

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Беловский институт (филиал) федерального государственного бюджетного
образовательного учреждения высшего образования
«Кемеровский государственный университет» (БИФ КемГУ)
Кафедра экономических наук и информационных технологий



УТВЕРЖДАЮ
Директор БИФ КемГУ

В. А. Саркисян

«27» февраля 2019г.

**Аннотация
рабочей программы дисциплины
Программирование**

Направление подготовки

**02.03.02 Фундаментальная информатика и информационные
технологии**

(цифр, название направления)

Направленность (профиль) подготовки

Открытые информационные системы

Форма обучения

очно-заочная

(очная, заочная, очно-заочная и др.)

1. Цели и задачи дисциплины

Цели преподавания дисциплины:

Целью изучения дисциплины «Программирование» является формирование способности осваивать методики использования программных средств для решения практических задач; получение знаний и навыков программирования на языке высокого уровня, самостоятельное приобретение с помощью информационных технологий и использование в практической деятельности новых знаний и умений.

Основными **задачами** дисциплины является обучение студентов:

- 1) анализу и алгоритмизации решаемых задач;
- 2) оформлению решения задачи в графическом виде (в виде схем алгоритмов);
- 3) программированию любого алгоритма, задачи, метода;
- 4) проектированию и отладке достаточно сложных программ;
- 5) тестированию и оптимизации разработанного программного продукта.

2. Требования к результатам освоения дисциплины (табл. из п.1 РП)

Код и наименование компетенции	Код и наименование индикатора достижения компетенции	Планируемые результаты обучения
УК-1 Способен осуществлять поиск, критический анализ и синтез информации, применять системный подход для решения поставленных задач	ИУК-1.1. Знает принципы сбора, отбора и обобщения информации. ИУК-1.2. Умеет соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности. ИУК-1.3. Имеет практический опыт работы с информационными объектами и сетью Интернет, опыт научного поиска, опыт библиографического разыскания, создания научных текстов.	знать: принципы сбора, отбора и обобщения информации в области алгоритмизации, языков и технологии программирования, пригодных для практического применения в области информационных систем и технологий; основные платформы, технологии и инструментальные программно-аппаратные средства для реализации информационных систем; основные понятия, определения и факты теории сложности алгоритмов. уметь: соотносить разнородные явления и систематизировать их в рамках избранных видов профессиональной деятельности, применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий; осуществлять выбор платформ и инструментальных программно- аппаратных средств для реализации информационных систем, применять современные технологии реализации информационных систем; основные методы теории сложности алгоритмов; оценивание сложности алгоритмов. владеть (практический опыт): работы с информационными объектами и сетью Интернет, опыт научного поиска, опыт библиографического разыскания, создания научных текстов при выработке навыков программирования, отладки и тестирования прототипов программно-технических комплексов задач; владения технологиями и инструментальными программно-аппаратными средствами для реализации информационных систем; навыками решения практических задач анализа алгоритмов; методами анализа сложности алгоритмов.

<p>ОПК-1 Способен применять фундаментальные знания, полученные в области математических и (или) естественных наук, и использовать их в профессиональной деятельности</p>	<p>ИОПК-1.1. Знает основные положения и концепции в данной области, базовые теории и основы материала, теории коммуникации; знает основную терминологию. ИОПК-1.2. Умеет осуществлять первичный сбор и анализ материала, интерпретировать различные объекты. ИОПК-1.3. Имеет практический опыт работы по решению стандартных задач и применяет его в профессиональной деятельности.</p>	<p>знать:</p> <ul style="list-style-type: none"> - конструктивные компоненты и структуру компьютерных программ; - объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов; - методы модульного программирования; основные программные среды и средства разработки программ на языке C++. <p>уметь:</p> <ul style="list-style-type: none"> - выбирать и использовать базовые структуры данных для организации сложных управляющих и информационных структур; - обосновывать выбор программного обеспечения и разрабатывать концептуальную и логическую модель данных. <p>владеть:</p> <ul style="list-style-type: none"> - навыками описания основных базовых конструкций; - математическими средствами исследования практической приемлемости конкретных алгоритмов.
<p>ОПК-2 Способен применять компьютерные/суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности</p>	<p>ИОПК-2.1. Знает основные положения и концепции в области программирования, архитектуру языков программирования, теории коммуникации, знает основную терминологию, знаком с содержанием Единого Реестра Российских программ. ИОПК-2.2. Умеет анализировать типовые языки программирования, составлять программы. ИОПК-2.3. Имеет практический опыт решения задач анализа, интеграции различных типов программного обеспечения, анализа типов коммуникаций.</p>	<p>знать:</p> <ul style="list-style-type: none"> - технологию работы на персональной ЭВМ (ПЭВМ), правила и приемы диалоговой работы на ПЭВМ при программировании типовых задач; - способы испытания и отладки программ; - общие принципы построения алгоритмов, основные алгоритмические конструкции; - понятие системы программирования; - основные элементы процедурного языка программирования, структуру программы, операторы и операции, управляющие структуры, структуры данных, файлы, классы памяти; - подпрограммы, составление библиотек программ; - объектно-ориентированную модель программирования, понятие классов и объектов, их свойств и методов. <p>уметь:</p> <ul style="list-style-type: none"> - использовать приемы и методы разработки программного обеспечения на основе современного стиля программирования; - самостоятельно осуществлять постановку и спецификацию задачи для решения на ПЭВМ; - формализовать поставленную задачу; - использовать языки программирования; - строить логически правильные и эффективные программы; - использовать нисходящий метод для решения задач. <p>владеть (иметь навыки):</p> <ul style="list-style-type: none"> - математическими средствами исследования практической приемлемости конкретных алгоритмов; - навыками работы в среде программирования (составление, отладка и тестирование программ; разработка и использование интерфейсных объектов).
<p>ОПК-3 Способен к разработке алгоритмических и программных решений в области</p>	<p>ИОПК-3.1. Знает методы теории алгоритмов, методы системного и прикладного программирования, основные положения и концепции в</p>	<p>знать:</p> <ul style="list-style-type: none"> - методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий;

<p>системного и прикладного программирования, математических, информационных и имитационных моделей, созданию информационных ресурсов глобальных сетей, образовательного контента, прикладных баз данных, тестов и средств тестирования систем и средств на соответствие стандартам и исходным требованиям</p>	<p>области математических, информационных и имитационных моделей, ИОПК-3.2. Умеет соотносить знания в области программирования, интерпретацию прочитанного, определять и создавать информационные ресурсы глобальных сетей, образовательного контента, средств тестирования систем. ИОПК-3.3. Имеет практический опыт применения разработки программного обеспечения.</p>	<p>основные понятия, определения и факты теории сложности алгоритмов. уметь: - применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий; - проектировать, реализовывать на языке высокого уровня и тестировать простые и средней сложности алгоритмы решения математических, технических и организационных задач; основные методы теории сложности алгоритмов; оценивание сложности алгоритмов; подтверждать корректность работы программной системы путем организации модульного тестирования и представления результатов тестов; разрабатывать и тестировать программные компоненты с использованием этих средств. владеть (иметь навыки): программирования, отладки и тестирования прототипов программно-технических комплексов задач; решения практических задач анализа алгоритмов; методами анализа сложности алгоритмов.</p>
--	---	---

3. Общая трудоемкость дисциплины – 11 з.е., 396 часов

4. Содержание дисциплины (дидактические единицы)

Определение, виды, свойства, измерение информации. Информационные технологии. Аппаратное, программное и алгоритмическое обеспечение информационных технологий. Структура компьютера. Архитектура Джона Фон Неймана. Многоядерные и многопроцессорные вычислительные системы.

Программирование как научная дисциплина, как искусство, как профессиональная деятельность. Пользователи, программисты, системные администраторы. Прикладное, системное, научно–исследовательское программирование.

Основные проблемы программирования. Среда программирования. Состав и назначение компонент среды программирования. Классификация ошибок в программе: синтаксические, семантические ошибки, ошибки времени выполнения.

Языки программирования высокого и низкого уровня, исторический экскурс. Алфавит, синтаксис, семантика, прагматика языка программирования. Семантические ошибки языков программирования. Средства определения синтаксиса: расширенные формулы Бэкуса–Наура (РБНФ). Классификация языков программирования по уровню абстракции. Обзор основных элементов языка программирования высокого уровня. Переменные. Константы, типизированные константы. Типы данных. Объявление, использование типов, переменных, констант. Формальная и фактическая эквивалентность типов. Совместимость типов. Преобразование типов. Элементарные конструкции.

Выражения. Классификация и приоритет операций. Построение и вычисление выражений. Операторы. Классификация операторов. Простые операторы. Структурированные операторы. Базовые структуры языка программирования.

Понятие структурного программирования. Теорема о структурном программировании. Подпрограммы. Процедуры и функции. Спецификация подпрограммы. Формальные и фактические параметры подпрограммы. Способы передачи фактических параметров в подпрограмму. Модульное программирование. Состав подпрограммы.

Структура модуля программы. Объявления, области действия и видимости объявлений. Локальные и глобальные переменные, константы, типы. Побочный эффект подпрограммы. Распределение памяти программы: сегмент данных, сегмент стека, куча.

Файловые типы данных. Классификация файловых типов. Стандартные подпрограммы для работы с файлами. Ссылочные типы данных и указатели. Статические и динамические переменные программы. Стандартные подпрограммы для работы с указателями.

Интуитивное определение алгоритма. Основные свойства алгоритмов. Основные методы разработки алгоритмов. Формальное определение алгоритма. Рекурсивные алгоритмы. Виды рекурсии. Простейшие примеры рекурсивных алгоритмов. Случаи оправданности, случаи недопустимости рекурсии.

Метод динамического программирования.

Объектно–ориентированное программирование (ООП). Понятие класса. Инкапсуляция, наследование, полиморфизм. Видимость полей и методов объекта. Механизмы раннего и позднего связывания. Статические и виртуальные методы. Иерархии классов.

Внутренняя сортировка (сортировка массивов). Понятие сложности алгоритма сортировки. Основные алгоритмы сортировки: сортировка прямым выбором, сортировка вставками, пузырьковая сортировка, шейкерная сортировка, сортировка Шелла, быстрая сортировка, сортировка слиянием.

Понятие и классификация структур данных. Линейные структуры данных. Стеки, очереди, деки. Связные списки: однонаправленный список, двунаправленный список, циклический список.

Рекурсивные структуры данных. Сильно ветвящиеся деревья. Двоичные деревья. Основные алгоритмы работы с двоичными деревьями.