

Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Кузбасский гуманитарно-педагогический институт
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»

Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан А.В. Фомина
«09» февраля 2023 г.

Рабочая программа дисциплины
К.М.06.02 Языки и методы программирования

Направление подготовки
01.03.02 Прикладная математика и информатика

Направленность (профиль) подготовки
ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ

Программа бакалавриата

Квалификация выпускника
бакалавр

Форма обучения
Очная

Год набора 2022

Новокузнецк 2023

Оглавление

1 Цель дисциплины	3
1.1 Формируемые компетенции.....	3
1.2 Индикаторы достижения компетенций.....	3
1.3 Знания, умения, навыки (ЗУВ) по дисциплине	4
2 Объём и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.	5
3. Учебно-тематический план и содержание дисциплины.....	5
3.1 Учебно-тематический план	5
3.2. Содержание занятий по видам учебной работы.....	6
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.....	10
5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.	13
5.1 Учебная литература	13
5.2 Материально-техническое и программное обеспечение дисциплины.....	13
5.3 Современные профессиональные базы данных и информационные справочные системы.....	14
6 Иные сведения и (или) материалы.....	14
6.1.Примерные темы курсовых работ	14
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации	15

1 Цель дисциплины.

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее - ОПОП): ОПК-4, ОПК-5.

Содержание компетенций как планируемых результатов обучения по дисциплине см. таблицы 1 и 2.

1.1 Формируемые компетенции

Таблица 1 - Формируемые дисциплиной компетенции

Наименование вида компетенции (универсальная, общепрофессиональная, профессиональная)	Наименование категории (группы) компетенций	Код и название компетенции
Общепрофессиональная	Информационно-коммуникационные технологии для профессиональной деятельности	ОПК-4. Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности
		ОПК-5. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения

1.2 Индикаторы достижения компетенций

Таблица 2 – Индикаторы достижения компетенций, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
ОПК-4. Способен понимать принципы работы современных информационных технологий и использовать их для решения задач профессиональной деятельности	ОПК 4.1 Анализирует и описывает принципы работы и требования к современным информационным технологиям, информационным системам и системам искусственного интеллекта, используемым в профессиональной деятельности (по профилю программы) в условиях цифровой экономики в РФ. ОПК 4.2 Учитывает требования информационной безопасности при решении задач профессиональной деятельности. ОПК 4.3 Применяет информационно-коммуникационные технологии и информационные системы для	К.М.06.02 Языки и методы программирования К.М.06.03 Базы данных К.М.06.04 Математические методы и программное обеспечение защиты информации К.М.06.05 Программные средства визуализации данных К.М.06.06 Пакеты прикладных программ для 3D-моделирования К.М.06.07 Теория языков и трансляций К.М.06.08 Современные технологии программирования К.М.06.09 Программирование в системах реального времени К.М.09.01(У) Технологическая (проектно-технологическая) практика К.М.09.04(П) Технологическая (проектно-технологическая) практика

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
	решения задач профессиональной деятельности.	К.М.09.05(Н) Научно-исследовательская работа
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК 5.1 Создает программный код в соответствии с техническим заданием (готовыми спецификациями). ОПК 5.2 Проверяет работоспособность программного обеспечения на основе разработанных тестовых наборов данных	К.М.06.01 Структуры и алгоритмы компьютерной обработки данных К.М.06.02 Языки и методы программирования К.М.06.08 Современные технологии программирования К.М.06.09 Программирование в системах реального времени К.М.09.04(П) Технологическая (проектно-технологическая) практика К.М.09.05(Н) Научно-исследовательская работа

1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 – Знания, умения, навыки, формируемые дисциплиной

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
ОПК-4. Способен решать задачи профессиональной деятельности с использованием существующих информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	4.2 Применяет информационно-коммуникационные технологии для решения задач профессиональной деятельности	Знать: - современные информационно-коммуникационные технологии, в том числе языки и методы программирования. Уметь: - выбирать, анализировать и оценивать языки и методы программирования с точки зрения их использования для создания программных продуктов. Владеть: - навыками применения современных языков и методов программирования в процессе решения задач профессиональной деятельности.
ОПК-5 Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения	ОПК 5.1 Создает программный код в соответствии с техническим заданием (готовыми спецификациями). ОПК 5.2 Проверяет работоспособность программного	Знать: - современные системы программирования. Уметь: - выбирать, анализировать и оценивать системы программирования с точки зрения их использования для реализации алгоритмов решения прикладных задач. Владеть: - навыками разработки алгоритмов для программных продуктов.

Код и название компетенции	Индикаторы достижения компетенции, закрепленные за дисциплиной	Знания, умения, навыки (ЗУВ), формируемые дисциплиной
	обеспечения на основе разработанных тестовых наборов данных	

2 Объем и трудоёмкость дисциплины по видам учебных занятий. Формы промежуточной аттестации.

Таблица 4 – Объем и трудоёмкость дисциплины по видам учебных занятий

Общая трудоёмкость и виды учебной работы по дисциплине, проводимые в разных формах	Объём часов по формам обучения
	ОФО
1 Общая трудоёмкость дисциплины	288
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	89
Аудиторная работа (всего):	86
в том числе:	
лекции	18
лабораторные работы	68
в интерактивной форме	
Внеаудиторная работа (всего):	3
подготовка курсовой работы (контактная работа)	3
3 Самостоятельная работа обучающихся (всего)	199
4 Промежуточная аттестация обучающегося - зачет с оценкой (3 семестр) - зачет с оценкой, курсовая работа (4 семестр)	

3. Учебно-тематический план и содержание дисциплины.

3.1 Учебно-тематический план

Таблица 5 - Учебно-тематический план очной формы обучения

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость (всего час.)	Трудоёмкость занятий (час.)				Формы текущего контроля и промежуточной аттестации успеваемости
			ОФО				
			Аудиторн. занятия		СРС		
			лекц.	практ.	лаб.		
Семестр 3							
	1. Процесс создания программного обеспечения	28			4	24	Тест
1	1.1 Жизненный цикл программного обеспечения	12			2	10	
2	1.2 Методологии разработки программного обеспечения	16			2	14	
	2. Парадигмы программирования	36			14	22	Тест
3	2.1 Процедурная (модульная)	8			4	4	Защита отчета по ЛР № 1-3

№ недели п/п	Разделы и темы дисциплины по занятиям	Общая трудоёмкость (всего час.)	Трудоемкость занятий (час.)				Формы текущего контроля и промежуточной аттестации успеваемости
			ОФО			СРС	
			Аудиторн. занятия				
			лекц.	практ.	лаб.		
	парадигма						
4	2.2 Функциональная парадигма	10			4	6	Защита отчета по ЛР № 4-5
5	2.3 Логическая парадигма	8			2	6	Защита отчета по ЛР № 6
6	2.4 Объектно-ориентированная парадигма	10			4	6	Защита отчета по ЛР № 7-8
	3 Техники написания и поддержки кода	80			18	62	Тест
7	3.1 Паттерны проектирования. Порождающие паттерны	14			4	10	Защита отчета по ЛР № 9-12
8	3.2 Структурные паттерны	24			6	18	Защита отчета по ЛР № 13-16
9	3.3 Паттерны поведения	24			6	18	Защита отчета по ЛР № 17-19
10	3.4 Оформление, рецензирование, рефакторинг кода	18			2	16	
	Промежуточная аттестация – <i>зачет с оценкой</i>						зачет с оценкой
ИТОГО по семестру 3		144			36	108	
Семестр 4							
	4. Язык программирования C#	58	6		12	40	Тест
11	4.1 Синтаксис, основные алгоритмические конструкции языка программирования C#	28	4		6	18	Защита отчета по ЛР № 1-3
12	4.2 Объектно-ориентированное программирование на C#	30	2		6	22	Защита отчета по ЛР № 4-6
	5. Язык программирования Python	83	12		20	51	Тест
13	5.1 Синтаксис, основные алгоритмические конструкции и стандартные модули языка Python	25	4		10	11	Защита отчета по ЛР № 7-11
14	5.2 Графический интерфейс пользователя	16	2		2	12	Защита отчета по ЛР № 12
15	5.3 Элементы функционального программирования в Python	24	2		4	18	Защита отчета по ЛР № 13-14
16	5.4 Объектно-ориентированное программирование в Python	18	4		4	10	Защита отчета по ЛР № 15-16
	Промежуточная аттестация – <i>зачет с оценкой, курсовая работа</i>	3					3
ИТОГО по семестру 4		144	18		32	91	3
	Всего:	288	18		68	199	3

3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	Семестр 3	
<i>Содержание лекционного курса</i>		

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
1	<i>Процесс создания программного обеспечения</i>	
1.1	Жизненный цикл программного обеспечения	<i>Жизненный цикл программного обеспечения. Модели жизненного цикла. Этапы жизненного цикла. Этапы разработки программного обеспечения. Реализация программы: высокоуровневое кодирование, детализированное кодирование.</i>
1.2	Методологии разработки программного обеспечения	<i>Методологии разработки программного обеспечения. Водопадная методология: этапы, особенности, достоинства, недостатки. Гибкие методологии: Scrum, экстремальное программирование, Kanban. Cleanroom: принципы, преимущества и недостатки.</i>
2	<i>Парадигмы программирования</i>	
2.1	Процедурная (модульная) парадигма	<i>Понятие «парадигма программирования». Процедурная декомпозиция. Процедуры и функции. Языки, поддерживающие процедурную парадигму.</i>
2.2	Функциональная парадигма	<i>Логика функциональности: комбинаторная логика или λ-исчисление. Списки и функциональные выражения. Языки функционального программирования: LISP, F#, Haskell.</i>
2.3	Логическая парадигма	<i>Язык логического программирования Prolog. Понятие «унификация». Prolog-машина: поле памяти, поле зрения, термы, функторы, детерминативы, предикаты, встроенные функции. База данных и база знаний. Предложение базы знаний: головное выражение и его раскрытие. Особенности синтаксиса. Управление исполнением программы. Динамическое пополнение и порождение программы. Организация вычислений и ввода-вывода.</i>
2.4	Объектно-ориентированная парадигма	<i>Основные понятия: объект, класс, поля, методы, экземпляры объектов. Конструктор и деструктор класса. Инкапсуляция. Понятие «Наследование». Наследование полей, методов. Полиморфизм. Виртуальные методы. Классы в C++. Определение методов класса. Переопределение операций. Подписи методов и необязательные аргументы. Производные классы, наследование.</i>
3	<i>Техники написания и поддержки кода</i>	
3.1	Паттерны проектирования. Порождающие паттерны	<i>Понятие «паттерн проектирования». Преимущества использования. Виды паттернов проектирования: порождающие паттерны, структурные паттерны, паттерны поведения. Порождающие паттерны: абстрактная фабрика, строитель, фабричный метод, прототип, одиночка.</i>
3.2	Структурные паттерны	<i>Структурные паттерны: адаптер, мост, компоновщик, декоратор, фасад, приспособленец, заместитель. Понятие «антипаттерн». Антипаттерны: программирование методом копирования и вставки, спагетти-код., золотой молоток, магические числа, жесткое кодирование, мягкое кодирование, ненужная сложность, лодочный якорь, изобретение велосипеда, изобретение одноколесного велосипеда,</i>

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
		<i>программирование перебором, слепая вера, бездумное комментирование, божественный объект, поток лавы.</i>
3.3	Паттерны поведения	<i>Поведенческие паттерны проектирования: стратегия, наблюдатель, команда, итератор, посредник, состояние.</i>
3.4	Оформление, рецензирование, рефакторинг кода	<i>Стандарт оформления кода. Общие требования к именованию классов, интерфейсов, методов и переменных. Стиль отступов для логических блоков, способ расстановки скобок, использование пробелов, стиль комментариев. Рецензирование кода (ревью): design review и code review. Преимущества, порядок, способы, сроки проведения и результаты. Рефакторинг кода.</i>
<i>Содержание лабораторных занятий</i>		
2	<i>Парадигмы программирования</i>	
2.1	Процедурная (модульная) парадигма	Лабораторная работа № 1. Простые алгоритмы на языке <i>Cu++</i> . Лабораторная работа № 2. Работа с массивами в <i>Cu++</i> : матричные вычисления. Лабораторная работа № 3. Обработка текстов в <i>Cu++</i> .
2.2	Функциональная парадигма	Лабораторная работа № 4. Разработка программ на языке <i>F#</i> . Лабораторная работа № 5. Разработка программ на языке <i>Haskell</i> .
2.3	Логическая парадигма	Лабораторная работа № 6. Разработка программ на языке <i>Prolog</i> .
2.4	Объектно-ориентированная парадигма	Лабораторная работа № 7. Разработка элементарных программ на языке <i>Cu++</i> . Лабораторная работа № 8. Разработка сложной программы на языке <i>Cu++</i> .
3	<i>Техники написания и поддержки кода</i>	
3.1	Паттерны проектирования. Порождающие паттерны	Лабораторная работа №9. Разработка программы с применением порождающего паттерна «Строитель». Лабораторная работа №10. Разработка программы с применением порождающего паттерна «Абстрактная фабрика». Лабораторная работа №11. Разработка программы с применением порождающего паттерна «Фабричный метод». Лабораторная работа №12. Разработка программы с применением порождающего паттерна «Одиночка».
3.2	Структурные паттерны	Лабораторная работа № 13. Разработка программы с применением структурного паттерна «Мост». Лабораторная работа № 14. Разработка программы с применением структурного паттерна «Адаптер». Лабораторная работа № 15. Разработка программы с применением структурного паттерна «Фасад». Лабораторная работа № 16. Разработка программы с применением структурного паттерна «Декоратор».
3.3	Паттерны поведения	Лабораторная работа № 17. Разработка программы с применением структурного паттерна «Состояние». Лабораторная работа № 18. Разработка программы с применением структурного паттерна «Стратегия». Лабораторная работа № 19. Разработка программы с применением структурного паттерна «Шаблонный метод».

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	Промежуточная аттестация - зачет с оценкой	
	Семестр 4	
	<i>Содержание лекционного курса</i>	
	<i>4. Язык программирования C#</i>	
4.1	Синтаксис, основные алгоритмические конструкции языка программирования C#	Алфавит языка. Типы данных. Переменные. Константы. Преобразование встроенных типов данных. Стандартный ввод-вывод. Встроенные функции. Математические функции. Операторы условия и выбора. Циклы. Функции. Массивы: одномерные, многомерные, «зубчатые». Работа со строками.
4.2	Объектно-ориентированное программирование на C#	Описание класса в C#. Поля и свойства класса. Описание методов класса. Параметры методов. Формальные параметры. Фактические параметры. Методы с произвольным количеством параметров. Конструкторы экземпляров класса. Наследование методов и свойств класса. Инкапсуляция. Полиморфизм. Область видимости переменной. Интерфейсы. Перегрузка методов интерфейсов.
	<i>5. Язык программирования Python</i>	
5.1	Синтаксис, основные алгоритмические конструкции и стандартные модули языка Python	Программа на Python с точки зрения интерпретатора. Типы данных. Последовательность операторов. Операторы условия и выбора. Циклы. Функции. Исключения. Понятие «Модуль» в языке Python. Пакет модулей. Интерфейс модуля. Встроенные функции. Функции преобразования типов. Числовые и строковые функции. Функции обработки данных. Функции определения свойств. Функции для доступа к внутренним структурам. Функции компиляции и исполнения. Функции ввода-вывода. Функции для работы с атрибутами.
5.2	Графический интерфейс пользователя в Python	Возможности графической библиотеки виджетов (Tk). Классы виджетов. Создание и конфигурирование виджета. Виджет форматированного текста. Менеджеры расположения.
5.3	Элементы функционального программирования в Python	Методы определения функции: с помощью оператора def и lambda-выражения. Список формальных параметров. Тело определения функции. Рекурсия. Функции как параметры и результат. Итераторы.
5.4	Объектно-ориентированное программирование в Python	Объекты в Python. Определение, конструктор, деструктор, методы класса. Особенности инкапсуляции в Python. Доступ к свойствам. Скрытие данных. Полиморфизм. Имитация типов.
	<i>Содержание лабораторных занятий</i>	
	<i>4. Язык программирования C#</i>	
4.1	Синтаксис, основные алгоритмические конструкции языка программирования C#	Лабораторная работа № 1. Простые алгоритмы на языке C#. Лабораторная работа № 2. Работа с массивами в C#: матричные вычисления. Лабораторная работа № 3. Обработка текстов в C#.
4.2	Объектно-ориентированное программирование на C#	Лабораторная работа № 4. Создание классов в C#. Лабораторная работа № 5. Использование наследования в C#. Лабораторная работа № 6. Разработка сложной программы на языке C#.

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	5. Язык программирования Python	
5.1	Синтаксис, основные алгоритмические конструкции и стандартные модули языка Python	Лабораторная работа № 7. Простые алгоритмы на языке Python. Лабораторная работа № 8. Работа со встроенными функциями в Python. Лабораторная работа № 9. Работа с массивами в Python: матричные вычисления. Лабораторная работа № 10. Обработка текстов в Python. Регулярные выражения. Лабораторная работа № 11. Работа с данными в различных форматах в Python.
5.2	Графический интерфейс пользователя	Лабораторная работа № 12. Разработка приложения с графическим интерфейсом в Python.
5.3	Элементы функционального программирования в Python	Лабораторная работа № 13. Элементы функционального программирования в Python: вычисление факториала числа, последовательность Фибоначчи. Лабораторная работа № 14. Элементы функционального программирования в Python: ленивый обход графа, ленивая сортировка.
5.4	Объектно-ориентированное программирование в Python	Лабораторная работа № 15. Создание классов в Python. Лабораторная работа № 16. Использование наследования в Python.
	Промежуточная аттестация - зачет, курсовая работа	

4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.

Для положительной оценки по результатам освоения дисциплины обучающемуся необходимо выполнить все установленные виды учебной работы. Оценка результатов работы обучающегося в баллах (по видам) приведена в таблицах 7-8.

Таблица 7 - Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС) в 3 семестре

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	80	Посещение лекционных занятий (ведение конспекта) (9 лекций)	0,2 балла - конспект 1 лекционного занятия	1-2
		Лабораторные работы (отчет о выполнении лабораторной работы) (19 работ).	2 балла - выполнение работы на 51-65% 2,5 балла – выполнение работы на 65,1-85% 3 балла – выполнение работы на 85,1-100%	38 – 57
		Тесты (3 работы)	Тест Баллы за тест: 4 балла (выполнено 51 - 65% заданий) 5 баллов (выполнено 66 - 85% заданий) 7 баллов (выполнено 86 - 100% заданий)	12-21
		Итого по текущей работе в семестре		51 - 80
Промежуточная аттестация (зачет с оценкой)	20	Тест.	6 балла (пороговое значение) 10 баллов (максимальное значение)	6 - 10
		Решение задачи 1.	2 балла (пороговое значение)	2 - 5

			5 баллов (максимальное значение)	
		Решение задачи 2.	2 балла (пороговое значение) 5 баллов (максимальное значение)	2 - 5
Итого по промежуточной аттестации (зачету с оценкой)				10 – 20 б.
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации				51 – 100 б.

Таблица 8 - Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС) в 4 семестре

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	80	Лекционные занятия (ведение конспекта) (8 занятий)	1 балл - конспект 1 лекционного занятия	8
		Лабораторные работы (отчет о выполнении лабораторной работы) (16 работ).	1,2 балла - выполнение работы на 51-65% 1,6 балла – выполнение работы на 65,1-85% 2 балла – выполнение работы на 85,1-100%	19 - 32
		Тесты (2 работы)	Баллы за тест: 7 баллов (выполнено 51 - 65% заданий) 8 баллов (выполнено 66 - 85% заданий) 10 баллов (выполнено 86 - 100% заданий)	14 - 20
		Курсовая работа	10 баллов (пороговое значение) 20 баллов (максимальное значение)	10 - 20
		Итого по текущей работе в семестре		
Промежуточная аттестация (зачет с оценкой)	20	Тест.	6 балла (пороговое значение) 10 баллов (максимальное значение)	6 - 10
		Решение задачи 1.	2 балла (пороговое значение) 5 баллов (максимальное значение)	2 - 5
		Решение задачи 2.	2 балла (пороговое значение) 5 баллов (максимальное значение)	2 - 5
Итого по промежуточной аттестации (зачету с оценкой)				10 – 20 б.
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации				51 – 100 б.

Таблица 9 - Балльно-рейтинговая оценка выполнения курсовой работы в 4 семестре

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы
Выполнение курсовой работы	80	Глава 1. Анализ средств разработки. 1.1 Анализ технологий программирования. 1.2 Анализ языков программирования. 1.3 Анализ сред разработки.	Баллы за часть 1.1: 8 балла (проведен анализ технологий программирования, сделаны выводы о применении каждой технологии для решения поставленной задачи, однако рассмотрены не все подходящие технологии) 10 балла (проведен анализ технологий программирования, сделаны корректные выводы о применении каждой технологии для решения поставленной задачи, но есть некоторые недочеты) 14 баллов (проведен в полном объеме анализ технологий программирования, сделаны корректные выводы о применении каждой технологии для решения поставленной задачи)	10-20

			Баллы за часть 1.2: 1 балла (проведен анализ языков программирования, однако не все языки программирования, подходящие для решения поставленной задачи рассмотрены) 2 баллов (проведен анализ языков программирования, однако имеются некоторые неточности, или недочеты) 3 баллов (в полном объеме проведен анализ языков программирования, сделаны корректные выводы о применении каждого языка программирования для решения поставленной задачи) Баллы за часть 1.3: 1 балла (проведен анализ сред разработки, однако рассмотрены не все подходящие среды разработки рассмотрены) 2 баллов (проведен анализ сред разработки, однако имеются некоторые неточности, или недочеты) 3 баллов (в полном объеме проведен анализ сред разработки, сделаны корректные выводы о применении каждой среды для решения поставленной задачи)	
		Глава 2. Разработка приложения 2.1 Разработка алгоритма 2.2 Программная реализация алгоритма 2.3 Результат разработки	Баллы за часть 2.1: 14 баллов (алгоритм составлен с помощью преподавателя) 18 баллов (алгоритм составлен, но наблюдаются некоторые неточности или недочеты) 20 баллов (алгоритм составлен самостоятельно и в полном объеме) Баллы за часть 2.2: 20 баллов (алгоритм реализован не в полном объеме) 25 баллов (алгоритм реализован, однако наблюдаются некоторые неточности или недочеты) 30 баллов (алгоритм реализован в полном объеме) Баллы за часть 2.3: 7 баллов (результат разработки представлен не в полном объеме) 9 баллов (результат представлен, однако наблюдаются некоторые неточности или недочеты) 10 баллов (результат представлен в полном объеме)	41-60
Итого за выполнение курсовой работы				51 - 80
Промежуточная аттестация (защита курсовой работы)	20	Устное выступление об основных результатах, полученных во время выполнения курсовой работы (5-7 минут)	6 баллов (пороговое значение) 10 баллов (максимальное значение)	6 - 10
		Сопровождение устного	2 балла (пороговое значение)	2 - 5

	выступления наглядным материалом (презентация)	5 баллов (максимальное значение)	
	Ответы на вопросы по теме курсовой работы	2 балла (пороговое значение) 5 баллов (максимальное значение)	2 - 5
Итого по промежуточной аттестации (защита курсовой работы)			10 – 20 б.
Суммарная оценка за курсовую работу: Сумма баллов текущей и промежуточной аттестации			51 – 100 б.

В промежуточной аттестации оценка выставляется в ведомость в 100-балльной шкале и в буквенном эквиваленте (таблица 10)

Таблица 10 – Соотнесение 100-балльной шкалы и буквенного эквивалента оценки

Сумма набранных баллов	Уровни освоения дисциплины и компетенций	Экзамен		Зачет
		Оценка	Буквенный эквивалент	Буквенный эквивалент
86 - 100	Продвинутый	5	отлично	Зачтено
66 - 85	Повышенный	4	хорошо	
51 - 65	Пороговый	3	удовлетворительно	
0 - 50	Первый	2	неудовлетворительно	Не зачтено

5 Материально-техническое, программное и учебно-методическое обеспечение дисциплины.

5.1 Учебная литература

Основная учебная литература

Трофимов, В.В. Алгоритмизация и программирование : учебник / В.В. Трофимов, Т.А. Павловская. – Москва : Издательство Юрайт, 2019. – 137 с. – ISBN 978-5-534-07834-3. – URL: <https://www.biblio-online.ru/viewer/algoritmizaciya-i-programmirovanie-423824>

Дополнительная учебная литература

Федоров, Д. Ю. Программирование на языке высокого уровня Python : учебное пособие для вузов / Д. Ю. Федоров. — 5-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 227 с. — (Высшее образование). — ISBN 978-5-534-17323-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/532868>.

Огнева, М. В. Программирование на языке C++: практический курс : учебное пособие для вузов / М. В. Огнева, Е. В. Кудрина. — Москва : Издательство Юрайт, 2023. — 335 с. — (Высшее образование). — ISBN 978-5-534-05123-0. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: <https://urait.ru/bcode/515142>.

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях КГПИ ФГБОУ ВО «КемГУ»:

404 Учебная аудитория для проведения: - занятий лекционного типа; - групповых и индивидуальных консультаций; - текущего контроля и промежуточной аттестации.	Учебный корпус №4. 654079, Кемеровская область, г. Новокузнецк, пр-
--	--

Специализированная (учебная) мебель: доска меловая, кафедра, столы, стулья. Оборудование: <i>переносное</i> - ноутбук, экран, проектор. Используемое программное обеспечение: MSWindows (MicrosoftImaginePremium 3 year по лицензионному договору № 1212/KMP от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). Интернет с обеспечением доступа в ЭИОС.	кт Metallургов, д. 19
502 Компьютерный класс. Учебная аудитория (мультимедийная) для проведения: - занятий лекционного типа; - занятий семинарского (практического) типа; - занятий лабораторного типа; - групповых и индивидуальных консультаций; - самостоятельной работы; - текущего контроля и промежуточной аттестации. Специализированная (учебная) мебель: доска меловая, столы компьютерные, стулья. Оборудование для презентации учебного материала: <i>стационарное</i> - компьютер, экран, проектор, наушники. Лабораторное оборудование: стационарное – компьютеры для обучающихся (16 шт.). Используемое программное обеспечение: MSWindows (MicrosoftImaginePremium 3 year по лицензионному договору № 1212/KMP от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО), MicrosoftVisualStudio (MicrosoftImaginePremium 3 year по лицензионному договору № 1212/KMP от 12.12.2018 г. до 12.12.2021 г.), Среда статистических вычислений Rv.4.0.2 (свободно распространяемое ПО). Интернет с обеспечением доступа в ЭИОС.	Учебный корпус №4. 654079, Кемеровская область, г. Новокузнецк, пр-кт Metallургов, д. 19

5.3 Современные профессиональные базы данных и информационные справочные системы.

Перечень СПБД и ИСС по дисциплине

CITForum.ru - on-line библиотека свободно доступных материалов по информационным технологиям на русском языке - <http://citforum.ru>

Научная электронная библиотека eLIBRARY.RU – крупнейший российский информационный портал в области науки, технологии, медицины и образования, содержащий рефераты и полные тексты - www.elibrary.ru

База данных Science Direct (более 1500 журналов издательства Elsevier, среди них издания по математике и информатике), режим доступа :<https://www.sciencedirect.com>

6 Иные сведения и (или) материалы.

6.1.Примерные темы курсовых работ

1. Разработка графического редактора.
2. Разработка редактора UML-диаграмм.
3. Разработка приложения моделирования и визуализации физических законов.
4. Разработка приложения «Автовокзал».
5. Разработка приложения моделирования движения транспортных средств на перекрестке.
6. Разработка приложения автоматизированного раскроя материала.

7. Разработка приложения «Библиотека».
8. Разработка приложения заказа такси.
9. Разработка приложения «Гостиница».
10. Разработка приложения для автостоянки.
11. Разработка приложения регистрации участников соревнования.
12. Разработка приложения для сервисного центра.
13. Разработка приложения «Планировщик».
14. Разработка приложения для автосервиса.
15. Разработка файлового менеджера.
16. Разработка приложения, моделирующего функционирование замкнутой биологической системы.
17. Разработка приложения для автобусного парка.
18. Разработка приложения моделирования планетарной системы.
19. Разработка приложения для вычисления площади произвольного многоугольника.
20. Разработка приложения моделирования муравейника.
21. Разработка компьютерной версии игры «Монополия».
22. Разработка приложения проведения on-line аукционов.
23. Разработка приложения заказа товаров.
24. Разработка приложения «Портфолио студента».
25. Разработка приложения «Туристическое агентство».
26. Разработка приложения моделирования химических процессов.
27. Разработка приложения формирования и отображения 3D-объектов.
28. Разработка приложения «Музей».
29. Разработка приложения «Зоопарк».
30. Разработка приложения моделирования улья.

6.2. Примерные вопросы и задания / задачи для промежуточной аттестации

Семестр 3

Таблица 11 - Примерные теоретические вопросы и практические задания / задачи к зачету с оценкой

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
1. Процесс создания программного обеспечения		
1.1 Жизненный цикл программного обеспечения	1. Жизненный цикл программного обеспечения. Модели жизненного цикла. Этапы жизненного цикла. 2. Этапы разработки программного обеспечения.	

	3. Реализация программы: высокоуровневое кодирование, детализированное кодирование.	
1.2 Методологии разработки программного обеспечения	4. Методологии разработки программного обеспечения. 5. Водопадная методология: этапы, особенности, достоинства, недостатки. 6. Гибкие методологии: Scrum, экстремальное программирование, Kanban. 7. Cleanroom: принципы, преимущества и недостатки.	
2. Парадигмы программирования		
2.1 Процедурная (модульная) парадигма	8. Парадигма программирования. Процедурная декомпозиция. 9. Процедуры и функции. 10. Языки, поддерживающие процедурную парадигму.	1. Реализовать сортировку массива методом пузырька на языке Си++. 2. Реализовать умножение матриц на языке Си++.
2.2 Функциональная парадигма	11. Логика функциональности: комбинаторная логика или λ -исчисление. 12. Списки и функциональные выражения. 13. Языки функционального программирования: LISP, F#, Haskell.	3. Реализовать решение квадратного уравнения на языке F#. 4. Реализовать решение биквадратного уравнения на языке F# 5. Реализовать вывод простых чисел на языке Haskell. 6. Реализовать вывод чисел Фибоначчи на языке Haskell.
2.3 Логическая парадигма	14. Язык логического программирования Prolog. 15. Унификация. Prolog-машина: поле памяти, поле зрения, термы, функторы, детерминативы, предикаты, встроенные функции. 16. База данных и база знаний. Предложение базы знаний: головное выражение и его раскрытие. 17. Особенности синтаксиса. Управление исполнением программы. 18. Динамическое пополнение и порождение программы. 19. Организация вычислений и ввода-вывода.	7. Реализовать вычисление факториала на языке Prolog. 8. Реализовать вычисление чисел Фибоначчи на языке Prolog 9. Разработать предикат, позволяющий записать элементы списка в обратном порядке. 10. Разработать предикат, позволяющий «повторять» за пользователем введенные символы.
2.4 Объектно-ориентированная парадигма	20. Основные понятия: объект, класс, поля, методы, экземпляры объектов. Конструктор и деструктор класса. 21. Инкапсуляция. Понятие «Наследование». Наследование полей, методов. Полиморфизм. Виртуальные методы. 22. Классы в Си++. Определение методов класса. Переопределение операций. Подписи методов и необязательные аргументы.	11. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол. 12. Построить описание класса, содержащего информацию о почтовом адресе организации.

	Производные классы, наследование.	Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.
3. Техники написания и поддержки кода		
3.1 Паттерны проектирования. Порождающие паттерны	23. Понятие «паттерн проектирования». Преимущества использования. Виды паттернов проектирования: порождающие паттерны, структурные паттерны, паттерны поведения. 24. Порождающие паттерны: абстрактная фабрика. 25. Порождающие паттерны: строитель. 26. Порождающие паттерны: фабричный метод. 27. Порождающие паттерны: прототип. 28. Порождающие паттерны: одиночка.	13. Реализовать паттерн абстрактная фабрика. 14. Реализовать паттерн строитель. 15. Реализовать паттерн фабричный метод. 16. Реализовать паттерн прототип. 17. Реализовать паттерн одиночка.
3.2 Структурные паттерны	29. Структурные паттерны: адаптер. 30. Структурные паттерны: мост. 31. Структурные паттерны: компоновщик. 32. Структурные паттерны: декоратор. 33. Структурные паттерны: фасад. 34. Структурные паттерны: приспособленец. 35. Структурные паттерны: заместитель. 36. Антипаттерны.	18. Реализовать паттерн мост. 19. Реализовать паттерн компоновщик. 20. Реализовать паттерн декоратор. 21. Реализовать паттерн фасад. 22. Реализовать паттерн приспособленец. 23. Реализовать паттерн заместитель.
3.3 Паттерны поведения	37. Паттерны поведения: стратегия. 38. Паттерны поведения: наблюдатель. 39. Паттерны поведения: команда. 40. Паттерны поведения: итератор. 41. Паттерны поведения: посредник. 42. Паттерны поведения: состояние.	24. Реализовать паттерн стратегия. 25. Реализовать паттерн наблюдатель. 26. Реализовать паттерн команда. 27. Реализовать паттерн итератор. 28. Реализовать паттерн посредник. 29. Реализовать паттерн состояние.
3.4 Оформление, рецензирование, рефакторинг кода	43. Стандарт оформления кода. 44. Рецензирование кода: design review и code review. 45. Рецензирование кода: преимущества, порядок, способы, сроки проведения и результаты. 46. Рефакторинг кода.	

Таблица 12 - Примерные теоретические вопросы и практические задания / задачи к зачету с оценкой

Разделы и темы	Примерные теоретические вопросы	Примерные практические задания / задачи
4. Язык программирования C#		
4.1 Синтаксис, основные алгоритмические конструкции языка программирования C#	1. Язык C#. Алфавит языка. Типы данных. Переменные. Константы. Преобразование встроенных типов данных. 2. Язык C#. Стандартный ввод-вывод. Встроенные функции. Математические функции. 3. Язык C#. Операторы условия и выбора. Циклы. 4. Язык C#. Функции. 5. Язык C#. Массивы: одномерные, многомерные, «зубчатые». 6. Язык C#. Работа со строками.	1. Реализовать сортировку массива методом пузырька на языке C#. 2. Реализовать умножение матриц на языке C#.
4.2 Объектно-ориентированное программирование на C#	7. Описание класса в C#. Поля и свойства класса. Описание методов класса. Параметры методов. Формальные параметры. Фактические параметры. Методы с произвольным количеством параметров. 8. Конструкторы экземпляров класса. Наследование методов и свойств класса. Инкапсуляция. Полиморфизм. Область видимости переменной. 9. Интерфейсы. Перегрузка методов интерфейсов.	3. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол. 4. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.
5. Язык программирования Python		
5.1 Синтаксис, основные алгоритмические конструкции и стандартные модули языка Python	10. Программа на Python с точки зрения интерпретатора. Типы данных. Последовательность операторов. Операторы условия и выбора. Циклы. Функции. Исключения. 11. Понятие «Модуль» в языке Python. Пакет модулей. Интерфейс модуля. Встроенные функции. Функции преобразования типов. Числовые и строковые функции.	5. Реализовать сортировку массива методом пузырька на языке Python. 6. Реализовать умножение матриц на языке Python.
5.2 Графический интерфейс пользователя в Python	12. Возможности графической библиотеки виджетов (Tk). Классы виджетов. 13. Создание и конфигурирование виджета. 14. Виджет форматированного текста. 15. Менеджеры расположения.	7. Реализовать графический интерфейс для алгоритма генерации чисел Фибоначчи. 8. Реализовать графический интерфейс для алгоритма генерации простых чисел.

5.3 Элементы функционального программирования в Python	16. Методы определения функции: с помощью оператора def и lambda-выражения. 17. Список формальных параметров. Тело определения функции. 18. Рекурсия. Функции как параметры и результат. Итераторы.	9. Реализовать вывод простых чисел на языке Python. 10. Реализовать вывод чисел Фибоначчи на языке Python.
5.4 Объектно-ориентированное программирование в Python	19. Объекты в Python. Определение, конструктор, деструктор, методы класса. 20. Особенности инкапсуляции в Python. Доступ к свойствам. Скрытие данных. Полиморфизм. Имитация типов.	11. Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и поворота на заданный угол. 12. Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность раздельного изменения составных частей адреса, создания и уничтожения объектов этого класса.

Составитель (и): старший преподаватель кафедры МФММ Гаврилова Ю.С.
(фамилия, инициалы и должность преподавателя (ей))