Подписано электронной подписью:
Вержицкий Данил Григорьевич
Должность: Директор КГПИ ФГБОУ ВО «КемГУ»
Дата и время: 2024-02-21 00:00:00
471086fad29a3b30e244c728abc3661ab35e9d50310dcf0e75e03a5b6fdf6436

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Кемеровский государственный университет»
Кузбасский гуманитарно-педагогический институт
федерального государственного бюджетного образовательного учреждения
высшего образования
«Кемеровский государственный университет»
Факультет информатики, математики и экономики

УТВЕРЖДАЮ
Декан
А. В. Фомина
23 июня 2021 г.

Рабочая программа дисциплины

Б1.О.15 Языки и методы программирования
Код, название дисциплины

Направление подготовки
02.03.03 Математическое обеспечение и администрирование информационных систем
Код, название направления

Направленность (профиль) подготовки
Программное и математическое обеспечение информационных технологий

Программа бакалавриата

Квалификация выпускника
бакалавр

Форма обучения
Очная

Год набора 2021

Новокузнецк 2021

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. 4
3. Учебно-тематический план и содержание дисциплины. 5
3.1 Учебно-тематический план 5
3.2. Содержание занятий по видам учебной работы 7
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. 14
5.3 Современные профессиональные базы данных и информационные справочные системы. 15
6 Иные сведения и (или) материалы. 15
6.1.Примерные темы курсовых работ 15
6.2. Примерные вопросы и задания / задачи для промежуточной аттестации 16

1 Цель дисциплины.

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

ОПК-3.

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-3 Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения

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-3 Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	3.1 Применяет современные информационные технологии, в том числе отечественные, и инструментальные средства для производства программного продукта: 3.2 Использует современные информационные технологии для тестирования и отладки программного обеспечения; 3.3 Использует методы и средства автоматизации проектирования программных продуктов 3.4 Владеет CASE (Computer-Aided Software Engineering) средствами 3.5 Анализирует и описывает принципы работы и требования к современным ИТ, ИС, СИИ, используемых в	Б1.О.12 Информатика Б1.О.15 Языки и методы программирования Б1.О.16 Математические методы и программное обеспечение защиты информации Б1.О.18 Операционные системы Б1.О.24 Информационные системы и технологии Б1.О.27 Базы данных Б1.О.30 Программная инженерия Б2.О.01(У) Учебная практика. Технологическая (проектно-технологическая) практика Б2.О.02(П) Производственная практика. Технологическая (проектно-технологическая) практика Б3.01(Д) Подготовка к

Код и название компетенции	Индикаторы достижения компетенции по ОПОП	Дисциплины и практики, формирующие компетенцию ОПОП
	профессиональной деятельности в условиях цифровой экономики 3.6 Используем возможности современных ИТ, ИС, СИИ для решения типовых задач профессиональной деятельности	процедуре защиты и защита выпускной квалификационной работы

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-3 Способен понимать и применять современные информационные технологии, в том числе отечественные, при создании программных продуктов и программных комплексов различного назначения	3.1 Применяет современные информационные технологии, в том числе отечественные, и инструментальные средства для производства программного продукта 3.6 Используем возможности современных ИТ, ИС, СИИ для решения типовых задач профессиональной деятельности	**Знать:** - современные информационные технологии, в том числе отечественные, языки и методы программирования, среды разработки; - принципы, методы работы, возможности, типовые технологические операции и процессы в современных ИТ и ИС; - ИТ, ИС, используемые в профессиональной деятельности для решения типовых профессиональных задач **Уметь**: - выбирать, анализировать и оценивать направление программирования с точки зрения его использования для создания программных продуктов. **Владеть**: - навыками применения современных ИТ и ИС в процессе создания программных продуктов.

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

Таблица 4 – Объем и трудоемкость дисциплины по видам учебных занятий

Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах	Объём часов по формам обучения		
	ОФО	ОЗФО	ЗФО
1 Общая трудоемкость дисциплины	216		
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	109		
Аудиторная работа (всего):	106		
в том числе:			

лекции	34		
практические занятия, семинары			
практикумы			
лабораторные работы	72		
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):	3		
в том числе, индивидуальная работа обучающихся с преподавателем			
подготовка курсовой работы /контактная работа	3		
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем)			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)	107		
4 Промежуточная аттестация обучающегося	Зачет – 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.)*	Трудоемкость занятий (час.)							Формы текущего контроля и промежуточной аттестации успеваемости
			ОФО				ЗФО			
			Аудиторн. занятия			СРС	Аудиторн. занятия		СРС	
			лекц.	практ.	лаб.		лекц	практ.		
Семестр 3										
	1. Процесс создания программного обеспечения	**4**	**2**			**2**				**Тест**
1	1.1 Жизненный цикл программного обеспечения	2	1			1				
2	1.2 Методологии разработки программного обеспечения	2	1			1				
	2. Парадигмы программирования	**32**	**8**		**16**	**8**				**Тест**
3	2.1 Процедурная (модульная) парадигма	10	2		6	2				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				ЗФО			
			Аудиторн. занятия			СРС	Аудиторн. занятия		СРС	
			лекц.	практ.	лаб.		лекц	практ.		
4	2.2 Функциональная парадигма	8	2		4	2				Защита отчета по ЛР № 4-5
5	2.3 Логическая парадигма	6	2		2	2				Защита отчета по ЛР № 6
6	2.4 Объектно-ориентированная парадигма	8	2		4	2				Защита отчета по ЛР № 7-8
	3 Техники написания и поддержки кода	**36**	**8**		**18**	**10**				**Тест**
7	3.1 Паттерны проектирования. Порождающие паттерны	12	2		6	4				Защита отчета по ЛР № 9-11
8	3.2 Структурные паттерны	10	2		6	2				Защита отчета по ЛР № 12-14
9	3.3 Паттерны поведения	10	2		4	4				Защита отчета по ЛР № 15-17
10	3.4 Оформление, рецензирование, рефакторинг кода	4	2			2				
	Промежуточная аттестация – *зачет*									зачет
ИТОГО по семестру 3		**72**	**18**		**32**	**22**				
Семестр 4										
	4. Язык программирования C#	**63**	**6**		**12**	**45**				**Тест**
11	4.1 Синтаксис, основные алгоритмические конструкции языка программирования C#	38	4		6	28				Защита отчета по ЛР № 1-3
12	4.2 Объектно-ориентированное программирование на C#	25	2		6	17				Защита отчета по ЛР № 4-6
	5. Язык программирования Python	**78**	**10**		**20**	**48**				**Тест**
13	5.1 Синтаксис, основные алгоритмические конструкции и стандартные модули языка Python	33	4		12	17				Защита отчета по ЛР № 7-11
14	5.2 Графический интерфейс пользователя	8	2		4	2				Защита отчета по ЛР № 12
15	5.3 Элементы функционального программирования в Python	23	2		6	15				Защита отчета по ЛР № 13-14
16	5.4 Объектно-ориентированное программирование в Python	14	2		6	6				Защита отчета по ЛР № 15-16
	Промежуточная аттестация – *зачет с оценкой, курсовая работа*	3								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				ЗФО			
			Аудиторн. занятия			СРС	Аудиторн. занятия		СРС	
			лекц.	практ.	лаб.		лекц	практ.		
ИТОГО по семестру 4		**144**	**16**		**40**	**85**				**3**
	Всего:	**180**	**34**		**72**	**107**				

3.2. Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	Семестр 3	
	Содержание лекционного курса	
1	*Процесс создания программного обеспечения*	
1.1	Жизненный цикл программного обеспечения	*Жизненный цикл программного обеспечения. Модели жизненного цикла. Этапы жизненного цикла.* *Этапы разработки программного обеспечения. Реализация программы: высокоуровневое кодирование, детализированное кодирование.*
1.2	Методологии разработки программного обеспечения	*Методологии разработки программного обеспечения. Водопадная методология: этапы, особенности, достоинства, недостатки.* *Гибкие методологии: Scrum, экстремальное программирование, Kanban.* *Cleanroom: принципы, преимущества и недостатки.*
2	*Парадигмы программирования*	
2.1	Процедурная (модульная) парадигма	*Понятие «парадигма программирования». Процедурная декомпозиция. Процедуры и функции. Языки, поддерживающие процедурную парадигму.*
2.2	Функциональная парадигма	*Логика функциональности: комбинаторная логика или λ-исчисление. Списки и функциональные выражения. Языки функционального программирования: LISP, F#, Haskell.*
2.3	Логическая парадигма	*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.*
2.4	Объектно-ориентированная парадигма	*Основные понятия: объект, класс, поля, методы, экземпляры объектов. Конструктор и деструктор класса. Инкапсуляция. Понятие «Наследование». Наследование полей, методов. Полиморфизм. Виртуальные методы.* *Классы в Си++. Определение методов класса. Переопределение операций. Подписи методов и необязательные аргументы. Производные классы, наследование.*
3	*Техники написания и поддержки кода*	

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
3.1	Паттерны проектирования. Порождающие паттерны	*Понятие «паттерн проектирования». Преимущества использования. Виды паттернов проектирования: порождающие паттерны, структурные паттерны, паттерны поведения.* *Порождающие паттерны: абстрактная фабрика, строитель, фабричный метод, прототип, одиночка.*
3.2	Структурные паттерны	*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, программирование перебором, слепая вера, бездумное комментирование, божественный объект, поток лавы.*
3.3	Паттерны поведения	*Поведенческие паттерны проектирования: стратегия, наблюдатель, команда, итератор, посредник, состояние.*
3.4	Оформление, рецензирование, рефакторинг кода	*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.*
Содержание лабораторных занятий		
2	*Парадигмы программирования*	
2.1	Процедурная (модульная) парадигма	Лабораторная работа № 1. Простые алгоритмы на языке *Си++*. Лабораторная работа № 2. Работа с массивами в *Си++*: матричные вычисления. Лабораторная работа № 3. Обработка текстов в *Си++*.
2.2	Функциональная парадигма	Лабораторная работа № 4. Разработка программ на языке *F#*. Лабораторная работа № 5. Разработка программ на языке *Haskell*.
2.3	Логическая парадигма	Лабораторная работа № 6. Разработка программ на языке *Prolog*.
2.4	Объектно-ориентированная парадигма	Лабораторная работа № 7. Разработка элементарных программ на языке *Си++*. Лабораторная работа № 8. Разработка сложной программы на языке *Си++*.
3	*Техники написания и поддержки кода*	
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».
3.2	Структурные паттерны	Лабораторная работа № 12. Разработка программы с применением структурного паттерна «Мост».

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
		Лабораторная работа № 13. Разработка программы с применением структурного паттерна «Адаптер». Лабораторная работа № 14. Разработка программы с применением структурного паттерна «Фасад».
3.3	Паттерны поведения	Лабораторная работа № 15. Разработка программы с применением структурного паттерна «Состояние». Лабораторная работа № 16. Разработка программы с применением структурного паттерна «Стратегия». Лабораторная работа № 17. Разработка программы с применением структурного паттерна «Шаблонный метод».
	Промежуточная аттестация - *зачет*	
	Семестр 4	
Содержание лекционного курса		
	4. Язык программирования C#	
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.
	5. Язык программирования Python	
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. Сокрытие данных. Полиморфизм. Имитация типов.

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
Содержание лабораторных занятий		
	4. Язык программирования C#	
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#.
	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 с оценкой, 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 7-9.

Таблица 7 - Шкала и показатели оценивания результатов учебной работы обучающихся по видам в балльно-рейтинговой системе (БРС) семестр 3

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации (шкала и показатели оценивания)	Баллы
Текущая учебная работа в семестре (Посещение занятий по	**80**	Посещение лекционных занятий (ведение конспекта) (9 лекций)	**0,6 балла** - конспект 1 лекционного занятия	5
		Лабораторные работы	**2 балла** - выполнение работы на 51-65%	34 – 51

расписанию и выполнение заданий)		(отчет о выполнении лабораторной работы) (17 работ).	**2,5 балла** – выполнение работы на 65,1-85% **3 балла** – выполнение работы на 85,1-100%	
		Тесты (3 работы)	**Тест** **Баллы за тест:** **5 баллов** (выполнено 51 - 65% заданий) **6 баллов** (выполнено 66 - 85% заданий) **8 баллов** (выполнено 86 - 100% заданий)	15-24
Итого по текущей работе в семестре				51 - 80
Промежуточная аттестация (зачет)	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)				10 – 20 б.
Суммарная оценка по дисциплине: Сумма баллов текущей и промежуточной аттестации				51 – 100 б.

Таблица 8 - Шкала и показатели оценивания результатов учебной работы обучающихся по видам в балльно-рейтинговой системе (БРС) семестр 4

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации (шкала и показатели оценивания)	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:** **10 баллов** (выполнено 51 - 65% заданий) **15 баллов** (выполнено 66 - 85% заданий) **20 баллов** (выполнено 86 - 100% заданий)	20 - 40
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) **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 - Шкала и показатели оценивания результатов учебной работы обучающихся по видам в балльно-рейтинговой системе (БРС) курсовая работа

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации (шкала и показатели оценивания)	Баллы
Выполнение	**80**	Глава 1. Анализ средств	**Баллы за часть 1.1:**	10-20

курсовой работы		разработки. 1.1 Анализ технологий программирования. 1.2 Анализ языков программирования. 1.3 Анализ сред разработки.	**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) **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	41-60

			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)	
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 выступления наглядным материалом (презентация)	**2 балла** (пороговое значение) **5 баллов** (максимальное значение)	2 - 5
		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

1. Трофимов, В. В. Алгоритмизация и программирование : учебник для академического бакалавриата / В. В. Трофимов, Т. А. Павловская ; под редакцией В. В. Трофимова. — Москва : Издательство Юрайт, 2019. — 137 с. — (Бакалавр. Академический курс. Модуль). — ISBN 978-5-534-07834-3. — Текст : электронный // ЭБС Юрайт [сайт]. — URL: https://urait.ru/bcode/423824

2. Калентьев А.А. Новые технологии в программировании : учебное пособие / А.А. Калентьев, Д.В. Гарайс, А.Е. Горяинов – Томск : Эль Контент, 2014. – 176 с. – ISBN 978-5-4332-0185-9.- URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=480503

3. Сузи, Р.А. Язык программирования Python: учебное пособие / Р.А. Сузи. – 2-е изд., испр. – Москва: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2007. – 327 с. - ISBN 978-5-94774-711-9. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=233288

4. Мирошниченко, И.И. Языки и методы программирования: учебное пособие / И.И. Мирошниченко, Е.Г. Веретенникова, Н.Г. Савельева. – Ростов н/Д: Издательско-полиграфический комплекс Рост. гос. экон. ун-та (РИНХ), 2019. – 188 с. - ISBN 978-5-7972-2604-8. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=567706

Дополнительная учебная литература

5. Прыкина, Е.Н. Основы логического программирования в среде Турбо Пролог : учебное пособие / Е.Н. Прыкина. – Кемерово : КемГУКИ, 2006. – 68 с. – ISBN 5-8154-0130-7. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=227891

6. Баженов, Р.И. Лабораторный практикум по функциональному программированию: учебно-методическое пособие / Р.И. Баженов. – Москва; Берлин : Директ-Медиа, 2017. – 90 с. – ISBN 978-5-4475-9458-9.- URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=480173

7. Язык программирования Си++. Курс лекций : учебное пособие / Фридман А.Л. – Москва : ИНТУИТ.РУ «Интернет-университет Информационных Технологий», 2004. – 262 с. – ISBN 5-9556-0017-5. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=233058

5.2 Материально-техническое и программное обеспечение дисциплины.

Учебные занятия по дисциплине проводятся в учебных аудиториях НФИ КемГУ:

615 Учебная аудитория (мультимедийная) для проведения: - занятий лекционного типа. **Специализированная (учебная) мебель:** доска меловая, кафедра, столы, стулья. **Оборудование для презентации учебного материала:** *стационарное* - компьютер, экран, проектор, акустическая система (колонки). **Используемое программное обеспечение:** Ubuntu Linux(свободно распространяемое ПО), LibreOffice (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). **Интернет с обеспечением доступа в ЭИОС.**	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19
501 Компьютерный класс. Учебная аудитория (мультимедийная)для проведения: - занятий лабораторного типа; - курсового проектирования (выполнения	654079, Кемеровская область, г. Новокузнецк, пр-кт Металлургов, д. 19

курсовых работ); - групповых и индивидуальных консультаций; - текущего контроля и промежуточной аттестации. **Специализированная (учебная) мебель:** доска меловая, кафедра, столы компьютерные, стулья. **Оборудование для презентации учебного материала:** *стационарное* - компьютер преподавателя, экран, проектор. **Оборудование:** *стационарное* - компьютеры для обучающихся (17 шт.). **Используемое программное обеспечение:** MS Windows (Microsoft Imagine Premium 3 year по сублицензионному договору № 1212/КМР 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), Opera 12 (свободно распространяемое ПО), Microsoft Visual Studio (Microsoft Imagine Premium 3 year по сублицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.). **Интернет с обеспечением доступа в ЭИОС.**	

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

Единое окно доступа к образовательным ресурсам - http://window.edu.ru/

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

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.	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.

	19. Организация вычислений и ввода-вывода.	
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. Производные классы, наследование.	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.
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.	24. Реализовать паттерн стратегия. 25. Реализовать паттерн наблюдатель. 26. Реализовать паттерн команда. 27. Реализовать паттерн итератор. 28. Реализовать паттерн

	42. Паттерны поведения: состояние.	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.	

Семестр 4

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	5. Реализовать сортировку массива методом пузырька на языке Python. 6. Реализовать умножение матриц на языке Python.

	функции. Функции преобразования типов. Числовые и строковые функции.	
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. Сокрытие 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): Гаврилова Ю. С., старший преподаватель кафедры математики, физики и математического моделирования

(*фамилия, инициалы и должность преподавателя (ей)*)