

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

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

«Кемеровский государственный университет»  
Кузбасский гуманитарно-педагогический институт

Факультет информатики, математики и экономики

«УТВЕРЖДАЮ»  
Декан ФИМЭ  
А.В. Фомина  
«10» февраля 2023 г.

**Рабочая программа дисциплины**

*К.М.07.01.02 Программирование*

Направление подготовки

*44.03.05 Педагогическое образование (с двумя профилями подготовки)*

Направленность (профиль) подготовки

***Информатика и Системы искусственного интеллекта***

Программа бакалавриата

Квалификация выпускника  
*бакалавр*

Форма обучения  
*Очная*

Год набора 2023

Новокузнецк 2023

## ОГЛАВЛЕНИЕ

Оглавление .....	2
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 .....	5
4 Порядок оценивания успеваемости и сформированности компетенций обучающегося в текущей и промежуточной аттестации.....	8
5 Учебно-методическое обеспечение дисциплины .....	8
5.1 Учебная литература .....	8
5.2 Материально-техническое и программное обеспечение дисциплины.....	9
5.3 Современные профессиональные базы данных и информационные справочные системы.....	10
6 Иные сведения и (или) материалы .....	11
6.1 Примерные темы письменных учебных работ .....	11
6.2 Примерные вопросы и задания / задачи для промежуточной аттестации .....	11

## 1 ЦЕЛЬ ДИСЦИПЛИНЫ

В результате освоения данной дисциплины у обучающегося должны быть сформированы компетенции основной профессиональной образовательной программы бакалавриата (далее — ОПОП): ПК-1.

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
профессиональные компетенции	Информационно-коммуникационные технологии для профессиональной деятельности	ПК-1 Способен осуществлять разработку и реализацию образовательных программ основного и среднего общего образования на основе специальных научных знаний в предметной области "Информатика"

### 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
ПК-1 Способен осуществлять разработку и реализацию образовательных программ основного и среднего общего образования на основе специальных научных знаний в предметной области "Информатика"	<p>ПК-1.1 Проектирует элементы образовательной программы и рабочую программу по информатике, формулирует дидактические цели и задачи обучения информатике и ИКТ и реализовывает их в учебном процессе, моделирует и реализовывает различные организационные формы обучения информатике</p> <p>ПК-1.2 Использует педагогические технологии для достижения личностных, предметных и метапредметных результатов обучающихся в предметной области Информатика</p> <p>ПК-1.3 Демонстрирует владение методикой преподавания по предмету. Информатика различных категорий обучающихся в соответствии с основной образовательной программой на основе деятельностного подхода и владения современными</p>	<p>К.М.07.01.02 Программирование</p> <p>К.М.07.01.03 Компьютерные сети и интернет-технологии</p> <p>К.М.07.01.04 Теоретические основы информатики</p> <p>К.М.07.01.05 Операционные системы</p> <p>К.М.07.01.06 Компьютерное моделирование</p> <p>К.М.07.01.07 Компьютерная графика и анимация</p> <p>К.М.07.01.08 Проектирование информационных систем</p> <p>К.М.07.01.09 Системы управления базами данных</p> <p>К.М.07.01.10 Оценивание и мониторинг образовательных результатов учащихся по информатике</p> <p>К.М.07.03(У) Технологическая практика. Стандарты подготовки школьников по информатике</p> <p>К.М.07.04 Видеомонтаж</p> <p>К.М.07.ДВ.01.01 Методика подготовки к государственной итоговой аттестации по информатике</p> <p>К.М.07.ДВ.01.02 Решение задач по информатике повышенной сложности</p> <p>К.М.09.02(П) Педагогическая практика. Основная школа</p> <p>К.М.09.03(П) Педагогическая практика. Старшая школа</p> <p>К.М.10.01(Пд) Преддипломная практика</p> <p>К.М.10.02(Г) Подготовка к сдаче и сдача государственного экзамена</p> <p>К.М.10.03(Д) Выполнение и защита выпускной</p>

<b>Код и название компетенции</b>	<b>Индикаторы достижения компетенции по ОПОП</b>	<b>Дисциплины и практики, формирующие компетенцию ОПОП</b>
	педагогическими технологиями	квалификационной работы

### 1.3 Знания, умения, навыки (ЗУВ) по дисциплине

Таблица 3 — Знания, умения, навыки, формируемые дисциплиной

<b>Код и название компетенции</b>	<b>Индикаторы достижения компетенции, закрепленные за дисциплиной</b>	<b>Знания, умения, навыки (ЗУВ), формируемые дисциплиной</b>
ПК-1 Способен осуществлять разработку и реализацию образовательных программ основного и среднего общего образования на основе специальных научных знаний в предметной области "Информатика"	<p>ПК-1.1 Проектирует элементы образовательной программы и рабочую программу по информатике, формулирует дидактические цели и задачи обучения информатике и ИКТ и реализовывает их в учебном процессе, моделирует и реализовывает различные организационные формы обучения информатике</p> <p>ПК-1.2 Использует педагогические технологии для достижения личностных, предметных и метапредметных результатов обучающихся в предметной области Информатика</p> <p>ПК-1.3 Демонстрирует владение методикой преподавания по предмету. Информатика различных категорий обучающихся в соответствии с основной образовательной программой на основе деятельностного подхода и владения современными педагогическими технологиями</p>	<p>Знать:</p> <ul style="list-style-type: none"> <li>- научное содержание и современное состояние предметной области «Программирование», лежащее в основе преподаваемого учебного предмета «Информатика»</li> <li>- методы проведения научного исследования в предметной области «Программирование»;</li> </ul> <p>Уметь:</p> <ul style="list-style-type: none"> <li>- использовать научные знания предметной области «Программирование» в педагогической деятельности по профилю подготовки;</li> <li>- применять научные знания предметной области «Программирование» при разработке образовательных программ, рабочих программ учебных предметов, курсов внеурочной деятельности;</li> </ul> <p>Владеть:</p> <ul style="list-style-type: none"> <li>- методами научного исследования в области программирования;</li> <li>- способами получения информации о современном состоянии научных исследований в предметной области «Программирование»</li> </ul>

## 2 ОБЪЁМ И ТРУДОЁМКОСТЬ ДИСЦИПЛИНЫ ПО ВИДАМ УЧЕБНЫХ ЗАНЯТИЙ. ФОРМЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ.

Таблица 4 — Объем и трудоемкость дисциплины по видам учебных занятий

<b>Общая трудоемкость и виды учебной работы по дисциплине, проводимые в разных формах</b>	<b>Объём часов по формам обучения</b>		
	<b>ОФО</b>	<b>ОЗФО</b>	<b>ЗФО</b>
1 Общая трудоемкость дисциплины	288		
2 Контактная работа обучающихся с преподавателем (по видам учебных занятий) (всего)	92		
Аудиторная работа (всего):	92		
в том числе:			
лекции	44		
практические занятия, семинары			
практикумы			
лабораторные работы	48		
в интерактивной форме			
в электронной форме			
Внеаудиторная работа (всего):			
в том числе, индивидуальная работа обучающихся с преподавателем			

подготовка курсовой работы /контактная работа			
групповая, индивидуальная консультация и иные виды учебной деятельности, предусматривающие групповую или индивидуальную работу обучающихся с преподавателем)			
творческая работа (эссе)			
3 Самостоятельная работа обучающихся (всего)	160		
4 Промежуточная аттестация обучающегося:	зачет с оценкой 3 семестр, экзамен – 36 ч. 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			ЗФО			
			Аудиторн. занятия		СР С	Аудиторн. занятия		СР С	
лекц.	практ.	лек. ц.	практ.						
<b>Семестр 3</b>									
1	Введение в алгоритмизацию и программирование	74	10	10	54				Устный опрос, индивидуальное задание
2	Структурный подход к программированию	70	10	10	50				Устный опрос, индивидуальное задание
3	Промежуточная аттестация								Зачет с оценкой
ИТОГО по семестру		144	20	20	104				
<b>Семестр 4</b>									
1	Модульное программирование. Программирование абстрактных типов данных	82	20	20	42				Устный опрос, индивидуальное задание
2	Объектно-ориентированное программирование	26	4	8	14				Устный опрос, индивидуальное задание
3	Промежуточная аттестация	36							Экзамен
Итого по семестру		144	24	28	56				
Всего:		288	44	48	160				

### 3.2 Содержание занятий по видам учебной работы

Таблица 6 – Содержание дисциплины

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
<b>Семестр 3</b>		
<i>Содержание лекционного курса</i>		
1	Введение в алгоритмизацию и программирование	
1.1	Методология императивного программирования	Основные принципы программирования. Императивный подход. Основные черты императивных языков. Методы и концепции.
1.2	Методология объектно-ориентированного программирования	Объектно-ориентированное программирование. Преимущества и недостатки объектно-ориентированного программирования. Взаимодействие программных объектов между собой. Классы объектов. Свойства объектов. Методы класса. События в объектно-ориентированном программировании.
1.3	Алгоритмические структуры. Язык блок-схем	Основные алгоритмические структуры. Графическая реализация алгоритма. Понятие блок-схемы. Основные виды блоков
1.4	Синтаксис и семантика формального языка.	Синтаксис и семантика формального языка. Концепции языков программирования. Описание синтаксиса. Формальное описание языков. Формальные методы описания синтаксиса.
2	Структурный подход к программированию	
2.1	Основные конструкции алгоритмических языков	Алгоритмы, структуры алгоритмов, структурное программирование. Этапы: Постановки задачи. Формализация задачи. Построение алгоритма. Данные и величины. Классификации данных по структуре. Исполнитель алгоритмов.
2.2	Простые типы языка программирования	Типы в языках программирования. Классификация типов данных. Числовые типы данных. Вещественные типы данных. Символьный тип данных. Логический тип данных.
2.3	Структурированные типы языка программирования высокого уровня	Строки. Кортежи и списки.
2.4	Алгоритмы поиска и сортировки	Задача поиска. Линейный поиск. Бинарный поиск. Задача сортировки. Основные виды сортировки: обменом, выбором, вставкой, слиянием.
2.5	Технология структурного программирования	Технология структурного программирования. Использование рациональной структуры программы. Модуль и основные принципы структурного подхода.
<i>Содержание лабораторных занятий</i>		
1	Введение в алгоритмизацию и программирование	
1.1	Применение линейных алгоритмов.	Линейный алгоритм. Блок-схема. Запись выражений.
1.2	Использование ветвления в программах	Использование ветвления в программах
1.3	Использование в программах цикла с предусловием	Оператор while. Операторы break и continue. Использование else совместно с циклом.
1.4	Использование в программах цикла for	Синтаксис оператора for. Генерирование числовых последовательностей с помощью функции range(). Итерирование по нескольким последовательностям с помощью функции zip().
1.5	Включения.	Включение списков. Включение словаря. Включение множества. Включение генератора.
2	Структурный подход к программированию	
2.1	Составные типы данных. Кортежи и списки.	Кортежи и списки. Создание кортежей и списков. Пустой кортеж. Получение элемента с помощью смещения. Списки списков. Извлечение подсписков с помощью срезов. Добавление и удаление элементов списка.
2.2	Применение кортежей и списков.	Поиск элементов с заданными свойствами. Матрицы., их описание, формирование и вывод. Прямоугольные и квадратные матрицы. Поиск элементов с заданными свойствами, в т. ч. Связанными с положением относительно главной и побочной диагоналей в квадратных матрицах.

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
2.3	Алгоритмы сортировки.	Алгоритмы сортировки элементов массива.
2.4	Использование словарей и множеств.	Создание словаря. Добавление или изменение элемента. Объединение словарей. Удаление элементов по ключу. Получение всех ключей, всех значений, всех пар «ключ-значение»
Промежуточная аттестация – экзамен		
<b>Семестр 4</b>		
<i>Содержание лекционного курса</i>		
1	Модульное программирование. Программирование абстрактных типов данных	
1.1	Функции.	Функции в языках программирования высокого уровня. Описание и вызов функций. Создание функций. Позиционные аргументы и аргументы — ключевые слова.
1.2	Рекурсия	Понятие рекурсии. Применение рекурсии. Виды рекурсии.
1.3	Внутренние функции. Замыкания. Анонимные функции.	Объявление внутренних функций. Понятие замыкания. Использование замыканий в программах
1.4	Генераторы. Декораторы. Пространство имен и область определения.	Назначение генераторов. Создание генераторов. Понятие декораторов. Создание декораторов. Декораторы в составе стандартной библиотеки Python.
1.5	Обработка исключений	Обработка исключений с помощью try и except. Создание собственных исключений.
1.6	Модульное программирование в python.	Понятие модуля. Создание модулей. Оператор import. Пакеты. Стандартная библиотека Python.
1.7	Динамические структуры данных (абстрактные типы данных): список, стек, очередь	Понятие списка, стека и очереди. Применение списков, стеков и очередей в программах.
1.8	Графы.	Понятие графа. Представление графов. Обход графов.
1.9	Файлы	Понятие файлов. Файловый ввод/вывод. Текстовые файлы. XML-файлы. CSV-файлы.
1.10	Сетевое взаимодействие.	Понятие сетевых протоколов. Узлы. Сокеты. Сеанс. Сервер и клиент. Организация сервера. Подключение к серверу.
2	Объектно-ориентированное программирование	
2.1	Методология и принципы объектно-ориентированного программирования.	Методология и принципы объектно-ориентированного программирования. Языки, поддерживающие методологию объектно-ориентированного программирования. Реализация абстракций данных методами объектно-ориентированного программирования (математические объекты: рациональные и комплексные числа). Реализация абстракций данных методами объектно-ориентированного программирования (вектора, матрицы). Библиотеки объектов.
2.2	Объектно-ориентированное программирование на языке Python.	Классы и объекты. Особенности работы с объектами. Модификаторы доступа, инкапсуляция. Полиморфизм и перегрузка методов. Наследование и полиморфизм.
<i>Содержание лабораторных занятий</i>		
1	Модульное программирование. Программирование абстрактных типов данных	
1.1	Создание и использование функций	Определение функций. Списки аргументов. Обязательные и необязательные аргументы. Значения по умолчанию. Передача аргументов в функцию.
1.2	Использование рекурсии	Решение задач на использование рекурсии
1.3	Использование генераторов и декораторов.	Создание собственных генераторов. Использование декораторов для отладки. Использование декораторов для решения задач динамического программирования.
1.4	Создание и использование модулей	Создание модуля. Подключение модуля к основной программе. Область видимости переменных модуля.
1.5	Списки, стеки, очередь, графы.	Решение задач на списки, стеки, очереди, графы.
2	Объектно-ориентированное	

№ п/п	Наименование раздела, темы дисциплины	Содержание занятия
	программирование	
2.1	<i>Создание классов. Определение атрибутов, методов, геттеров, и сеттеров.</i>	Создание класса. Создание экземпляра класса. Использование класса в программе.
2.2	<i>Специальные методы классов. Использование композиции.</i>	Переопределение специальных методов класса. Использование композиции
	Промежуточная аттестация - экзамен	

## 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 таблице 7.

Таблица 7 — Балльно-рейтинговая оценка результатов учебной работы обучающихся по видам (БРС)

Учебная работа (виды)	Сумма баллов	Виды и результаты учебной работы	Оценка в аттестации	Баллы (17 недель)
Текущая учебная работа в семестре (Посещение занятий по расписанию и выполнение заданий)	<b>60</b>	Лекционные занятия (конспект) (10/12 занятий)	<b>1 балл</b> — посещение 1 лекционного занятия	10 – 20/24
		Лабораторные работы (отчет о выполнении лабораторной работы) (10/14 работы).	<b>1 балл</b> — посещение 1 практического занятия и выполнение работы на 51-65% <b>2 балла</b> — посещение 1 занятия и существенный вклад на занятии в работу всей группы, самостоятельность и выполнение работы на 85,1-100%	24 - 48
<b>Итого по текущей работе в каждом семестре</b>				51 - 100
Промежуточная аттестация (зачет с оценкой / экзамен)	40 (100% /баллов приведенной шкалы)	Теоретический вопрос 1	<b>5 баллов</b> (пороговое значение) <b>10 баллов</b> (максимальное значение)	5 - 10
		Теоретический вопрос 2	<b>5 баллов</b> (пороговое значение) <b>10 баллов</b> (максимальное значение)	5 - 10
		Выполнение задания	<b>10 баллов</b> (пороговое значение) <b>20 баллов</b> (максимальное значение)	10 - 20
<b>Итого по промежуточной аттестации (экзамену)</b>				(51 – 100% по приведенной шкале) 20 – 40 б.
<b>Суммарная оценка по дисциплине:</b> Сумма баллов текущей и промежуточной аттестации				51 – 100 б.

## 5 УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ.

### 5.1 Учебная литература

Основная учебная литература



1. Шкаберина, Г. Ш. Программирование. Основы языка Python : учебное пособие / Г. Ш. Шкаберина, Н. Л. Резова. — Красноярск : СибГУ им. академика М. Ф. Решетнёва, 2018. — 92 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/147450>. — Режим доступа: для авториз. пользователей.

2. Лямин, А. В. Объектно-ориентированное программирование : учебное пособие / А. В. Лямин, Е. Н. Череповская. — Санкт-Петербург : НИУ ИТМО, 2017. — 143 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/110457>. — Режим доступа: для авториз. пользователей.

### Дополнительная учебная литература

1. Бадмаева, Э. С. Сборник задач по программированию : учебное пособие / Э. С. Бадмаева, О. А. Лобсанова. — Улан-Удэ : БГУ, 2019. — 95 с. — ISBN 978-5-9793-1426-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/154267>. — Режим доступа: для авториз. пользователей.

2. Златопольский, Д. М. Основы программирования на языке Python / Д. М. Златопольский. — 2-ое изд., испр. и доп. — Москва : ДМК Пресс, 2018. — 396 с. — ISBN 978-5-97060-641-4. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/131683>. — Режим доступа: для авториз. пользователей.

3. Златопольский, Д. М. 1400 задач по программированию : руководство / Д. М. Златопольский. — Москва : ДМК Пресс, 2020. — 192 с. — ISBN 978-5-97060-827-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/140594>. — Режим доступа: для авториз. пользователей.

## 5.2 Материально-техническое и программное обеспечение дисциплины

Таблица 8 — Информационные технологии и программное обеспечение аудиторных занятий и самостоятельной работы

№п/п	Наименование помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом, в том числе помещения для самостоятельной работы, с указанием перечня основного оборудования, учебно-наглядных пособий и используемого программного обеспечения	Адрес (местоположение) помещений для проведения всех видов учебной деятельности, предусмотренной учебным планом (в случае реализации образовательной программы в сетевой форме дополнительно указывается наименование организации, с которой заключен договор)
1	<p><b>604</b> Учебная аудитория для проведения:</p> <ul style="list-style-type: none"> <li>- занятий лекционного типа;</li> <li>- занятий семинарского (практического) типа;</li> <li>- групповых и индивидуальных консультаций;</li> <li>- текущего контроля и промежуточной аттестации.</li> </ul> <p><b>Специализированная (учебная) мебель:</b> доска меловая, столы, стулья.</p> <p><b>Оборудование для презентации учебного материала:</b> переносное - ноутбук, экран, проектор.</p> <p><b>Используемое программное обеспечение:</b> MS Windows (Microsoft Imagine Premium 3 year по лицензионному договору № 1212/КМР от 12.12.2018</p>	654079, Кемеровская область, г. Новокузнецк, Центральный район, просп. Metallургов, дом № 19

	г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), FoxitReader (свободно распространяемое ПО), Firefox 14 (свободно распространяемое ПО), Яндекс.Браузер (отечественное свободно распространяемое ПО). <b>Интернет с обеспечением доступа в ЭИОС.</b>	
2	<b>602 Компьютерный класс / Лаборатория информационных систем.</b> Учебная аудитория для проведения: - занятий семинарского (практического) типа; - учебных и производственных практик; - курсового проектирования (выполнения курсовых работ); - групповых и индивидуальных консультаций; - самостоятельной работы; - текущего контроля и промежуточной аттестации. <b>Специализированная (учебная) мебель:</b> доска меловая, кафедра, столы компьютерные, стулья. <b>Оборудование:</b> стационарное – компьютеры для обучающихся (17 шт.). <b>Используемое программное обеспечение:</b> MS Windows (Microsoft Imagine Premium 3 year по лицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), LibreOffice (свободно распространяемое ПО), BloodshedDev C++ 4.9.9.2 (свободно распространяемое ПО), FoxitReader (свободно распространяемое ПО), Firefox (свободно распространяемое ПО), Java (бесплатная версия), Microsoft SQL Server 2008 (Microsoft Imagine Premium 3 year по лицензионному договору № 1212/КМР от 12.12.2018 г. до 12.12.2021 г.), SWI-Prolog (свободно распространяемое ПО), UML-диаграммы (бесплатная версия), Denwer (свободно распространяемое ПО), Eclipse (свободно распространяемое ПО), FreePascal (свободно распространяемое ПО), Lazarus (свободно распространяемое ПО), Pascal ABC.NET (свободно распространяемое ПО), Adobe Reader XI (свободно распространяемое ПО), WinDjView (свободно распространяемое ПО), Python (свободно распространяемое ПО), PyCharm Community Edition (бесплатное ПО). <b>Интернет с обеспечением доступа в ЭИОС.</b>	654079, Кемеровская область, г. Новокузнецк, Центральный район, просп. Metallургов, дом № 19

### **5.3 Современные профессиональные базы данных и информационные справочные системы.**

#### **Перечень СПБД и ИСС по дисциплине**

1. Информационная система «Единое окно доступа к образовательным ресурсам. Раздел «Информатика и информационные технологии» - <http://www.window.edu.ru>
2. Веб-сервис для хостинга IT-проектов и их совместной разработки- <https://github.com/>

## 6 ИНЫЕ СВЕДЕНИЯ И (ИЛИ) МАТЕРИАЛЫ

### 6.1 Примерные темы письменных учебных работ

Подобрать по теме «Введение в программирование» 5 проблемных заданий по возрастанию их уровня сложности. Подобрать цифровые образовательные ресурсы в сети «Интернет» в качестве дидактического обеспечения данной темы. Подготовить вопросы, позволяющие определить качество усвоения материала обучающимися. Подготовить задания по этой же теме с учётом индивидуальных особенностей обучающихся (дифференцированных по уровням сложности).

Подобрать по теме «Ветвление» 5 проблемных заданий по возрастанию их уровня сложности. Подобрать цифровые образовательные ресурсы в сети «Интернет» в качестве дидактического обеспечения данной темы. Подготовить вопросы, позволяющие определить качество усвоения материала обучающимися. Подготовить задания по этой же теме с учётом индивидуальных особенностей обучающихся (дифференцированных по уровням сложности).

Подобрать по теме «Циклы» 5 проблемных заданий по возрастанию их уровня сложности. Подобрать цифровые образовательные ресурсы в сети «Интернет» в качестве дидактического обеспечения данной темы. Подготовить вопросы, позволяющие определить качество усвоения материала обучающимися. Подготовить задания по этой же теме с учётом индивидуальных особенностей обучающихся (дифференцированных по уровням сложности).

Подобрать по теме «Работа с числовой информацией» 5 проблемных заданий по возрастанию их уровня сложности. Подобрать цифровые образовательные ресурсы в сети «Интернет» в качестве дидактического обеспечения данной темы. Подготовить вопросы, позволяющие определить качество усвоения материала обучающимися. Подготовить задания по этой же теме с учётом индивидуальных особенностей обучающихся (дифференцированных по уровням сложности).

### 6.2 Примерные вопросы и задания / задачи для промежуточной аттестации

Таблица 9 - Примерные теоретические вопросы и практические задания к экзамену

	Разделы и темы	Примерные теоретические вопросы	Примерные практические задания
2 семестр			
1	Введение в алгоритмизацию и программирование		
1.1	Методология императивного программирования	1. Сформулируйте основные принципы программного управления.	

		2. Сформулируйте суть императивного подхода и основные черты императивных языков.	
1.2	Методология объектно-ориентированного программирования	1. Основные принципы объектно-ориентированного программирования. 2. Дайте определения терминам «класс», «объект», «метод», «свойство».	
1.3	Алгоритмические структуры. Язык блок-схем	1. Перечислите основные алгоритмические структуры. 2. Приведите основные условные обозначения элементов блок-схем.	
1.4	Синтаксис и семантика формального языка.	1. Понятие языков программирования. Понятие синтаксиса языков программирования. 2. Формальные методы описания синтаксиса. Диаграмма Бэкуса-Наура.	1. По данной диаграмме Бэкуса-Наура приведите словесное определение синтаксиса.
2	<b>Структурный подход к программированию</b>		
2.1	Основные конструкции алгоритмических языков	1. Операции ввода/вывода в Python 2. Арифметические операторы в Python	1. Ввести число и вывести его квадрат. 2. Ввести целое число и определить остаток от деления его на пять.
2.2	Простые типы языка программирования	1. Числовые типы данных. 2. Логические типы данных	1. Введите целое число и выведите второй разряд в двоичном представлении данного числа. 2. Введите целое число и поменяйте в его двоичном представлении единицу в самом младшем разряде на 0.
2.3	Структурированные типы языка программирования высокого уровня	1. Понятие строк 2. Понятие списков и кортежей.	1. Введите строку и выведите её без первого и последнего символов. 2. Из исходного списка получите список, содержащий только элементы с нечетными номерами.
2.4	Алгоритмы поиска и сортировки	1. Понятие поиска. Способы поиска. 2. Понятие сортировки. Способы сортировки.	1. Написать алгоритм линейного поиска. 2. Написать алгоритм бинарного поиска. 3. Написать алгоритм сортировки обменом. 4. Написать алгоритм сортировки выбором. 5. Написать алгоритм сортировки вставкой.
2.5	Технология структурного программирования	1. Основные конструкции структурного программирования. 2. Нисходящее и восходящее программирование.	
<b>3 семестр</b>			
1	<b>Модульное программирование. Программирование абстрактных типов данных</b>		
1.1	Функции.	1. Понятие функции. Вызов функции. 2. Создание функции	1. Напишите функцию, определяющую, является ли целое число простым. 2. Напишите функцию, определяющую n-й член последовательности Фибоначчи.

1.2	Рекурсия	1. Понятие рекурсии. Применение рекурсии. 2. Виды рекурсии.	1. Напишите программу, вычисляющую факториал числа. 2. Напишите программу, определяющую порядок переноса дисков в головоломке «Ханойская башня»
1.3	Внутренние функции. Замыкания. Анонимные функции.	1. Понятие внутренней функции. Понятие замыкания. 2. Анонимные функции. Использование анонимной функции.	1. Напишите замыкание, возводящее число в заданную на момент определения степень. 2. Напишите анонимную функцию для упорядочивания списка строк по длине строки.
1.4	Генераторы. Декораторы. Пространство имен и область определения.	1. Понятие генератора. Назначение генератора. 2. Понятие декоратора. Назначение декоратора.	1. Напишите генератор, создающий последовательность простых чисел. 2. Напишите декоратор, выводящий список параметров, переданных функции.
1.5	Обработка исключений	1. Обработка исключений с помощью оператора try – except. 2. Создание собственных исключений.	1. Создайте обработчик исключений для перехвата ошибки при вводе целого числа. 2. Создайте функцию, возбуждающую исключение при выходе параметра за допустимый диапазон.
1.6	Модульное программирование в python.	1. Понятие модуля. Создание модуля. 2. Включение модуля в основную программу	1. Создайте модуль, хранящий каталог продукции, включающий артикул, наименование и цену, и позволяющий получить наименование и цену по данному артикулу, добавить в каталог продукцию с указанными артикулом, наименованием и ценой, а также удалить продукцию с указанным артикулом. 2. Создайте модуль, содержащий функции, вычисляющие площади таких геометрических фигур как круг, треугольник и прямоугольник.
1.7	Динамические структуры данных (абстрактные типы данных): список, стек, очередь, графы.	1. Понятие списка, стека и очереди. 2. Реализация списка, стека и очереди в Python	1. Реализуйте очередь для хранения информации о записи клиентов в массажный салон. 2. Реализуйте список для хранения информации о порядке движения автобуса по маршруту.
1.8	Графы.	1. Понятие графа 2. Способы представления графа	1. Реализуйте алгоритм обход графа в ширину. 2. Реализуйте алгоритм обход графа в глубину.
1.9	Файлы	1. Порядок работы с текстовыми файлами Порядок работы с XML-файлами	1. Напишите программу, выводящую заданную информацию в текстовый файл. 2. Напишите программу, выводящую заданную информацию в XML-файл.
1.10	Сетевое взаимодействие.	1. Понятие сети передачи данных. Понятие сетевых протоколов. 2. Стек протоколов TCP/IP	1. Реализуйте программу, подключающуюся к сайту по заданному адресу и выводящую содержание веб-страницы в текстовом виде. 2. Напишите программу, ожидающую подключение по определенному порту и

			передающую при установлении соединения на компьютер-клиент текст «Hello».
2	Объектно-ориентированное программирование		
2.1	Методология и принципы объектно-ориентированного программирования.	1. Понятие объектно-ориентированного программирования. 2. Основные принципы объектно-ориентированного программирования.	
2.2	Объектно-ориентированное программирование на языке Python.	1. Синтаксис определения классов. Создание объектов. 2. Реализация наследования.	1. Создайте класс для хранения информации о координатах вершин треугольника на плоскости, а также вычисления его периметра и площади. 2. Создайте базовый класс для хранения такой информации об автомобиле, как его регистрационный номер и цвет кузова. Создайте два производных класса: один для хранения информации о грузовом автомобиле: к полям добавьте грузоподъемность. И второй — для хранения информации об автобусах, в него добавьте данные о количестве сидячих и стоячих мест.