

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

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



«30» мая 2025

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ
Б1.О.25«Объектно-ориентированное программирование»

Направление подготовки 09.03.03 Прикладная информатика

Направленность (профиль) Прикладная информатика в экономике

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Объектно-ориентированное программирование» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 09.03.03 Прикладная информатика.

Программу составил(и):

А.В. Ковалева

И.О. Фамилия, должность, ученая степень, ученое звание


подпись

И.О. Фамилия, должность, ученая степень, ученое звание

подпись

Рабочая программа дисциплины «Объектно-ориентированное программирование» утверждена на заседании кафедры информационных технологий протокол №15 от «14» мая 2025г.

Заведующий кафедрой (разработчика)

В. В. Подколзин


подпись

Рабочая программа обсуждена на заседании кафедры анализа данных и искусственного интеллекта протокол №13 от «20» мая 2025 г.

Заведующий кафедрой (выпускающей)

А. В. Коваленко


подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол №4 от «23» мая 2025 г.

Председатель УМК факультета

А. В. Коваленко


подпись

Рецензенты:

Бегларян М. Е., Проректор по учебной работе, Краснодарский кооперативный институт (филиал) АНО ВО Центросоюза РФ «Российский университет кооперации»

Рубцов Сергей Евгеньевич, кандидат физико-математических наук, доцент кафедры математического моделирования ФГБОУ ВО «КубГУ»

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Цель дисциплины – изучение студентами объектно-ориентированного подхода в программировании. Все практические примеры рассматриваются на языке программирования Java. Воспитательная цель: формирование свободного и творческого подхода к программированию на современных языках высокого уровня, интереса к наблюдению за тенденциями и новостями в области средств разработки программного обеспечения. Содержательное наполнение дисциплины обусловлено общими задачами в подготовке бакалавра. Научной основой для построения программы данной дисциплины является теоретико-прагматический подход в обучении.

1.2 Задачи дисциплины

Основные задачи курса на основе системного подхода:

- закрепление навыков алгоритмизации и программирования на основе углубленного изучения объектно-ориентированного подхода в программировании;
- знакомство с принципами инкапсуляции, наследования и полиморфизма;
- обучение созданию мультиплатформенных приложений.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование» относится к «Обязательная часть» Блока 1 «Дисциплины (модули)» учебного плана.

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:

ОПК-7 **Способен разрабатывать алгоритмы и программы, пригодные для практического применения**

ИД-1.ОПК-7 *Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности*

Знать *Основы программирования*

Языки программирования и работы с базами данных

Методы анализа и обобщения отечественного и международного опыта в соответствующей области исследований

Методологии разработки программного обеспечения и технологии программирования

Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения

Методы и средства проектирования программного обеспечения

Методы и средства проектирования баз данных

Методы и средства проектирования программных интерфейсов

Уметь	<i>Проводить анализ исполнения требований</i>
	<i>Вырабатывать варианты реализации требований</i>
	<i>Использовать существующие типовые решения и шаблоны проектирования программного обеспечения</i>
	<i>Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов</i>
	<i>Вырабатывать варианты реализации требований</i>
Владеть	<i>Разработка, изменение и согласование архитектуры программного обеспечения с заинтересованными лицами</i>
	<i>Проектирование структур данных</i>
	<i>Проектирование программных интерфейсов</i>
	<i>Анализ возможностей реализации требований к программному обеспечению</i>
	<i>Оценка времени и трудоемкости реализации требований к программному обеспечению</i>
ИД- 2.ОПК-7 Знать	<i>Разработка структуры программного кода ИС</i>
	<i>Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения</i>
	<i>Возможности существующей программно-технической архитектуры</i>
	<i>Возможности современных и перспективных средств разработки программных продуктов, технических средств</i>
	<i>Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения</i>
Уметь	<i>Проводить оценку и обоснование рекомендуемых решений</i>
	<i>Верифицировать структуру программного кода</i>
	<i>Оформлять результаты научно-исследовательских и опытно-конструкторских работ</i>
	<i>Проводить анализ исполнения требований</i>

Владеть Устранение обнаруженных несоответствий

Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС

Разработка технической документации на компьютерное программное обеспечение с использованием существующих стандартов

Составление новых тестовых случаев и повторение тестирования при необходимости

Тестирование программного обеспечения

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)					
			3					
Контактная работа, в том числе:		72,3	72,3					
Аудиторные занятия (всего):		68	68					
Занятия лекционного типа		34	34					
Лабораторные занятия		34	34					
Занятия семинарского типа (семинары, практические занятия)								
Иная контактная работа:		4,3	4,3					
Контроль самостоятельной работы (КСР)		4	4					
Промежуточная аттестация (ИКР)		0,3	0,3					
Самостоятельная работа, в том числе:		36	36					
Проработка учебного (теоретического) материала		10	10					
Выполнение индивидуальных заданий		10	10					
Подготовка к текущему контролю		16	16					
Контроль:		35,7	35,7					
Подготовка к экзамену		35,7	35,7					
Общая трудоемкость	час.	144	144					
	в том числе контактная работа	72,3	72,3					
	зач. ед	4	4					

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.
Разделы (темы) дисциплины, изучаемые в 3 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	2	3	4	5	6	7
1.	Основные конструкции ЯП Java.	6	2		2	2
2.	Основы ООП.	6	2		2	2
3.	Методы. Модификаторы доступа.	6	2		2	2
4.	Наследование. Глобальный суперкласс Object.	6	2		2	2
5.	Полиморфизм.	6	2		2	2
6.	Абстрактные классы.	6	2		2	2
7.	Интерфейсы.	12	4		4	4
8.	Исключения.	12	4		4	4
9.	Дженерики.	6	2		2	2
10.	Коллекции.	12	4		4	4
11.	Лямбда-выражения.	6	2		2	2
12.	Функциональные интерфейсы.	6	2		2	2
13.	Основы создания оконных приложений.	8	2		2	4
14.	Обработка событий.	6	2		2	2
ИТОГО по разделам дисциплины		104	34		34	
Контроль самостоятельной работы (КСР)		4				
Промежуточная аттестация (ИКР)		0,3				
Подготовка к текущему контролю		35,7				
Общая трудоемкость по дисциплине		144				

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

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Основные конструкции ЯП Java.	Примитивные типы данных, основные операции, основные операторы, функция main	К
2.	Основы ООП.	Понятие класса, три базовых свойства ООП: инкапсуляция, наследование, полиморфизм, конструкторы, создание экземпляров объектного типа	К
3.	Методы. Модификаторы доступа.	Структура объявления методов. Совмещение методов. Модификаторы доступа. Статические методы.	К
4.	Наследование. Глобальный суперкласс Object.	Наследование. Ключевое слово super. Замещение методов. Использование final методов. Глобальный суперкласс Object и его методы equals, hashCode, clone, getClass, toString	К
5.	Полиморфизм.	Свойство полиморфизма. Динамическое назначение методов.	К
6.	Абстрактные классы.	Абстрактные классы и абстрактные методы. Ограничения при использовании абстрактных классов. Внутренние классы.	К
7.	Интерфейсы.	Интерфейсы. Реализация интерфейсов классом. Наследование интерфейсов.	К

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
8.	Исключения.	Иерархия классов-исключений в Java. Блоки try, catch, finally. Множественный catch. Ручная генерация исключений (throw), ключевое слово throws. Создание собственных классов исключений.	К
9.	Дженерики.	Классы-оболочки. Параметризованные типы. Параметризованные классы. Параметризованные методы. Параметризованные интерфейсы. Ограничения при использовании дженериков.	К
10.	Коллекции.	Иерархия интерфейсов коллекций в Java. Интерфейс List и классы, его реализующие, интерфейс Set и классы, его реализующие, интерфейс Queue и классы, его реализующие, интерфейс Map и классы, его реализующие	К
11.	Лямбда-выражения.	Лямбда-выражение, лямбда-оператор, функциональный интерфейс, передача параметров в лямбда-выражение, обобщенный функциональный интерфейс	К
12.	Функциональные интерфейсы.	Функциональные интерфейсы Predicate, Consumer, Function, Supplier, UnaryOperator, BinaryOperator	К
13.	Основы создания оконных приложений.	Пакеты AWT и Swing, компоненты JFrame, JPanel, JButton, JTextField, JTextArea, добавление компонентов на фрейм, позиционирование компонентов основные методы класса Component	К
14.	Обработка событий.	Общая схема обработки событий в Java, интерфейсы-слушатели: ActionListener, WindowListener, KeyListener, MouseListener, классы-адаптеры	К

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Основные конструкции ЯП Java.	Изучение среды Eclipse/NetBeans. Создание и запуск простого приложения. Работа с массивами и матрицами. Средства ввода/вывода	РЗ
2.	Основы ООП.	Создание простых классов и работы с экземплярами классов.	РЗ
3.	Методы. Модификаторы доступа.	Создание классов с совмещенными методами, практическое рассмотрение различий между модификаторами доступа.	РЗ
4.	Наследование. Глобальный суперкласс Object.	Создание иерархий объектных типов, использование ключевого слова super, замещение методов в подклассах	РЗ
5.	Полиморфизм.	Использование объектных ссылок при передаче параметров в методы	РЗ
6.	Абстрактные классы.	Создание программ с иерархиями объектных типов на основе абстрактных классов. Пример полиморфизма на основе создания массива из различных объектных типов	РЗ

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
7.	Интерфейсы.	Создание интерфейсов и классов, их реализующих	РЗ
8.	Исключения.	Работа со встроенными исключениями языка Java. Речная генерация исключения. Создание собственного класса исключения	РЗ
9.	Дженерики.	Создание программ на основе параметризованных классов.	РЗ
10.	Коллекции.	Работа с классами ArrayList, LinkedList, HashSet, PriorityQueue, HashMap	РЗ
11.	Лямбда-выражения.	Создание функционального интерфейса и его реализация с использованием лямбда-выражений	РЗ
12.	Функциональные интерфейсы.	Реализация интерфейсов Predicate, Consumer, Function, Supplier, UnaryOperator, BinaryOperator	РЗ
13.	Основы создания оконных приложений.	Создание оконного приложения с использованием классов JFrame и JPanel, работа с элементами управления JButton, JTextField, JTextArea, рисование на панели с использованием методов класса Graphics2D	РЗ
14.	Обработка событий.	Создание обработчиков событий и их связь с элементами управления окна	РЗ

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

Не предусмотрены

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	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

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

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) – расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

Комплексное использование в учебном процессе всех вышеперечисленных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения

результатов индивидуальной работы членов команды с делением полномочий и ответственности;

- проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

- анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

- развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
3	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	8
Итого			8

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

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

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

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.

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

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

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

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме заданий по темам, вопросов к коллоквиуму и **промежуточной аттестации** в форме вопросов и заданий к экзамену.

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

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

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

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

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

Для лиц с нарушениями зрения:

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

Для лиц с нарушениями слуха:

- в печатной форме,
- в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

- в печатной форме,
- в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины*	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Основные конструкции ЯП Java.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 1	Вопрос на экзамене 1-7 Задача к экзамену 1-4
2	Основы ООП.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 1 Задания для самостоятельной работы 1-3	Вопрос на экзамене 8-9 Задача к экзамену 1-4
3	Методы. Модификаторы доступа.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 1 Задания для самостоятельной работы 1-3	Вопрос на экзамене 10 Задача к экзамену 1-4
4	Наследование. Глобальный суперкласс Object.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 5	Вопрос на экзамене 11, 18

			Задания для самостоятельной работы 4-7	
5	Полиморфизм.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 5 Задания для самостоятельной работы 4-7	Вопрос на экзамене 12
6	Абстрактные классы.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 2, 6	Вопрос на экзамене 13 Задача к экзамену 5
7	Интерфейсы.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 7 Задания для самостоятельной работы 8-12	Вопрос на экзамене 14
8	Исключения.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 7 Задания для самостоятельной работы 8-12	Вопрос на экзамене 15-16
9	Дженерики.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 8 Задания для самостоятельной работы 17	Вопрос на экзамене 19-23
10	Коллекции.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 4 Задания для самостоятельной работы 18-21	Вопрос на экзамене 24-30
11	Лямбда-выражения.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Задания для самостоятельной работы 8-9	Вопрос на экзамене 31
12	Функциональные интерфейсы.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Задания для самостоятельной работы 8-9	Вопрос на экзамене 32
13	Основы создания оконных приложений.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 3 Задания для самостоятельной работы 13-16	Вопрос на экзамене 33
14	Обработка событий.	<i>ИД-1.ОПК-7</i> <i>ИД-2.ОПК-7</i>	Типовые контрольные задания 3 Задания для самостоятельной работы 13-16	Вопрос на экзамене 34

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **пороговому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **удовлетворительно /зачтено**):

ОПК-7	Способен разрабатывать алгоритмы и программы, пригодные для практического применения
ИД-1.ОПК-7	<i>Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности</i>
Знать	<i>Основы программирования</i> <i>Языки программирования и работы с базами данных</i> <i>Методы анализа и обобщения отечественного и международного опыта в соответствующей области исследований</i> <i>Методологии разработки программного обеспечения и технологии программирования</i> <i>Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения</i> <i>Методы и средства проектирования программного обеспечения</i> <i>Методы и средства проектирования баз данных</i> <i>Методы и средства проектирования программных интерфейсов</i>
Уметь	<i>Проводить анализ исполнения требований</i> <i>Вырабатывать варианты реализации требований</i> <i>Использовать существующие типовые решения и шаблоны проектирования программного обеспечения</i> <i>Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов</i> <i>Вырабатывать варианты реализации требований</i>
Владеть	<i>Разработка, изменение и согласование архитектуры программного обеспечения с заинтересованными лицами</i> <i>Проектирование структур данных</i> <i>Проектирование программных интерфейсов</i> <i>Анализ возможностей реализации требований к программному обеспечению</i> <i>Оценка времени и трудоемкости реализации требований к программному обеспечению</i> <i>Разработка структуры программного кода ИС</i>

ИД-2.ОПК-7
Знать *Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения*
Возможности существующей программно-технической архитектуры

Возможности современных и перспективных средств разработки программных продуктов, технических средств

Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения

Уметь *Проводить оценку и обоснование рекомендуемых решений*

Верифицировать структуру программного кода

Оформлять результаты научно-исследовательских и опытно-конструкторских работ

Проводить анализ исполнения требований

Владеть *Устранение обнаруженных несоответствий*

Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС

Разработка технической документации на компьютерное программное обеспечение с использованием существующих стандартов

Составление новых тестовых случаев и повторение тестирования при необходимости

Тестирование программного обеспечения

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **хорошо /зачтено**):

ОПК-7 **Способен разрабатывать алгоритмы и программы, пригодные для практического применения**

ИД-1.ОПК-7 *Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности*

Знать *Основы программирования на C++*

Языки программирования и работы с базами данных на высоком уровне

Методы анализа и обобщения отечественного и международного опыта в соответствующей области исследований

Методологии разработки программного обеспечения и технологии программирования

Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения

Методы и средства проектирования программного обеспечения

Методы и средства проектирования баз данных

Методы и средства проектирования программных интерфейсов

Уметь

Проводить анализ исполнения требований на высоком уровне

Вырабатывать варианты реализации требований

Использовать существующие типовые решения и шаблоны проектирования программного обеспечения

Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов

Вырабатывать варианты реализации требований

Владеть

Разработка, изменение и согласование архитектуры программного обеспечения с заинтересованными лицами

Проектирование структур данных

Проектирование программных интерфейсов

Анализ возможностей реализации требований к программному обеспечению

Оценка времени и трудоемкости реализации требований к программному обеспечению

Разработка структуры программного кода ИС

**ИД-
2.ОПК-7
Знать**

Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения

Возможности существующей программно-технической архитектуры

Возможности современных и перспективных средств разработки программных продуктов, технических средств

Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения

Уметь *Проводить оценку и обоснование рекомендуемых решений на высоком уровне*

Верифицировать структуру программного кода

Оформлять результаты научно-исследовательских и опытно-конструкторских работ

Проводить анализ исполнения требований на высоком уровне

Владеть *Устранение обнаруженных несоответствий*

Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач

Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС

Разработка технической документации на компьютерное программное обеспечение с использованием существующих стандартов

Составление новых тестовых случаев и повторение тестирования при необходимости

Тестирование программного обеспечения

Соответствие **продвинутому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **отлично /зачтено**):

ОПК-7 **Способен разрабатывать алгоритмы и программы, пригодные для практического применения**

ИД-1.ОПК-7 *Аргументировано применяет методы проектирования, разработки и реализации программных продуктов и программных комплексов в различных областях человеческой деятельности*

Знать *Основы программирования на C++*

Языки программирования и работы с базами данных на высоком уровне

Методы анализа и обобщения отечественного и международного опыта в соответствующей области исследований

Методологии разработки программного обеспечения и технологии программирования

Принципы построения архитектуры программного обеспечения и виды архитектуры программного обеспечения

Методы и средства проектирования программного обеспечения

Современные методы и средства проектирования баз данных

Методы и средства проектирования программных интерфейсов

Уметь

Проводить анализ исполнения требований на высоком уровне

Вырабатывать варианты реализации требований

Использовать существующие типовые решения и шаблоны проектирования программного обеспечения

Применять методы и средства проектирования программного обеспечения, структур данных, баз данных, программных интерфейсов

Вырабатывать варианты реализации требований

Владеть

Разработка, изменение и согласование архитектуры программного обеспечения с заинтересованными лицами

Проектирование структур данных на высоком уровне

Проектирование программных интерфейсов

Анализ возможностей реализации требований к программному обеспечению

Аргументированная оценка времени и трудоемкости реализации требований к программному обеспечению

Разработка структуры программного кода ИС

**ИД-
2.ОПК-7
Знать**

Использует инструментальные, программные и аппаратные средства измерений для оценки качества программного обеспечения

Возможности существующей программно-технической архитектуры

Возможности современных и перспективных средств разработки программных продуктов, технических средств

Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке программного обеспечения

Уметь	<i>Проводить оценку и обоснование рекомендуемых решений на высоком уровне</i> <i>Верифицировать структуру программного кода на высоком уровне</i> <i>Оформлять результаты научно-исследовательских и опытно-конструкторских работ</i> <i>Проводить анализ исполнения требований на высоком уровне</i>
Владеть	<i>Устранение обнаруженных несоответствий</i> <i>Деятельность, направленная на решение задач аналитического характера, предполагающих выбор и многообразие актуальных способов решения задач</i> <i>Верификация структуры программного кода ИС относительно архитектуры ИС и требований заказчика к ИС</i> <i>Разработка технической документации на компьютерное программное обеспечение с использованием существующих стандартов</i> <i>Составление новых тестовых случаев и повторение тестирования при необходимости</i> <i>Тестирование программного обеспечения</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

Типовые контрольные задания:

1. Определить класс `train`, содержащий поля: название пункта назначения, номер поезда, время отправления. Ввести данные в массив из пяти элементов типа `train`, упорядочить элементы по номерам поездов. Добавить возможность вывода информации о поезде, номер которого введен пользователем. Добавить возможность сортировки массив по пункту назначения.
2. Создать абстрактный класс Телефонный справочник с методами, позволяющими вывести на экран информацию о записях в телефонном справочнике, а также определить соответствие записи критерию поиска. Создать производные классы: Персона (фамилия, адрес, номер телефона), Организация (название, адрес, телефон, факс, контактное лицо), Друг (фамилия, адрес, номер телефона, дата рождения) со своими методами вывода информации на экран, и определения соответствия искомому типу. Создать массив из n элементов различных классов, вывести полную информацию из массива на экран, а также организовать поиск в массиве по фамилии.
3. Решить с использованием формы: определить класс для работы с комплексными числами. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел.

4. Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям.
5. Создать базовый класс – «работник» и производные классы – «служащий с почасовой оплатой», «служащий в штате» и «служащий с процентной ставкой». Определить в каждом классе функцию начисления зарплаты.
6. Создать абстрактный класс Фигура с функцией – площадь и периметр. Реализовать интерфейс в классах: «круг», «прямоугольный треугольник», «трапеция» со своими функциями площади и периметра. Для проверки определить массив ссылок на интерфейс, которым присваиваются адреса различных объектов.
7. Определить интерфейс Тело с функциями вычисления объема и площади поверхности. Реализовать этот интерфейс в классах Конус и Пирамида. Для проверки определить массив ссылок на интерфейс, которым присваиваются адреса различных объектов
8. Определить параметризованный класс Вектор с методами добавления элемента, удаления заданного элемента, замены порядка элементов на обратный. Создать экземпляры этого класса для параметра типа Double и String

Типовые задания для самостоятельной работы

1. Описать класс параллелепипед. Предусмотреть методы создания объектов, перемещения, изменения размеров, вычисления высоты, нахождения центра тяжести. Написать программу, демонстрирующую работу с этим классом.
2. Описать класс «телефонный справочник». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку, добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом.
3. Описать класс «список». Предусмотреть возможность добавления и удаления элементов в начало, конец списка, в указанную позицию, сортировку по убыванию и возрастанию, подсчет количества элементов в списке, вывод на экран части списка с указанием границ вывода.
4. Определить класс «Автомобиль», экземпляры которого имеют поля – марка, мощность, объем двигателя. Определить функции назначения и изменения полей и функцию вывода на экран. Определить производный класс – такси, экземпляры которого имеют поле – тариф. Определить функцию вывода на экран. Определить функцию переназначения тарифа и функцию расчета стоимости поездки.
5. Определить класс «Жидкость», экземпляры которого имеют поля – название, плотность. Определить функции назначения и изменения полей и функцию вывода на экран. Определить производный класс – раствор кислоты, экземпляры которого имеют поле – концентрация. Определить функцию вывода на экран. Определить функцию переназначения концентрации.
6. Определить класс «Автомобиль», экземпляры которого имеют поля – марка, мощность, объем двигателя. Определить функции назначения и изменения полей и функцию вывода на экран. Определить производный класс – грузовой автомобиль, экземпляры которого имеют поле – грузоподъемность. Определить функцию вывода на экран. Определить функцию переназначения грузоподъемности и функцию, определяющую возможность перевозки груза.
7. Определить класс «Работник», и производные классы – «Служащий с почасовой оплатой», «Служащий в штате», «Служащий с процентной ставкой». Определить функции назначения и изменения полей и функцию вывода на экран и функцию назначения заработной платы.

8. Определить интерфейс Root с методами вычисления вычисления корней и вывода результата на экран. Реализовать интерфейс в классах Linear и Square.
 9. Определить интерфейс Function с методами вычисления функции в заданной точке и вывода результата на экран. Реализовать интерфейс в классах Ellipse и Hyperbola.
 10. Определить интерфейс Pair с методами арифметическими операциями. Реализовать интерфейс в классах Complex и Rational (рациональная дробь).
 11. Определить интерфейс Triad с методами увеличения на 1. Реализовать интерфейс в классах Date и Time.
 12. Определить интерфейс Pair с методами арифметическими операциями. Реализовать интерфейс в классах Money и Fraction (дробное число).
 13. Нарисовать ромб, заданный стороной и двумя углами, треугольник, заданный координатами вершин, и эллипс. Выделить область пересечения этих фигур.
 14. Нарисовать ромб, заданный стороной и двумя углами, прямоугольник, заданный координатами концов своей диагонали, и окружность, заданную координатами центра и радиусом. Выделить область пересечения этих фигур.
 15. Нарисовать произвольный восьмиугольник, треугольник, заданный координатами своих вершин, и эллипс. Выделить область пересечения этих фигур.
 16. Нарисовать два треугольника, заданных координатами своих вершин. В треугольнике с большей площадью нарисовать медиану меньшего угла. В треугольнике с меньшей площадью нарисовать биссектрисы всех углов.
 17. Написать программу, демонстрирующую работу с объектами двух типов, T1 и T2, для чего создать систему соответствующих классов. Каждый объект должен иметь идентификатор (в виде произвольной строки символов) и одно или несколько полей для хранения состояния (текущего значения) объекта.
- Перечень типов объектов: Треугольник (Triangle), Квадрат (Quad), Прямоугольник (Rectangle), Четырехугольник (Tetragon), Пятиугольник (Pentagon).

Перечень дополнительных методов:

- move() переместить объект на плоскости
- compare (T1 t1, T2 t2) сравнение объектов по площади
- isIntersect(T1 t1, T2 t2) определяет факт пересечения объектов
- isInclude(T1 t1, T2 t2) определяет факт включения t2 в t1.

Вариант	T1	T2	Методы
1	Triangle	Quad	Move, compare
2	Quad	Pentagon	Move, isIntersect
3	Triangle	Rectangle	Move, compare

18. Написать программу для моделирования Т-образного сортировочного узла на железной дороге с использованием класса, реализующего интерфейс Deque. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность ввода исходных данных с клавиатуры и из файла.

19. Написать программу, моделирующую управление каталогом в файловой системе. Для каждого файла в каталоге содержатся следующие сведения: имя файла, дата создания, количество обращений к файлу. Программа должна обеспечивать:

- начальное формирование каталога файлов;
- вывод каталога файлов;
- удаление файлов, дата создания которых раньше заданной;

- выборку файла с наибольшим количеством обращений.

Выбор конкретного действия должен осуществляться с помощью меню. Для представления каталога использовать класс, реализующий интерфейс List.

20. Написать программу моделирования работы автобусного парка. Сведения о каждом автобусе содержат: номер автобуса, фамилию и инициалы водителя, номер маршрута. Обеспечить выбор с помощью меню и выполнение следующих функций:

- начальное формирование данных о всех автобусах в парке в виде списка (ввод с клавиатуры или из файла);
- имитация выезда автобуса из парка: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся в парке, и записывает эти данные в список автобусов, находящихся на маршруте;
- имитация въезда автобуса в парк: вводится номер автобуса; программа удаляет данные об этом автобусе из списка автобусов, находящихся на маршруте, и записывает эти данные в список автобусов, находящихся в парке;
- вывод сведений об автобусах, находящихся в парке, и об автобусах, находящихся на маршруте.

Для представления необходимых списков использовать класс, реализующий интерфейс List.

21. Написать программу учета заявок на авиабилеты. Каждая заявка содержит: пункт назначения, номер рейса, фамилию и инициалы пассажира, желаемую дату вылета. Обеспечить выбор с помощью меню и выполнение следующих функций:

- добавление заявок в список;
- удаление заявок;
- вывод заявок по заданному номеру рейса и дате вылета;
- вывод всех заявок.

Для хранения данных использовать класс, реализующий интерфейс List.

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

Вопросы для подготовки к экзамену

1. Характерные особенности языка Java.
2. Типы данных.
3. Объявление переменных и констант.
4. Массивы.
5. Основные операции.
6. Основные операторы.
7. Ввод/вывод в Java.
8. Классы (основные понятия).
9. Классы (конструктор, совмещенные методы).
10. Модификаторы доступа.
11. Наследование.
12. Полиморфизм, динамическое назначение методов.
13. Абстрактные классы.

14. Интерфейсы.
15. Внутренние классы.
16. Исключения (try, catch, finally).
17. Исключения (throw, throws).
18. Класс Object.
19. Классы-оболочки.
20. Класс Integer.
21. Дженерики (параметризованные классы).
22. Параметризованные методы.
23. Ограничения при использовании обобщений.
24. Коллекции.
25. Интерфейс Iterator.
26. Интерфейс ListIterator;
27. Интерфейс List, и классы его реализующие.
28. Интерфейс Set, и классы его реализующие.
29. Интерфейс Queue, и классы его реализующие.
30. Интерфейс Map.
31. Лямбда-выражения.
32. Функциональные интерфейсы.
33. Основы создания оконных приложений.
34. Схема обработки событий.

Типовые задачи к экзамену

1. Создать класс точка, предусмотреть в нем несколько конструкторов, методы определения координатной четверти, которой принадлежит точка, перемещения точки, метод проверяющий являются ли две точки симметричными относительно начала координат, метод, вычисляющий расстояние между двумя точками.
2. Создать класс Равнобедренная трапеция, члены класса: координаты 4-х точек. Предусмотреть в классе конструктор и методы: проверка, является ли фигура равнобедренной трапецией; вычисления и вывода сведений о фигуре: длины сторон, периметр, площадь. В функции main продемонстрировать работу с классом: дано N трапеций, найти количество трапеций, у которых площадь больше средней площади.
3. Создать класс Angle для работы с углами на плоскости, задаваемые в градусах и минутах. Обязательно должны быть реализованы перевод в радианы, приведение к диапазону 0-360, увеличение и уменьшение угла на заданную величину, получение синуса, сравнение углов.
4. Создать класс Date для работы с датами в формате «год.месяц.день». Дата представляется структурой с тремя полями. Класс должен включать не менее трех методов инициализации: числами, строкой и датой. Обязательными операциями являются: вычисление даты через заданное количество дней, вычитание заданного количества дней из даты, определение високосности года, присвоение и получение отдельных частей, сравнение дат, вычисление количества дней между датами.
5. Создать абстрактный класс Figure с методами вычисления площади и периметра. Создать производные классы Rectangle (прямоугольник), Circle (круг), Trapezium (трапеция) со своими методами вычисления площади и периметра. Самостоятельно определить какие поля можно задать в базовом классе, а какие - в производных.

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

Методические рекомендации, определяющие процедуры оценивания выполнения самостоятельных заданий:

Задание считается выполненным при выполнении следующих условий:

- предоставлен исходный код на Java в среде Eclipse / NetBeans;
- продемонстрирована работоспособность приложения;
- студент понимает исходный код и отвечает на вопросы по его организации.

Методические рекомендации, определяющие процедуры оценивания контрольных заданий:

Задание считается выполненным, если предоставлен исходный текст программы, который содержит не более 3 синтаксических ошибок, логика работы программы эффективна, программа работает на любых входных данных.

Методические рекомендации, определяющие процедуры оценивания на экзамене:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является зачет и экзамен. Студенты допускаются к сдаче экзамена, только после получения зачета по дисциплине. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом

ФОС промежуточной аттестации состоит из вопросов к экзамену, задач по дисциплине и результатам текущего контроля.

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

Форма проведения экзамена: устно.

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

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

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы экзамена.

Критерии оценки:

оценка «неудовлетворительно» выставляется в случае выполнения одного из условий:

- непонимание сущности излагаемых вопросов, грубые ошибки в ответе, неуверенные и неточные ответы на дополнительные вопросы;
- предоставлен не работоспособный код для экзаменационной задачи;

оценка «удовлетворительно» в случае выполнения условий:

- частично ответил на два вопроса билета или достаточно полно ответил хотя бы на один вопрос;
- предоставлен код, логика работы которого верна, но содержит более 5 синтаксических ошибок;

оценка «хорошо» в случае выполнения условий:

- достаточно полно ответил на два вопроса билета;
- даны частичные ответы на дополнительные вопросы;

- предоставлен код, логика работы которого верна, но содержит не более 5 синтаксических ошибок;

оценка «отлично» в случае выполнения условий:

- глубокие исчерпывающие знания по вопросам билета;
- даны правильные и конкретные ответы на дополнительные вопросы;
- предоставлен работоспособный код

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 (модуля)

5.1 Основная литература:

1. Тузовский, А. Ф. Объектно-ориентированное программирование : учебное пособие для вузов / А. Ф. Тузовский. — Москва : Издательство Юрайт, 2021. — 206 с. — (Высшее образование). — ISBN 978-5-534-00849-4. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/470223>

2. Бабушкина, И. А. Практикум по объектно-ориентированному программированию : [16+] / И. А. Бабушкина, С. М. Окулов. — 5-е изд., электрон. — Москва : Лаборатория знаний, 2020. — 369 с. : ил. — Режим доступа: по подписке. — URL: <https://biblioclub.ru/index.php?page=book&id=221691> (дата обращения: 15.08.2021). — Библиогр.: с. 358. — ISBN 978-5-00101-780-6. — Текст : электронный.

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. Синица С.Г. Уварова А.В. Программирование на Java: учебное пособие. КубГУ, Краснодар, 2016 г. ISBN: 978-5-8209-1215-3

2. Николаев, Е.И. Объектно-ориентированное программирование : учебное пособие/ Е.И. Николаев ; Министерство образования и науки Российской Федерации, Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Северо-Кавказский федеральный университет». - Ставрополь : СКФУ, 2015. - 225 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=458133>

5.3. Периодические издания:

1. Базы данных компании «Ист Вью» <http://dlib.eastview.com>
2. Электронная библиотека GREBENNIKON.RU <https://grebennikon.ru/>

5.4. 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» <http://www.biblioclub.ru/>
3. ЭБС «BOOK.ru» <https://www.book.ru>
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» <https://e.lanbook.com>

Профессиональные базы данных

1. Scopus <http://www.scopus.com/>
2. ScienceDirect <https://www.sciencedirect.com/>
3. Журналы издательства Wiley <https://onlinelibrary.wiley.com/>
4. Научная электронная библиотека (НЭБ) <http://www.elibrary.ru/>
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН <http://archive.neicon.ru>
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) <https://rusneb.ru/>
7. Президентская библиотека им. Б.Н. Ельцина <https://www.prilib.ru/>
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) <https://www.ccdc.cam.ac.uk/structures/>
9. Springer Journals: <https://link.springer.com/>
10. Springer Journals Archive: <https://link.springer.com/>
11. Nature Journals: <https://www.nature.com/>
12. Springer Nature Protocols and Methods: <https://experiments.springernature.com/sources/springer-protocols>
13. Springer Materials: <http://materials.springer.com/>
14. Nano Database: <https://nano.nature.com/>
15. Springer eBooks (i.e. 2020 eBook collections): <https://link.springer.com/>
16. "Лекториум ТВ" <http://www.lektorium.tv/>
17. 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. Американская патентная база данных <http://www.uspto.gov/patft/>

3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ <http://megapro.kubsu.ru/MegaPro/Web>
2. Электронная библиотека трудов ученых КубГУ <http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6>
3. Среда модульного динамического обучения <http://moodle.kubsu.ru>
4. База учебных планов, учебно-методических комплексов, публикаций и конференций <http://infoneeds.kubsu.ru/>
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий <http://mschool.kubsu.ru>;
6. Электронный архив документов КубГУ <http://docspace.kubsu.ru/>
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" <http://icdau.kubsu.ru/>

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

По курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал. В ходе лекционных занятий разбираются основы объектно-ориентированного программирования на примере языка Java, проводится анализ наиболее популярных встроенных библиотек языка. После прослушивания лекции рекомендуется выполнить упражнения, приводимые в аудитории для самостоятельной работы.

По курсу предусмотрено проведение лабораторных занятий, на которых дается прикладной систематизированный материал. В ходе занятий пишутся программы различной сложности на языке программирования Java с использованием среды разработки Eclipse/NetBeans, а также приводятся примеры разработки программных приложений. После занятия рекомендуется выполнить упражнения, приводимые в аудитории для самостоятельной работы.

При самостоятельной работе студентов необходимо изучить литературу, приведенную в перечнях выше, для осмысления вводимых понятий, анализа предложенных подходов и методов разработки программ. Разрабатывая решение новой задачи, студент должен уметь выбрать эффективные и надежные структуры данных для представления информации, подобрать соответствующие алгоритмы для их обработки, учесть специфику языка программирования, на котором будет выполнена реализация. Студент должен уметь выполнять тестирование и отладку алгоритмов решения задач с целью обнаружения и устранения в них ошибок.

Важнейшим этапом курса является самостоятельная работа по дисциплине. В процессе самостоятельной работы студент приобретает навык создания законченного программного продукта.

Используются активные, инновационные образовательные технологии, которые способствуют развитию общекультурных, общепрофессиональных компетенций и профессиональных компетенций обучающихся:

- проблемное обучение;
- разноуровневое обучение;
- проектные методы обучения;
- исследовательские методы в обучении;
- информационно- коммуникационные технологии.

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

Учебно-методическим обеспечением курсовой работы студентов являются:

1. учебная литература;
2. нормативные документы ВУЗа;
3. методические разработки для студентов.

Самостоятельная работа студентов включает:

- оформление итогового отчета (пояснительной записки).
- анализ нормативно-методической базы организации;
- анализ научных публикации по заранее определённой теме;
- анализ и обработку информации;
- работу с научной, учебной и методической литературой,
- работа с конспектами лекций, ЭБС.

Для самостоятельной работы представляется аудитория с компьютером и доступом в Интернет, к электронной библиотеке вуза и к информационно-справочным системам.

Перечень учебно-методического обеспечения:

1. Основная образовательная программа высшего профессионального образования федерального государственного бюджетного образовательного учреждения высшего образования «Кубанский государственный университет» по направлению подготовки.
2. Положение о проведении текущего контроля успеваемости и промежуточной аттестации в федеральном государственном бюджетном образовательном учреждении высшего образования «Кубанский государственный университет».
3. Общие требования к построению, содержанию, оформлению и утверждению рабочей программы дисциплины Федерального государственного образовательного стандарта высшего профессионального образования.
4. Методические рекомендации по содержанию, оформлению и применению образовательных технологий и оценочных средств в учебном процессе, основанном на Федеральном государственном образовательном стандарте.
5. Учебный план основной образовательной программы по направлению подготовки.
6. Федеральный государственный образовательный стандарт высшего профессионального образования по направлению подготовки.

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

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

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

7.1 Перечень информационно-коммуникационных технологий

- Проверка домашних заданий и консультирование посредством электронной почты.
- Использование электронных презентаций при проведении лекционных занятий
- Система MOODLE
- Проверка домашних заданий и консультирование посредством ЭОИС КубГУ

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

OpenOffice
Компилятор C++
Oracle VirtualBox 6
VMware Workstation 16
Putty 0.76 или Kitty 0.76
FileZilla 3.57.0
WinSCP 5.19
Advanced port scanner 2.5
Python 3 (3.7 И 3.9)
numpy 1.22.0
opencv 4.5.5
Keras 2.7.0
Tensor flow 2.7.0
matplotlib 3.5.1
PyCharm 2021
Cuda Toolkit 11.6
Фреймворк Django
Firefox, любая версия
Putty, любая версия
Visual Studio Code, версия 1.52+
Eclipse PHP Development Tools, версия 2020-06+
Плагин Remote System Explorer (RSE) для Eclipse PDT
JetBrains PHP Storm
GIT
Java Version 8 Update 311
Clojure 1.10.3.1029.ps1
SWI Prolog 8.4
Intellij Idea IDE 2021
Mozilla Firefox 96
Google Chrome 97
GitHub Desktop 2.9
PHP Storm 2021
FileZilla 3.57.0
Putty 0.76

8. Материально-техническое обеспечение по дисциплине (модулю)

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	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.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.