритма;
- текст разработанной программы на выбранном языке программирования;
- тестовые примеры и результаты тестирования программы: оценка ускорения и эффективности разработанного параллельного алгоритма, оптимальные размеры входных данных на которых достигается максимум ускорения при различном числе узлов вычислительной системы и др.;
- заключение
- список использованной литературы.

Проверка индивидуальных заданий по темам, разобранным на лабораторных занятиях, осуществляется через неделю на текущем лабораторном занятии, либо в течение недели после этого занятия на консультации.

Для разъяснения непонятных вопросов лектором и ассистентом еженедельно проводятся консультации, о времени которых группы извещаются заранее.

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.

Учебная деятельность проходит в соответствии с графиком учебного процесса. Процесс самостоятельной работы контролируется во время аудиторных занятий и индивидуальных консультаций. Самостоятельная работа студентов проводится в форме изучения отдельных теоретических вопросов по предлагаемой литературе и выполнении практических заданий по разобранным во время аудиторных занятий примерам.

Фонд оценочных средств дисциплины состоит из средств текущего контроля (см. список задач и вопросов коллоквиума) и итоговой аттестации (зачета, экзамена).

В качестве оценочных средств, используемых для текущего контроля успеваемости, предлагается перечень вопросов, которые прорабатываются в процессе освоения курса. Данный перечень охватывает все основные разделы курса, включая знания, получаемые во время самостоятельной работы. Кроме того, важным элементом технологии является самостоятельное решение студентами и сдача заданий. Это полностью индивидуальная форма обучения. Студент рассказывает свое решение преподавателю, отвечает на дополнительные вопросы

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

8.1 Перечень информационных технологий.

1. Мультимедиа и коммуникационные технологии.
2. Элементы дистанционных технологий.
3. Мировые информационные образовательные ресурсы.
4. Аудиовизуальные и интерактивные средства обучения.
5. Мобильное обучение.
6. Облачные технологии.

8.2 Перечень необходимого программного обеспечения.

- MS Office: MS Word, MS Excel, MS PowerPoint
- Matlab
- Fortran, C++, C#, Python

8.3 Перечень информационных справочных систем:

1. Электронная библиотечная система eLIBRARY.RU (<http://www.elibrary.ru>)
2. Электронная библиотечная система «Университетская библиотека ONLINE» (<http://www.biblioclub.ru>)
3. ЭБС Издательства «Лань» <http://e.lanbook.com/> ООО Издательство «Лань»
4. ЭБС «Университетская библиотека онлайн» www.biblioclub.ru ООО «Директ-Медиа»

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

№	Вид работ	Материально-техническое обеспечение дисциплины (модуля) и оснащенность
1.	Лекционные занятия	Лекционная аудитория, специально оборудованная мультимедийными демонстрационными комплексами, учебной мебелью

2.	Лабораторные занятия	Помещение для проведения лабораторных занятий оснащенное учебной мебелью, доской маркером или мелом
3.	Групповые (индивидуальные) консультации	Помещение для проведения групповых (индивидуальных) консультаций, учебной мебелью, доской маркером или мелом
4.	Текущий контроль, промежуточная аттестация	Помещение для проведения текущей и промежуточной аттестации, оснащенное учебной мебелью.
5.	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