

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



\* Т.А. Хагуров

подпись

«31» мая 2024 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 2024

Рабочая программа дисциплины Б1.В.ДВ.01.01 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(и):  
Голуб М. В., зав. кафедрой, д. ф.-м. н.



Рабочая программа дисциплины Б1.В.ДВ.01.01 ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ И КОМПЬЮТЕРНЫЙ ИНЖИНИРИНГ утверждена на заседании кафедры ТЕОРИЯ ФУНКЦИИ протокол №12 от 07.05.2024 г.  
Заведующий кафедрой (разработчика) Голуб М. В.



Утверждена на заседании учебно-методической комиссии факультета математики и компьютерных наук протокол №3 «14» мая 2024 г.  
Председатель УМК факультета/института Шмалько С. П.



Рецензенты:

Фоменко Сергей Иванович, канд. физ. - мат. наук,  
старший научный сотрудник лаборатории волновых процессов

**Анопко Михаил Викторович,**  
**Генеральный директор ООО «УК АЙСТРИМ»**

## 1 Цели и задачи изучения дисциплины.

### 1.1 Цель освоения дисциплины.

Цель дисциплины «Объектно-ориентированное программирование и компьютерный инжиниринг» 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.

### 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;
- приобрести навыки решения программистских и инженерных задач современными численными методами и приемами программирования.

### 1.3 Место дисциплины (модуля) в структуре образовательной программы.

Дисциплина «Объектно-ориентированное программирование и компьютерный инжиниринг» относится к вариативной части профессионального цикла Блока1 "Дисциплины (модули)" учебного плана (Б1.В.ДВ). Для успешного освоения дисциплины обучающийся должен владеть знаниями, умениями и навыками по программе дисциплин Б1.О.20 «Линейная алгебра», Б1.О.23 «Дифференциальные уравнения», Б1.О.14 «Технология программирования и работа на электронно-вычислительной машине (ЭВМ)», Б1.О.13 «Численные методы».

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

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

Код и наименование индикатора достижения компетенции	Результаты обучения по дисциплине
<b>ПК-4 Способен разрабатывать программное обеспечение для решения прикладных задач в сфере профессиональной деятельности</b>	
ИПК-4.4. Ориентируется в современных алгоритмах компьютерной математики и имеет практический опыт разработки программных модулей на основе механико-математических моделей	Знает основные приемы программирования на основе объектно-ориентированных технологий для компьютерного моделирования при решении прикладных задач; основные концепции разработки вычислительных программ и приложения для компьютерного моделирования.
	Умеет реализовать компьютерные алгоритмы иерархий объектов для реализации алгоритмов решения численных задач математической физики и инженерии, проводить анализ производительности программ и анализ результатов расчета.
	Владеет навыками программирования и разработки эффективных иерархических алгоритмов, анализа структуры вычислительных алгоритмов, связями между задачами и промежуточными методами компьютерного проекта.

Результаты обучения по дисциплине достигаются в рамках осуществления всех видов контактной и самостоятельной работы обучающихся в соответствии с утвержденным учебным планом.

Индикаторы достижения компетенций считаются сформированными при достижении соответствующих им результатов обучения.

## 2. Структура и содержание дисциплины

### 2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зачетных единиц (72 часа), их распределение по видам работ представлено в таблице

Виды работ	Всего часов	Форма обучения			
		очная		очно-заочная	заочная
		7 семестр (часы)	– семестр (часы)	– семестр (часы)	– курс (часы)
<b>Контактная работа, в том числе:</b>	<b>22,2</b>	<b>22,2</b>			
<b>Аудиторные занятия (всего):</b>	<b>18</b>	<b>18</b>			
занятия лекционного типа	–	–			
лабораторные занятия	18	18			
<b>Иная контактная работа:</b>	<b>4,2</b>	<b>4,2</b>			
Контроль самостоятельной работы (КСР)	4	4			
Промежуточная аттестация (ИКР)	0,2	0,2			
<b>Самостоятельная работа, в том числе:</b>	<b>49,8</b>	<b>49,8</b>			
<i>Контрольная работа</i>	12	12			
<i>Самостоятельное изучение разделов, самоподготовка (проработка и повторение лекционного материала и материала учебников и учебных пособий, подготовка к лабораторным и практическим занятиям, коллоквиумам и т. д.)</i>	37,8	37,8			
Подготовка к текущему контролю	–	–			
<b>Контроль:</b>	<b>–</b>	<b>–</b>			
Подготовка к экзамену	–	–			
<b>Общая трудоёмкость</b>	<b>час.</b>	<b>72</b>	<b>72</b>		
	<b>в том числе контактная работа</b>	<b>22,2</b>	<b>22,2</b>		
	<b>зач. ед</b>	<b>2</b>	<b>2</b>		

### 2.2 Содержание дисциплины

Распределение видов учебной работы и их трудоёмкости по разделам дисциплины. Разделы (темы) дисциплины, изучаемые в 7 семестре (очная форма обучения)

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1.	Концепция объектно-ориентированного программирования (ООП).			4	12	
2.	Сравнительный анализ реализации ООП в языках C++, C#, Python и Fortran			8	23	
3.	Разработка компьютерных моделей на основе ООП для решения прикладных задач			4	8,4	
4.	Анализ эффективности ООП и вычисленный эксперимент			2	6,4	
	<i>ИТОГО по разделам дисциплины</i>	67,8		18	49,8	
	Контроль самостоятельной работы (КСР)	4				
	Промежуточная аттестация (ИКР)	0,2				
	Подготовка к текущему контролю	–				
	Общая трудоемкость по дисциплине	72				

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

## 2.3 Содержание разделов (тем) дисциплины

### 2.3.1 Занятия лекционного типа

Занятия лекционного типа не предусмотрены учебным планом

### 2.3.2 Занятия семинарского типа (практические / семинарские занятия/ лабораторные работы)

№	Наименование раздела (темы)	Тематика занятий/разбор	Форма текущего контроля
1.	Концепция объектно-ориентированного программирования (ООП).	Сущность объектно-ориентированного подхода в программировании. Цикл разработки программного обеспечения (ПО), назначение и содержание этапов. Роль анализа в процессе разработки программного обеспечения. Основные понятия объектно-ориентированного анализа. Методологии процедурного, структурного и объектно-ориентированного программирования, принципы ООП, абстракция, инкапсуляция, наследование и полиморфизм. Класс и объект, поля, методы и свойства.	РГЗ
2.	Сравнительный анализ реализации ООП в языках C++, C#, Python и Fortran	Технология применения объектно-ориентированных языков, их классификация и архитектура. Перегрузка операций. Преобразование типов. Реализации ООП в C++, C#, Python и Fortran, разработка классов и решения проблем наследования и полиморфизма.	РГЗ
3.	Разработка компьютерных моделей на основе ООП для решения прикладных задач	Математическое и компьютерное моделирование, алгоритмизация, дизайн компьютерного проекта, анализ кода. Шаблоны функций. Шаблоны классов. Исключения. Стандартная библиотека шаблонов. Стандартная библиотека классов для управления потоками. Методы и средства организации и программирования интерфейса.	РГЗ
4.	Анализ эффективности ООП и вычисленный эксперимент	Анализ ускорения и эффективности ООП. Вычислительные эксперименты.	РГЗ

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 выполнение контрольной работы (КР).

При изучении дисциплины применяется электронное обучение (проектор и ЭВМ), дистанционные образовательные технологии в соответствии с ФГОС ВО.

### 2.3.3 Примерная тематика курсовых работ (проектов)

Курсовые работы не предусмотрены учебным планом.

### 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	Проработка и повторение лекционного материала и материала учебников и учебных пособий	<i>Онлайн-лекции Национального исследовательского технологического университет «МИСиС» «C++ и основы ООП (видеоуроки)» [Электронный ресурс]: сайт. – URL: <a href="http://www.youtube.com/playlist?list=PLE9F6A65165CBC023">http://www.youtube.com/playlist?list=PLE9F6A65165CBC023</a></i>
2	Подготовка к лабораторным занятиям	<i>Методические указания по выполнению лабораторных работ, утвержденные на заседании кафедры теории функций факультета математики и компьютерных наук ФГБОУ ВО «КубГУ», протокол №7 от 18.04.2019 г.</i>
3	Подготовка к коллоквиуму	<i>Методические указания по выполнению лабораторных работ, утвержденные на заседании Совета экономического факультета ФГБОУ ВО «КубГУ», протокол №8 от 29.06.2017 г. Режим доступа: <a href="https://www.kubsu.ru/ru/econ/metodicheskie-ukazaniya">https://www.kubsu.ru/ru/econ/metodicheskie-ukazaniya</a></i>
4	Выполнение расчетно-графических заданий и контрольных работ	<i>Методические указания по выполнению расчетно-графических заданий, утвержденные на заседании Совета экономического факультета ФГБОУ ВО «КубГУ», протокол №8 от 29.06.2017 г. Режим доступа: <a href="http://docspace.kubsu.ru/docspace/handle/1/1125">http://docspace.kubsu.ru/docspace/handle/1/1125</a></i>

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,
- в форме аудиофайла,
- в печатной форме на языке Брайля.

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,
- в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

### 3. Образовательные технологии, применяемые при освоении дисциплины (модуля)

В ходе изучения дисциплины предусмотрено использование следующих образовательных технологий: лекции, практические занятия, подготовка письменных аналитических работ, самостоятельная работа студентов.

Компетентностный подход в рамках преподавания дисциплины реализуется в использовании интерактивных технологий и активных методов (проектных методик, разбора конкретных ситуаций) в сочетании с внеаудиторной работой.

Информационные технологии, применяемые при изучении дисциплины: использование информационных ресурсов, доступных в информационно-телекоммуникационной сети Интернет.

Адаптивные образовательные технологии, применяемые при изучении дисциплины – для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

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

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «*Объектно-ориентированное программирование и компьютерный инжиниринг*».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме *тестовых заданий, разноуровневых заданий, отчетов по индивидуальным и расчетно-графическим заданиям* и **промежуточной аттестации** в форме вопросов и заданий к зачету.

##### Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Код и наименование индикатора (в соответствии с п. 1.4)	Результаты обучения (в соответствии с п. 1.4)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	<i>ПК-4.4. Ориентируется в современных алгоритмах компьютерной математики и имеет практический опыт разработки программных модулей на основе механико-математических моделей</i>	Знает основные приемы программирования на основе объектно-ориентированных технологий для компьютерного моделирования при решении прикладных задач; основные концепции разработки вычислительных программ и приложения для компьютерного моделирования.	<i>РГЗ №1</i>	<i>Вопрос на зачете 1-10</i>
2	<i>ПК-4.4. Ориентируется в современных алгоритмах компьютерной математики и имеет практический опыт разработки программных модулей на основе механико-математических моделей</i>	Умеет реализовать компьютерные алгоритмы иерархией объектов для реализации алгоритмов решения численных задач математической физики и инженерии, проводить анализ производительности программ и анализ результатов расчета.	<i>РГЗ №2</i>	<i>Вопрос на зачете 10-22</i>
3	<i>ПК-4.4. Ориентируется в современных алгоритмах компьютерной математики и имеет практический опыт разработки программных модулей на основе механико-математических моделей</i>	Владеет навыками программирования и разработки эффективных иерархических алгоритмов, анализа структуры вычислительных алгоритмов, связями между задачами и промежуточными методами компьютерного проекта.	<i>РГЗ №3</i>	<i>Вопрос на зачете 22–33</i>

**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**  
**Примерный перечень вопросов и заданий**

**Расчетно-графические задания**

**Тип 1**

*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:
  6. 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.
  5. 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);

– Проекция одного вектора на другой;

– Определение угла между векторами (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 ^ -- 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.

## ***Тун 2***

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  $\omega$ :

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

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 работы.

### Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен/зачет)

- 1) Понятие объектно-ориентированное программирование (ООП). Отличия ООП от процедурного программирования. Понятие класса и экземпляра. Архитектура Фон-Неймана и параллельные компьютеры.
- 2) Классификация многопроцессорных систем: классификация Флинна; сильно и слабосвязанные процессоры; системы с разделяемой и распределенной памятью, кластеры 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 ти-пов. 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, индексы.
- 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. Оценка производительности параллельной программы: ускорение, эффективность.

## Критерии оценивания результатов обучения

*Критерии оценивания по зачету:*

*«зачтено»:* студент владеет теоретическими знаниями по данному разделу, знает вопросы основного учебно-программного материала, допускает незначительные ошибки; студент умеет обоснованно применять полученные знания в области объектно-ориентированного программирования к задачам прикладного и практического значения; справился с выполнением заданий, предусмотренных программой дисциплины.

*«не зачтено»:* материал не усвоен или усвоен частично, студент затрудняется реализовывать базовые методы при объектно-ориентированного программирования решении прикладных задач, довольно ограниченный объем выполненных заданий, предусмотренных программой дисциплины.

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 учебной литературы, информационных ресурсов и технологий

### 5.1. Учебная 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. Бабушкина, И. А. Практикум по объектно-ориентированному программированию: учебное пособие / И. А. Бабушкина, С. М. Окулов. — 5-е изд. — Москва: Лаборатория знаний, 2020. — 369 с. — ISBN 978-5-00101-780-6. — Текст: электронный // Лань: электронно-библиотечная система. — URL: <https://e.lanbook.com/book/135561>

3 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>

## 5.2. Периодическая литература

1. Журнал "Вычислительная механика сплошных сред" <http://www2.icmm.ru/journal/>
2. Базы данных компании «Ист Вью» <http://dlib.eastview.com>

## 5.3. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

### Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» <https://urait.ru/>
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» [www.biblioclub.ru](http://www.biblioclub.ru)
3. ЭБС «ЛАНЬ» <https://e.lanbook.com>

### Профессиональные базы данных:

1. Web of Science (WoS) <http://webofscience.com/>
2. Scopus <http://www.scopus.com/>
3. ScienceDirect [www.sciencedirect.com](http://www.sciencedirect.com)
4. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
5. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
6. "Лекториум ТВ" <http://www.lektorium.tv/>
7. Университетская информационная система РОССИЯ <http://uisrussia.msu.ru>

### Информационные справочные системы:

1. Консультант Плюс - справочная правовая система (доступ по локальной сети с компьютеров библиотеки)

### Ресурсы свободного доступа:

1. КиберЛенинка (<http://cyberleninka.ru/>);
2. Курсы ведущих вузов России" <http://www.openedu.ru/>;
3. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
4. Онлайн-курсы и сертификаты от ведущих вузов мира <https://ru.coursera.org/>.
5. Российская система прочностного анализа на основе метода спектральных конечных элементов Fidesys <http://www.cae-fidesys.com/ru/about/info>

### Собственные электронные образовательные и информационные ресурсы

#### КубГУ:

1. Среда модульного динамического обучения <http://moodle.kubsu.ru>
2. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://mschool.kubsu.ru/>
3. Библиотека информационных ресурсов кафедры информационных образовательных технологий [http://mschool.kubsu.ru](http://mschool.kubsu.ru;);
4. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>

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

– *Общие рекомендации по самостоятельной работе обучающихся.*

*Самостоятельная работа обучающихся выполняется по заданию и при методическом руководстве преподавателя, но без его непосредственного участия. Самостоятельная работа подразделяется на самостоятельную работу на аудиторных*

занятиях и на внеаудиторную самостоятельную работу. Самостоятельная работа обучающихся включает как полностью самостоятельное освоение отдельных тем (разделов) дисциплины, так и проработку тем (разделов), осваиваемых во время аудиторной работы. Во время самостоятельной работы обучающиеся читают и конспектируют учебную, научную и справочную литературу, выполняют задания, направленные на закрепление знаний и отработку умений и навыков, готовятся к текущему и промежуточному контролю по дисциплине.

Организация самостоятельной работы обучающихся регламентируется нормативными документами, учебно-методической литературой и электронными образовательными ресурсами, включая:

Порядок организации и осуществления образовательной деятельности по образовательным программам высшего образования – программам бакалавриата, программам специалитета, программам магистратуры (утвержден приказом Министерства образования и науки Российской Федерации от 5 апреля 2017 года №301).

Письмо Министерства образования Российской Федерации №14-55-996ин/15 от 27 ноября 2002 г. "Об активизации самостоятельной работы студентов высших учебных заведений".

Положение о самостоятельной работе студентов (утверждено приказом № 272 КубГУ от 03 марта 2016 г.).

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

Индивидуальные консультации по предмету являются важным фактором, способствующим индивидуализации обучения и установлению воспитательного контакта между преподавателем и обучающимся инвалидом или лицом с ограниченными возможностями здоровья.

## 7. Материально-техническое обеспечение по дисциплине (модулю)

Наименование специальных помещений	Оснащенность специальных помещений	Перечень лицензионного программного обеспечения
Учебные аудитории для проведения занятий лекционного типа		
Учебные аудитории для проведения занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Мебель: учебная мебель Технические средства обучения: Компьютеры	Microsoft Windows Microsoft Office Professional Plus Fortran, C++, C# MatLab

Для самостоятельной работы обучающихся предусмотрены помещения, укомплектованные специализированной мебелью, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду университета.

Наименование помещений для самостоятельной работы обучающихся	Оснащенность помещений для самостоятельной работы обучающихся	Перечень лицензионного программного обеспечения
Помещение для самостоятельной работы обучающихся (читальный зал Научной библиотеки)	Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к	

	<p>информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)</p>	
<p>Помещение для самостоятельной работы обучающихся (ИС 6, ИС 7)</p>	<p>Мебель: учебная мебель Комплект специализированной мебели: компьютерные столы Оборудование: компьютерная техника с подключением к информационно-коммуникационной сети «Интернет» и доступом в электронную информационно-образовательную среду образовательной организации, веб-камеры, коммуникационное оборудование, обеспечивающее доступ к сети интернет (проводное соединение и беспроводное соединение по технологии Wi-Fi)</p>	