

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Факультет компьютерных наук и информационных технологий

УТВЕРЖДАЮ
декан факультета КНиИТ,
к.ф.м.н. С.В. Миронов
«23» 20 23 г.



Рабочая программа дисциплины
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ПРОГРАММИРОВАНИЕ

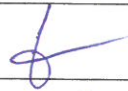
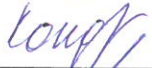

Направление подготовки бакалавриата
09.03.03 – Прикладная информатика

Профиль подготовки бакалавриата
Прикладная информатика в экономике

Квалификация (степень) выпускника
Бакалавр

Форма обучения
очная

Саратов,
2023

Статус	ФИО	Подпись	Дата
Преподаватель-разработчик	Хамутова М.В.		
Председатель НМК	Кондратова Ю.Н.		
Заведующий кафедрой	Миронов С.В.		
Специалист учебного управления			

1. Цели освоения дисциплины

Цель освоения дисциплины «Информационные технологии и программирование» — формирование общих представлений об основных понятиях, методах и принципах разработки программных продуктов, развитие у студентов алгоритмического мышления, способностей к формализации, умение анализировать, находить и исправлять ошибки в собственном и чужом программном коде, развитие у студентов компетенций в области разработки и применения программных продуктов в профессиональной деятельности.

2. Место дисциплины в структуре ООП

Данная учебная дисциплина относится к обязательной части Блока 1 «Дисциплины (Модули)» ООП направлена на формирование у обучающихся общепрофессиональных компетенций.

Для успешного усвоения данной дисциплины необходимы компетенции, сформированные у студентов в результате обучения в средней образовательной школе при изучении дисциплин, связанных с информатикой и программированием.

Компетенции, сформированные студентами при изучении данной дисциплины, используются при изучении следующих дисциплин «Основы объектно-ориентированного программирования», «Базы данных», «Информационные системы и технологии», а также при написании курсовых и выпускных квалификационных работ.

3. Результаты обучения по дисциплине

Код и наименование компетенции	Код и наименование индикаторов достижения компетенции	Результаты обучения
<p>ОПК-2. Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности.</p>	<p>1.1_Б.ОПК-2. Понимает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы реализации таких процессов и методов</p> <p>2.1_Б.ОПК-2. Выбирает и использует современные информационно-коммуникационные и интеллектуальные технологии, инструментальные среды, программно-технические платформы и программные средства, в том числе отечественного производства, для решения задач профессиональной деятельности.</p> <p>3.1_Б.ОПК-2. Анализирует профессиональные задачи, выбирает и использует подходящие ИТ-решения.</p>	<p>Знает</p> <ul style="list-style-type: none"> — основные технологии программирования; — основные форматы представления данных; — синтаксическую и семантическую организацию языка программирования, на котором записан исходный код; — основные методы поиска и анализа информации с использованием современных информационных технологий; — методы и средства проверки работоспособности программного обеспечения. <p>Умеет</p> <ul style="list-style-type: none"> — описать предметную область поставленной задачи, используя основные концепции информатики. <p>Владеет</p> <ul style="list-style-type: none"> — навыками сравнения фактического и ожидаемого результатов; — навыками оценки и согласования сроков выполнения поставленных задач; — технологией поиска и обмена информацией в глобальных и локальных сетях.

<p>ОПК-7. Способен разрабатывать алгоритмы и компьютерные программы, пригодные для практического применения.</p>	<p>1.1_Б.ОПК-7. Понимает процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы реализации таких процессов и методов.</p> <p>2.1_Б.ОПК-7. Использует современные языки программирования для разработки алгоритмов и компьютерных программ, пригодных для практического применения, поддерживает базы данных и информационные хранилища.</p> <p>3.1_Б.ОПК-7. Применяет современные программные среды разработки информационных систем и технологий, методы отладки и тестирования, читает коды программных продуктов, написанные на освоенных языках программирования, и вносит требуемые изменения.</p> <p>4.1_Б.ОПК-7. Готов самостоятельно осваивать новые для себя языки программирования, среды разработки информационных систем и технологии.</p> <p>5.1_Б.ОПК-7. Анализирует профессиональные задачи, разрабатывает подходящие ИТ-решения.</p>	<p>Знает</p> <ul style="list-style-type: none"> — основные этапы решения задач на персональном компьютере; — методы и приемы отладки программного кода; — принципы документирования исходного программного кода; — основные методы поиска и анализа информации с использованием современных информационных технологий. — основные среды разработки для языка с++. <p>Умеет</p> <ul style="list-style-type: none"> — использовать существующие типовые решения и шаблоны проектирования программного обеспечения; — применять нормативные документы, определяющие требования к оформлению программного кода; — использовать методы и приемы формализации задач; — использовать современные программные среды разработки; — выявлять ошибки в программном коде; — устанавливать на компьютере необходимую среду разработки. <p>Владеет</p> <ul style="list-style-type: none"> — навыками приведения наименований переменных, функций, классов, структур данных и файлов в соответствие с заданными требованиями; — навыками использования современных отладчиков программного кода; — навыками применения библиотек программных модулей, шаблонов при разработке программного продукта; — навыками анализа и проверки исходного программного кода; — навыками выбора необходимой среды разработки с учетом установленной операционной системы.
--	---	--

4. Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 8 зачетных единиц, 288 часов.

№ п/п	Раздел дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Формы текущего контроля успеваемости (по неделям семестра) Формы промежуточной аттестации (по семестрам)
				Лекции	Лаб. работы		Практ. занятия	СР	контроль	
					Общая трудоемкость	Из них практическая подготовка				
1	История развития информационных технологий и программирования	1	1	2				2		контрольная работа на 15 неделе
2	Основные парадигмы программирования. Основные языки программирования	1	2	2				2		контрольная работа на 15 неделе
3	Основные конструкции языков программирования высокого уровня	1	3-6	8	8	4	8	4		контрольная работа на 15 неделе
4	Основные типы данных языков программирования высокого уровня на примере C++	1	7-11	10	14	7	14	4		контрольная работа на 15 неделе
5	Структурное программирование на примере языка программирования C++	1	12-18	14	14	7	14	6		контрольная работа на 15 неделе
итого за 1 семестр				36	36	18	36	18	54	экзамен
6	Абстрактные типы данных. Списки	2	1,2	4	4	2	4	4		контрольная работа на 14 неделе
7	Абстрактные типы данных. Деревья и графы	2	3,4	4	4	2	2	6		контрольная работа на 14 неделе
8	Понятие алгоритмов. Алгоритмы сортировки и поиска	2	5-8	8	8	4	4	6		контрольная работа на 14 неделе

9	Обобщенное программирование на примере языка C++. Стандартная библиотека шаблонов	2	9-12	8	8	4	4	6		контрольная работа на 14 неделе
10	Разработка, тестирование, документирование программного продукта	2	13-16	8	8	4	2	6		контрольная работа на 14 неделе
итого за 2 семестр				32	32	16	16	28		зачет
всего				68	68	34	52	46	54	

Содержание дисциплины

История развития информационных технологий и программирования

Понятие информатики. Понятие информационных технологий. Виды информационных технологий. История развития информатики. Теории и факты развития информатики. Современные концепции и принципы, связанные с информатикой, как с наукой.

Ученые, связанные с развитием информационных технологий. Современное состояние информационных технологий и программирования. Стандарты в области информационных технологий и программирования.

Основные парадигмы программирования

Основные языки программирования. Понятие парадигмы программирования. Современные парадигмы программирования. Их достоинства и недостатки.

Языки программирования низкого уровня. Языки программирования высокого уровня. История развития языков программирования.

Основные конструкции языков программирования высокого уровня

Язык программирования C++. Директивы препроцессора. Сборка и компиляция проекта.

Основные конструкции: алфавит, ключевые слова, идентификаторы, блоки. Операции, приоритет операций.

Операторы. Простой и составной оператор. Операторы присвоения.

Операторы ветвления. Операторы цикла. Операторы управления.

Переменная. Время жизни переменной, область видимости переменной.

Основные типы данных языков программирования высокого уровня на примере C++

Понятие типа данных. Базовые типы данных: целочисленные, логические, вещественные, символьные. Приведение типов данных.

Составные типы данных: массивы, строки, перечисления, объединения, структуры, указатели.

Понятие потока. Потoki ввода-вывода, файловые потоки, строковые

потоки.

Структурное программирование на примере языка программирования C++

Понятие функции в языке C++. Функции, возвращающие и не возвращающие значение. Глобальные и локальные переменные. Формальные и фактические переменные. Способы передачи параметров в функцию: по значению, по указателю и по ссылке.

Понятие рекурсии. Математические рекурсивные функции. Примеры рекурсивных функций. Рекурсия и метод «разделяй и властвуй». Реализация бэктрекинга (backtracking) посредством рекурсии.

Перегрузка функций. Шаблоны функций. Исключения. Виды исключений.

Абстрактные типы данных. Списки

Понятие абстрактных типов данных. Примеры абстрактных типов данных.

Способы представления абстрактных типов данных.

Абстрактный тип данных — список. Виды списков. Стек, очередь, односвязный и двусвязный список, кольцевой список.

Абстрактные типы данных. Деревья и графы

Абстрактный тип данных — деревья. Виды деревьев. Обходы деревьев. Использование деревьев для алгоритмов поиска и сортировки.

Абстрактный тип данных — граф. Виды графов. Способы представления графов. Обходы графов.

Хеш-таблицы. Открытое и закрытое хеширование.

Понятие алгоритмов. Алгоритмы сортировки и поиска

Понятие алгоритма. Свойства алгоритмов. Виды алгоритмов.

Сложность алгоритмов. Временная сложность алгоритмов. Оценка временной сложности.

Методы разработки алгоритмов. Алгоритмы сортировки и поиска. Элементарные алгоритмы сортировки. Сортировки по методу декомпозиции. Сортировки за линейное время.

Алгоритмы поиска в строке.

Обобщенное программирование на примере языка C++.

Стандартная библиотека шаблонов

Парадигма обобщенного программирования. Стандартная библиотека шаблонов.

Контейнеры. Примеры контейнеров, их преимущества и недостатки. Итераторы. Виды итераторов.

Алгоритмы. Виды алгоритмов. Применение алгоритмов для различных контейнеров.

Разработка, тестирование, документирование программного продукта

Постановка задачи. Формализация предметной области программного проекта. Построение модели. Выбор языка программирования, разработка спецификации для компонентов программного продукта.

Проектирование структуры программы. Отладка программы. Чтение, понимание и выделение главной идеи прочитанного исходного кода

Нисходящее тестирование. Кодирование и документирование программы. Единая система программной документации. Взаимодействие с другими пользователями и разработчиками.

План практических занятий

На практических занятиях студенты проходят короткий опрос для контроля текущей успеваемости, разрабатывают алгоритмы решения предложенных заданий, пишут код программы, разбирают допущенные ошибки, разбирают код программ, реализованных в рамках самостоятельной работы. Задания для решения в аудитории приведены в ФОС, раздел «Задания для практических занятий», задания для домашней работы приведены в ФОС, раздел «Задания для самостоятельной работы».

№ занятия	Тема	Задания для решения в аудитории	Задания для домашней работы
1–2	Основные конструкции языков программирования высокого уровня	I.1–I.12	III.1–III.5
3–4	Основные типы данных языков программирования высокого уровня на примере C++	II.1–II.18	IV.1–IV.5
5–6	Структурное программирование на примере языка программирования C++	III.1–III.8	V.1–V.9
7	Контрольная работа №1		
8–9	Абстрактные типы данных. Списки	IV.1–IV.9	VI.1–VI.5
10–11	Абстрактные типы данных. Деревья и графы	V.1–V.8	VII.1–VII.5
12–13	Понятие алгоритмов. Алгоритмы сортировки и поиска	VI.1–VI.8	VIII.1–VIII.5
14–15	Обобщенное программирование на примере языка C++. Стандартная библиотека шаблонов	VII.1–VII.7	IX.1
16	Контрольная работа №2		
17	Разработка, тестирование, документирование программного продукта	VIII.1	X.1

План лабораторных занятий

На лабораторных занятиях студенты разрабатывают алгоритмы поставленных заданий, реализуют разработанные алгоритмы на языке программирования C++, проверяют корректность работы реализованных программ и объясняют ход решения преподавателю.

Задания для лабораторного практикума приведены в ФОС, раздел «Задания для лабораторных занятий».

№ занятия	Тема	Задания для лабораторного практикума
1–4	Основные конструкции языков программирования высокого уровня	I.1–I.20, II.1–II.20, III.1–III.20, IV.1–IV.20
5–9	Основные типы данных языков программирования высокого уровня на примере C++	V.1–V.20, VI.1–VI.20, VII.1–VII.20, VIII.1–VIII.20, IX.1–IX.20
10–14	Структурное программирование на примере языка программирования C++	X.1–X.20, XI.1–XI.20
15–18	Абстрактные типы данных. Списки	XII.1–XII.20, XIII.1–XIII.20
19–23	Абстрактные типы данных. Деревья и графы	XIV.1–XIV.20, XV.1–XV.20
24–28	Понятие алгоритмов. Алгоритмы сортировки и поиска	XVI.1–XVI.20, XVII.1–XVII.20
29–33	Обобщенное программирование на примере языка C++. Стандартная библиотека шаблонов	XVIII.1–XVIII.20
34	Разработка, тестирование, документирование программного продукта	XIX.1

5. Образовательные технологии, применяемые при освоении дисциплины

Предусматривается широкое использование в учебном процессе активных и интерактивных форм проведения занятий: дистанционные технологии, мультимедийные презентации теоретического материала.

В рамках практической подготовки по данной дисциплине используются практические задания, выполнение которых направлено на формирование профессиональных навыков в области разработки информационных систем и программных продуктов. Примеры практических заданий приведены в фондах оценочных средства.

При обучении лиц с ограниченными возможностями здоровья и инвалидов используются подходы, способствующие созданию безбарьерной образовательной среды: технологии дифференциации и индивидуализации обучения, увеличивается время на самостоятельное освоение материала. Используется сочетание разных форм и способов передачи учебной информации: вербальный, невербальный, с использованием средств визуализации информации (презентации) и разных способов отчетности (письменно, устно, с использованием электронных дистанционных технологий).

6. Учебно-методическое обеспечение самостоятельной работы студентов. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

В рамках самостоятельной работы студенты самостоятельно прорабатывают практические примеры для наилучшего усвоения методов и алгоритмов теоретического курса, оценивают предложенные преподавателем

примеры программ и собственную работу по определенным критериям.

Фонд оценочных средств дисциплины включает в себя задания для самостоятельной работы (методические рекомендации, критерии оценивания и список заданий), задания для практических занятий (методические рекомендации, критерии оценивания и список заданий), задания для лабораторных занятий (методические рекомендации, критерии оценивания и список заданий) по соответствующим разделам, задания для контрольной работы (методические рекомендации, критерии оценивания и варианты заданий), контрольные вопросы, вопросы для проведения промежуточной аттестации.

Текущий контроль проводится в форме устных опросов на лекционных и практических занятиях, разбора и обсуждения решаемых задач на практических занятиях, контрольной работы по теме.

7. Данные для учета успеваемости студентов в БАРС

Таблица максимальных баллов по видам учебной деятельности

1	2	3	4	5	6	7	8	9
Семестр	Лекции	Лабораторные занятия	Практические занятия	Самостоятельная работа	Автоматизированное тестирование	Другие виды учебной деятельности	Промежуточная аттестация	Итого
1	5	30	5	10	0	10	40	100
2	5	30	5	10	0	10	40	100

Программа оценивания учебной деятельности студента

Семестр 1

Лекции. Посещаемость, опрос, активность и др. за один семестр — от 0 до 5 баллов.

Лабораторные занятия. Контроль выполнения лабораторных заданий в течение одного семестра — от 0 до 30 баллов.

Практические занятия. Контроль выполнения практических заданий в течение одного семестра — от 0 до 5 баллов.

Самостоятельная работа. Выполнение заданий для самостоятельной работы — от 0 до 10 баллов.

Автоматизированное тестирование. Не предусмотрено.

Другие виды учебной деятельности. Контрольная работа №1 (от 0 до 5 баллов). Изучение факультативного материала, успешное выступление с докладом по теме факультативного материала, своевременность выполнения текущих заданий (от 0 до 5 баллов). Таким образом в течение семестра студент может получить от 0 до 10 баллов.

Промежуточная аттестация. Промежуточная аттестация состоит из экзамена. Методика оценивания знаний обучающихся по дисциплине «Информационные технологии и программирование» в ходе промежуточной

аттестации

25–40 баллов — ответ на «отлично»:

Ответ студента содержит:

– глубокое знание программного материала, а также основного содержания и новаций лекционного курса по сравнению с учебной литературой;

– знание концептуально-понятийного аппарата всего курса;

– знание монографической литературы по курсу,

– также свидетельствует о способности:

– самостоятельно критически оценивать основные положения курса;

– увязывать теорию с практикой.

15–24 баллов — ответ на «хорошо»:

Ответ студента свидетельствует:

– о полном знании материала по программе;

– о знании рекомендованной литературы,

– а также содержит в целом правильное, но не всегда точное и аргументированное изложение материала.

10–14 баллов — ответ на «удовлетворительно»:

Ответ студента содержит:

– поверхностные знания важнейших разделов программы и содержания лекционного курса;

– затруднения с использованием научно-понятийного аппарата и терминологии курса;

– стремление логически четко построить ответ, а также свидетельствует о возможности последующего обучения.

0–9 баллов — неудовлетворительный ответ:

Студент имеет существенные пробелы в знании основного материала по программе и допускает принципиальные ошибки при изложении материала.

Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 1-й семестр по дисциплине «Информационные технологии и программирование» составляет 100 баллов.

Таблица пересчета полученной студентом суммы баллов по дисциплине «Информационные технологии и программирование» в оценку (экзамен)

от 91 баллов и более	«отлично»
от 75 до 90 баллов	«хорошо»
от 61 до 74 баллов	«удовлетворительно»
меньше 61 баллов	«неудовлетворительно»

Семестр 2

Лекции. Посещаемость, опрос, активность и др. за один семестр — от 0 до 5 баллов.

Лабораторные занятия. Контроль выполнения лабораторных заданий в течение одного семестра — от 0 до 30 баллов.

Практические занятия. Контроль выполнения практических заданий в течение одного семестра — от 0 до 5 баллов.

Самостоятельная работа. Выполнение заданий для самостоятельной работы — от 0 до 10 баллов.

Автоматизированное тестирование. Не предусмотрено.

Другие виды учебной деятельности. Контрольная работа №2 (от 0 до 5 баллов). Изучение факультативного материала, успешное выступление с докладом по теме факультативного материала, своевременность выполнения текущих заданий (от 0 до 5 баллов). Таким образом в течение семестра студент может получить от 0 до 10 баллов.

Промежуточная аттестация. Промежуточная аттестация состоит из зачета.

Методика оценивания знаний обучающихся по дисциплине «Информационные технологии и программирование» в ходе промежуточной аттестации

Промежуточная аттестация проходит в виде устной беседы по вопросам к зачету. Ответ на «зачтено» оценивается от 20 до 40 баллов. Ответ на «не зачтено» оценивается от 0 до 19 баллов. Таким образом, максимально возможная сумма баллов за все виды учебной деятельности студента за 2-й семестр по дисциплине «Информационные технологии и программирование» составляет 100 баллов.

Таблица пересчета полученной студентом суммы баллов по дисциплине «Информационные технологии и программирование» в оценку (зачет)

70-100 баллов	«зачтено»
0-69 баллов	«не зачтено»

8. Учебно-методическое и информационное обеспечение дисциплины «Информационные технологии и программирование»

Литература:

1. Шень А.Х. Методы построения алгоритмов: практикум, Москва, Саратов: Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 335 с. [Электронный ресурс] <https://www.iprbookshop.ru/89445.html>

2. Огнева М.В., Кудрина Е.В. Программирование на языке С++: практический курс: учебное пособие для вузов, Москва: Издательство Юрайт, 2023. 335 с. [Электронный ресурс] <https://urait.ru/bcode/515142>

3. Конова Е.А., Поллак Г.А. Алгоритмы и программы. Язык С++: учебное пособие, Санкт-Петербург: Лань, 2022. 384 с. [Электронный ресурс] <https://e.lanbook.com/book/140730>

Интернет-ресурсы:

4. <https://course.sgu.ru/course/view.php?id=1025> – Курс «Информатика и программирование мехмат 1 семестр» в системе moodle

5. <https://course.sgu.ru/course/view.php?id=1141> – Курс «Информатика и программирование мехмат 2 семестр» в системе moodle

6. Справочник по языку С++ [Электронный ресурс] <https://learn.microsoft.com/ru-ru/cpp/cpp/?view=msvc-170>

Программное обеспечение:

7. Лицензионное программное обеспечение: VisualStudio 20XX (поставляемое по лицензии DreamSparkPremiumElectronicDelivery)

9. Материально-техническое обеспечение дисциплины «Информационные технологии и программирование»

Для проведения лекционных занятий необходимы: маркерная доска, мультимедийный проектор, компьютер с доступом к сети Интернет.

Для проведения семинарских занятий необходимы: маркерная доска, мультимедийный проектор, компьютер с доступом к сети Интернет.

Для проведения лабораторных занятий необходим компьютерный класс с установленным программным обеспечением и доступом к сети Интернет.

Реализация практической подготовки в рамках учебных занятий запланирована на базе механико-математического факультета СГУ.

Программа составлена в соответствии с требованиями ФГОС ВО для направления 09.03.03 «Прикладная информатика» (профиль подготовки «Прикладная информатика в экономике»).

Авторы: доцент, кандидат технических наук М. В. Хамутова

Программа одобрена на заседании кафедры математической кибернетики и компьютерных наук 04.05.2023 протокол №21