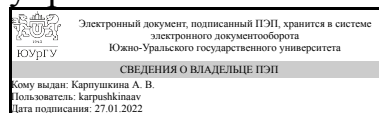


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Директор института
Высшая школа экономики и
управления



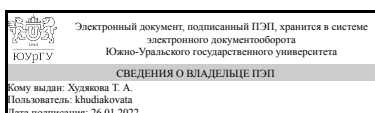
А. В. Карпушкина

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.12.01 Основы программирования
для направления 09.03.03 Прикладная информатика
уровень Бакалавриат
форма обучения очная
кафедра-разработчик Цифровая экономика и информационные технологии

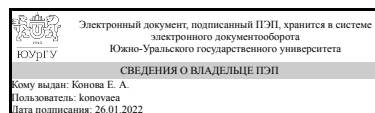
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.03 Прикладная информатика, утверждённым приказом Минобрнауки от 19.09.2017 № 922

Зав.кафедрой разработчика,
Д.ЭКОН.Н., доц.



Т. А. Худякова

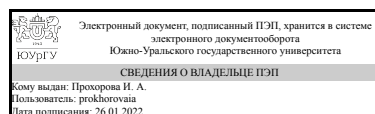
Разработчик программы,
старший преподаватель



Е. А. Конова

СОГЛАСОВАНО

Руководитель направления
к.техн.н., доц.



И. А. Прохорова

1. Цели и задачи дисциплины

Целью освоения дисциплины «Основы программирования» в соответствии с требованием ФГОС, является изучение теоретических основ программирования, изучение основ языка C++ и приобретение навыков разработки, отладки программ в современной высокотехнологичной среде разработчика. В результате изучения дисциплины студент должен приобрести необходимые компетенции, а именно: понимать принципы работы современных информационных технологий и программных средств и использовать их при решении задач профессиональной деятельности, разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий.

Краткое содержание дисциплины

Дисциплина «Основы программирования» - формирует базовые знания, умения и навыки студентов по алгоритмизации, концепции типов данных и языков программирования, применению технологий программирования, работе с документацией. В разделах курса изучаются • введение в понятийный аппарат: интегрированные среды разработчика, данные и алгоритмы, модульная структура приложений; • концепция типов данных; • основные алгоритмические конструкции языков программирования, программирование основных алгоритмов; • принцип модульности при разработке приложений: функции, основные принципы модульности; • конструируемые типы данных: массивы, указатели, структуры, строки текста – синтаксические правила описания и семантика; • парадигма структурного программирования; • инструменты отладки, правила тестирования приложений, документирование.

2. Компетенции обучающегося, формируемые в результате освоения дисциплины

Планируемые результаты освоения ОП ВО (компетенции)	Планируемые результаты обучения по дисциплине
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	Знает: Основные конструкции языка программирования высокого уровня, основные компоненты современной среды программирования Умеет: Проектировать программу, кодировать программу, осуществлять тестирование программы, а также отлаживать программу с использованием инструментов среды программирования Имеет практический опыт: Работы с современной средой программирования, проектирования и решения простых задач
ОПК-5 Способен устанавливать программное и аппаратное обеспечение для информационных и автоматизированных систем	Знает: Среда программирования для создания программ на языках высокого уровня Умеет: Устанавливать среду программирования, создавать и отлаживать программы в среде программирования Имеет практический опыт: Установки и использования среды программирования для решения профессиональных задач

ОПК-7 Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Знает: Основные структуры данных и алгоритмы их обработки Умеет: Разрабатывать алгоритмы и создавать программы на основе концепции структурного программирования Имеет практический опыт: Разработки алгоритмов и создания программ, а также использования встроенных структур данных языка программирования высокого уровня
--	--

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
Нет	1.О.14 Операционные системы, 1.О.12.03 Объектно-ориентированное программирование, 1.О.16 Информационные системы и технологии, 1.О.18 Пакеты прикладных программ, 1.О.15 Вычислительные системы, сети и телекоммуникации, 1.О.17 Математическая логика и теория алгоритмов, 1.О.13 Базы данных, 1.О.12.02 Программирование на языках высокого уровня, Учебная практика, научно-исследовательская работа (получение первичных навыков научно-исследовательской работы) (4 семестр), Учебная практика, ознакомительная практика (2 семестр)

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин:

Нет

4. Объём и виды учебной работы

Общая трудоёмкость дисциплины составляет 4 з.е., 144 ч., 74,5 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах
		Номер семестра
Общая трудоёмкость дисциплины	144	144
<i>Аудиторные занятия:</i>	64	64
Лекции (Л)	32	32
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	32	32

Лабораторные работы (ЛР)	0	0
Самостоятельная работа (СРС)	69,5	69,5
с применением дистанционных образовательных технологий	0	
Изучение теоретического материала	10,5	10,5
Подготовка к экзамену.	38	38
Решение задач на самостоятельную разработку	21	21
Консультации и промежуточная аттестация	10,5	10,5
Вид контроля (зачет, диф.зачет, экзамен)	-	экзамен

5. Содержание дисциплины

№ раздела	Наименование разделов дисциплины	Объем аудиторных занятий по видам в часах			
		Всего	Л	ПЗ	ЛР
1	Раздел 1. Основные понятия. Алгоритмы и данные. Концепция типов. Операции над данными. Правила записи и семантика выражений.	8	4	4	0
2	Раздел 2. Операторы языка C: введение. Операторы условия и выбора. Проектирование алгоритмов с ветвлением.	8	2	6	0
3	Раздел 3. Операторы цикла в C++. Проектирование циклических алгоритмов.	8	4	4	0
4	Раздел 4. Функции: синтаксические правила и проектирование алгоритмов.	8	4	4	0
5	Раздел 5. Массивы как структура хранения данных. Функции и алгоритмы работы с массивами. Указатели: теоретическое введение и практическое использование. Принцип модульности при разработке приложения.	12	6	6	0
6	Раздел 6. Работа с текстом. Объектный тип std::string.	4	2	2	0
7	Раздел 7. Работа с двумерными массивами. Обмен данными с файлами.	6	4	2	0
8	Раздел 8. Парадигма структурного программирования. Концепция типов данных: структура как тип. Модульный стиль проектирования приложений.	10	6	4	0

5.1. Лекции

№ лекции	№ раздела	Наименование или краткое содержание лекционного занятия	Кол-во часов
1	1	Лекция 1. Введение в C++. Терминология предметной области. Среды разработчика. Концепция типов данных. Краткое введение в алгоритмы	2
2	1	Лекция 2. Концепция типов данных. Операции над данными. Правила записи и семантика арифметических и логических выражений.	2
3	2	Лекция 3. Операторы в формальных языках. Классификация операторов C++. Условный оператор. Проектирование алгоритмов с ветвлением.	2
4	3	Лекция 4. Циклические алгоритмы: общие вопросы. Операторы цикла C++ как инструменты для проектирования циклических алгоритмов.	2
5	3	Лекция 5. Классификация циклических алгоритмов. Циклы со счётчиком, циклы по событию, итерационные циклы. Проектирование циклических алгоритмов.	2

6	4	Лекция 6. Функции: синтаксис и семантика описания и обращения. Формальные и фактические параметры. Принцип модульности кратко.	2
7	4	Лекция 7. Функции: принцип модульности. Перегрузка функций, два механизма передачи параметров, область действия и время жизни имен. Проектирование и алгоритмы.	2
8	5	Лекция 8. Массивы как структура хранения данных. Семантика массива. Алгоритмы работы с массивами.	2
9	5	Лекция 9. Указатели: теоретическое введение и практическое использование. Полезности: динамические массивы, массивы как параметры функций, функции, возвращающие указатель, параметры по адресу.	2
10	5	Лекция 10. Введение в модульный стиль программирования.	2
11	6	Лекция 11. Работа с текстовыми данными. Объект std::string. Описание объектного типа. Кратко: свойства и методы string.	2
12	7	Лекция 12. Хранение данных приложения в файлах. Работа с файлами данных: чтение данных из файлов, выгрузка в файл.	2
13	7	Лекция 13. Работа с двумерными массивами: описание, алгоритмы, функции работы с массивами, чтение - запись из/в файла.	2
14	8	Лекция 14. Парадигма структурного программирования. Проектирование приложений в модульном стиле.	2
15	8	Лекция 15. Структуры как тип данных.	2
16	8	Лекция 16. Проектирование приложений в модульном стиле.	2

5.2. Практические занятия, семинары

№ занятия	№ раздела	Наименование или краткое содержание практического занятия, семинара	Кол-во часов
1	1	Практическое занятие 1. Введение в алгоритмы. Дано определение алгоритма, описаны способы представления алгоритмов, приведены базовые алгоритмические конструкции: линейные и алгоритмы с ветвлением. Даны примеры. К выполнению два упражнения, в каждом по две задачи.	2
2	1	Практическое занятие 2. Алгоритмы. Дано определение циклического алгоритма, названы и описаны виды циклов, приведены примеры. К выполнению три упражнения, представляющие разные виды циклов.	2
3	2	Практическое занятие 3. Операции и выражения. Приведены краткие сведения об операциях C++, о синтаксических правилах записи выражений и их семантике. К выполнению три упражнения: арифметические выражения, условная операция и логические выражения.	2
4	2	Практическое занятие 4. Линейные алгоритмы. Содержание задания: ответить на вопросы по теме "операции и выражения", решить две задачи на линейный алгоритм.	2
5	2	Практическое занятие 5. Условный оператор. Описаны синтаксис и семантика условного оператора. Приведены примеры записи логических выражений. К выполнению два упражнения.	2
6	3	Практическое занятие 6. Операторы цикла и циклические алгоритмы. Описаны синтаксис и семантика операторов цикла for, while и do. Приведены примеры проектирования простых циклов. Даны три упражнения: как проектировать арифметический цикл, итерационный и управляемый событием	2
7	3	Практическое занятие 7. Проектирование циклических алгоритмов. К выполнению три упражнения для кодирования итерационного и сложного циклических алгоритмов.	2
8	4	Практическое занятие 8. Функции. Изучение принципов модульной	2

		технологии разработки программ и использование их в своем программировании. Изучение синтаксических правил описания функций и обращения к ним. Изучение механизмов обращения к функции, передачи данных в функцию и механизма возвращения данных. К выполнению три упражнения.	
9	4	Практическое задание 9. Функции. Изучение принципов модульной технологии разработки программ и использование их в своем программировании. Изучение механизмов обращения к функции, передачи данных в функцию и механизма возвращения данных. Исследуются перегруженные функции, синтаксис и механизмы передачи параметров по ссылке, семантика static переменных. К выполнению три упражнения.	2
10	5	Практическое задание 10. Работа с массивами. Изучение синтаксических правил описания массивов и механизмов их реализации. Разработка алгоритмов обработки массивов в функциях. Массивы как параметры функций. Параллельные массивы. Перегрузка функций. Предложено к выполнению четыре упражнения.	2
11	5	Практическое занятие 11. Практическое использование указателей. Знакомство с указателями, как типом данных. Использование указателей при передаче параметров в функцию, соотношение между указателями и массивами. Введение в динамические массивы. Разработка алгоритмов обработки массивов в функциях. К выполнению три упражнения	2
12	5	Практическое занятие 12. Принцип модульности в решении содержательных задач. Цель работы: построение модели задачи предметной области, разработка функций, реализующих алгоритмы работы с массивами, использование типовых алгоритмов в решении задач.	2
13	6	Практическое занятие 13. Работа с текстовыми данными. Объект std::string. Цель работы. Использование методов string для работы с текстом. Использование функций для реализации алгоритмов работы со строками. К выполнению три упражнения и задание для самостоятельной работы.	2
14	7	Практическое задание 13. Двумерные массивы. Обмен данными с файлами	2
15	8	Практическое занятие 15. Структуры как тип данных. К выполнению упражнение на использование структурного типа.	2
16	8	Практическое задание 14. Case: проектирование приложения в модульном стиле. Задание является итоговой интегрированной работой по дисциплине "Основы Программирования".	2

5.3. Лабораторные работы

Не предусмотрены

5.4. Самостоятельная работа студента

Выполнение СРС			
Подвид СРС	Список литературы (с указанием разделов, глав, страниц) / ссылка на ресурс	Семестр	Кол-во часов
Изучение теоретического материала	Электронный ЮУрГУ. Конспект лекций, по разделам. Основная и дополнительная литература, по разделам.	1	10,5
Подготовка к экзамену.	Электронный ЮУрГУ. Практические задания, по темам. https://edu.susu.ru/course/view.php?id=140830 Основная и дополнительная литература, по разделам.	1	38

Решение задач на самостоятельную разработку	Электронный ЮУрГУ. Практические задания, по темам. https://edu.susu.ru/course/view.php?id=140830 Основная и дополнительная литература, по разделам.	1	21
---	---	---	----

6. Текущий контроль успеваемости, промежуточная аттестация

Контроль качества освоения образовательной программы осуществляется в соответствии с Положением о балльно-рейтинговой системе оценивания результатов учебной деятельности обучающихся.

6.1. Контрольные мероприятия (КМ)

№ КМ	Се-местр	Вид контроля	Название контрольного мероприятия	Вес	Макс. балл	Порядок начисления баллов	Учитывается в ПА
1	1	Текущий контроль	Задание для самостоятельной работы: Проектирование циклических алгоритмов.	1	5	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором две задачи. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - алгоритмы эффективны и правильно работают - 4 балла, проведено тестирование – 1 балл. Максимальное количество баллов – 5. Весовой коэффициент мероприятия – 1.	экзамен
2	1	Текущий контроль	Задание для самостоятельной работы: Проектирование циклических алгоритмов	1	6	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором две задачи. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - алгоритм эффективный и правильно работает - 5 баллов, проведено тестирование – 1 балл. Максимальное количество	экзамен

						баллов – 6. Весовой коэффициент мероприятия – 1.	
3	1	Текущий контроль	Задание для самостоятельной работы: Функции: описание и обращение.	1	6	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором две задачи. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - алгоритмы эффективны и правильно работают - 2 балла, проведено тестирование – 1 балл. За две задачи максимальное количество баллов – 6. Весовой коэффициент мероприятия – 1.	экзамен
4	1	Текущий контроль	Задание для самостоятельной работы: Проектирование функций и основные алгоритмы.	1	6	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором две задачи. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - алгоритмы эффективны и правильно работают - 2 балла, проведено тестирование – 1 балл. За две задачи максимальное количество баллов – 6. Весовой коэффициент мероприятия – 1.	экзамен
5	1	Текущий контроль	Задание для самостоятельной разработки. Алгоритмы работы с массивами.	1	10	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором три задачи. Оценке подлежит правильность выполнения каждой. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: -	экзамен

						алгоритмы эффективны и правильно работают - 2 балла, проведено тестирование – 1 балл. Один балл добавляется за соблюдение общих принципов модульности. За три задачи максимальное количество баллов – 10. Весовой коэффициент мероприятия – 1.	
6	1	Текущий контроль	Задание для самостоятельной разработки. Массивы и указатели. Модульный стиль проектирования	1	5	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором две задачи. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - алгоритмы эффективны и правильно работают - по 2б. за задачу, проведено тестирование – 1 балл. За две задачи максимальное количество баллов – 5. Весовой коэффициент мероприятия – 1.	экзамен
7	1	Текущий контроль	Задание для самостоятельной работы: Решение содержательных задач на основе концепции структурного программирования	1	8	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код, в котором две задачи. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: качество построения инфологической модели - 1б., выбор способа представления данных -2б., функциональная декомпозиция - 2б, эффективность алгоритмов - 2б, тестирование - 1б. Максимальное количество баллов – 8. Весовой коэффициент мероприятия – 1..	экзамен
8	1	Текущий контроль	Задание для самостоятельной работы: Работа с текстом: объект std::string	1	4	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия	экзамен

						используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - алгоритм эффективный и правильно работает - 2 балла, проведено тестирование – 2 балла. Максимальное количество баллов – 4. Весовой коэффициент мероприятия – 1.	
9	1	Текущий контроль	Задание для самостоятельной работы: Структуры как тип данных. Обновление: чтение данных из файла	1	10	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код. Оценке подлежит правильность выполнения. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Общий балл при оценке складывается из следующих показателей: - структура данных адекватно отражает модель объекта - 3б., алгоритмы эффективны и правильно работают - 6 баллов, проведено тестирование – 1 балла Максимальное количество баллов – 10. Весовой коэффициент мероприятия – 1.	экзамен
10	1	Текущий контроль	Практическое задание Case: проектирование приложения в модульном стиле	1	20	Кейс: Задание является итоговой интегрированной работой по дисциплине "Основы программирования", в которой студент самостоятельно разрабатывает приложение на основе модульного стиля. При оценивании результатов мероприятия используется балльно-рейтинговая система оценивания результатов учебной деятельности обучающихся (утверждена приказом ректора от 24.05.2019 г. № 179). Критерии оценки: выполнена функциональная декомпозиция задачи, построена блочная модель приложения - 5б, для представления данных выбраны структуры данных, которые наилучшим образом отражают модель данных прикладной задачи - 5 б, для реализации отдельных блоков модели выбраны типовые алгоритмы, написан и отлажен код	экзамен

						функций - 5б, соблюдены принципы модульности и реализован сценарий, данные подгружаются из файлов – 5б. Максимальное количество баллов – 20. Весовой коэффициент мероприятия – 1.	
11	1	Промежуточная аттестация	Задание для проведения промежуточной аттестации	-	40	<p>Экзамен проводится в традиционной форме, наивысшая оценка 40 баллов. Задание для проведения промежуточной аттестации содержит теоретическую часть и практическое задание. Теоретическая часть выполняется письменно, включает 10 вопросов из разных разделов курса. На подготовку ответов студенту дается 60 минут. Правильный ответ на вопрос соответствует 2 баллам, неточный соответствует 1 баллу, неправильный ответ или его отсутствие соответствует 0 баллов. Практическая часть содержит задание на разработку программы, уровень сложности которой соответствует заданию номер 10 практического курса. Выполняется в среде разработчика, на выполнение задания отводится 60 минут. Наивысшая оценка 20 баллов. Критерии оценки: правильное построение функциональной модели, выбор структур данных, соответствующих модели, эффективность алгоритмов, соблюдение правил модульного стиля: по 5 баллов за каждый пункт. Совокупный балл определяется сложением. В случае возникновения разногласий проводится собеседование, при котором студенту могут быть заданы уточняющие вопросы. Экзамен завершен, если по совокупности баллов студент набрал не менее 60 % общего рейтинга обучающегося, в ином случае студент направляется на пересдачу.</p>	экзамен

6.2. Процедура проведения, критерии оценивания

Вид промежуточной аттестации	Процедура проведения	Критерии оценивания
экзамен	Экзамен проводится в традиционной форме. Задание для проведения промежуточной аттестации содержит	В соответствии с пп. 2.5, 2.6

	теоретическую часть и практическое задание. Теоретическая часть выполняется письменно, включает 10 вопросов из разных разделов курса. На подготовку ответов студенту дается 60 минут. Практическая часть содержит задание на разработку программы, уровень сложности соответствует заданию номер 10 практического курса. Выполняется в среде разработчика Visual Studio. На выполнение задания отводится 60 минут. По завершении работы подводятся итоги, выводится оценка, и в случае необходимости проводится собеседование, при котором студенту могут быть заданы уточняющие вопросы.	Положения
--	---	-----------

6.3. Оценочные материалы

Компетенции	Результаты обучения	№ КМ											
		1	2	3	4	5	6	7	8	9	10	11	
ОПК-2	Знает: Основные конструкции языка программирования высокого уровня, основные компоненты современной среды программирования	+	+	+	+	+				+	+	+	+
ОПК-2	Умеет: Проектировать программу, кодировать программу, осуществлять тестирование программы, а также отлаживать программу с использованием инструментов среды программирования	+	+	+	+	+	+	+	+	+	+	+	+
ОПК-2	Имеет практический опыт: Работы с современной средой программирования, проектирования и решения простых задач	+	+	+	+	+	+				+	+	+
ОПК-5	Знает: Среды программирования для создания программ на языках высокого уровня	+	+	+	+	+	+	+	+	+	+	+	+
ОПК-5	Умеет: Устанавливать среду программирования, создавать и отлаживать программы в среде программирования	+	+	+	+	+	+	+	+	+	+	+	+
ОПК-5	Имеет практический опыт: Установки и использования среды программирования для решения профессиональных задач	+	+	+	+	+	+	+	+	+	+	+	+
ОПК-7	Знает: Основные структуры данных и алгоритмы их обработки				+	+	+	+				+	+
ОПК-7	Умеет: Разрабатывать алгоритмы и создавать программы на основе концепции структурного программирования	+	+	+	+	+	+	+	+			+	+
ОПК-7	Имеет практический опыт: Разработки алгоритмов и создания программ, а также использования встроенных структур данных языка программирования высокого уровня	+	+	+	+	+	+	+	+			+	+

Фонды оценочных средств по каждому контрольному мероприятию находятся в приложениях.

7. Учебно-методическое и информационное обеспечение дисциплины

Печатная учебно-методическая документация

а) основная литература:

1. Конова, Е. А. Алгоритмы и программы. Язык С++ [Текст] учеб. пособие для вузов по направлению "Приклад. информатика" Е. А. Конова, Г. А. Поллак. - СПб. и др.: Лань, 2016. - 384 с. ил.
2. Подбельский, В. В. Язык Си++ [Текст] учеб. пособие для вузов по направлениям "Приклад. математика" и "Вычисл. машины, комплексы, системы и сети" В. В. Подбельский. - 5-е изд. - М.: Финансы и статистика, 2004. - 559 с.

б) дополнительная литература:

1. Павловская, Т. А. С/С++. Программирование на языке высокого уровня [Текст] учебник для вузов по направлению "Информатика и вычисл. техника" Т. А. Павловская. - СПб. и др.: Питер, 2020. - 460 с. ил.

в) отечественные и зарубежные журналы по дисциплине, имеющиеся в библиотеке:
Не предусмотрены

г) методические указания для студентов по освоению дисциплины:

1. Учебник для изучения основ С++
2. Решебник для выполнения практических работ

из них: учебно-методическое обеспечение самостоятельной работы студента:

1. Учебник для изучения основ С++
2. Решебник для выполнения практических работ

Электронная учебно-методическая документация

№	Вид литературы	Наименование ресурса в электронной форме	Библиографическое описание
1	Основная литература	Электронный каталог ЮУрГУ	Конова Е. А. Информатика и программирование [Текст] : учеб. пособие к прак. работам по направлению 230700.62 "Приклад. информатика" / Е. А. Конова, Г. А. Поллак ; Юж.-Урал. гос. ун-т, Каф. Информатика ; ЮУрГУ, Челябинск : Издательский Центр ЮУрГУ , 2014 http://www.lib.susu.ac.ru/ftd?base=SUSU_METHOD&key=000532629
2	Дополнительная литература	Электронный каталог ЮУрГУ	Конова Е. А. Практический курс программирования на языках С и С++ [Текст] : учеб. пособие / Е. А. Конова, Г. А. Поллак, А. М. Ткачев ; под ред. Е. А. Коновой ; Юж.-Урал. гос. ун-т, Каф. Информатика ; ЮУрГУ, Челябинск : Издательство ЮУрГУ , 2004 http://www.lib.susu.ac.ru/ftd?base=SUSU_METHOD&key=000305742
3	Методические пособия для самостоятельной работы студента	Учебно-методические материалы кафедры	Учебный курс в электронном ЮУрГУ: 2021/2022 Основы программирования (очная, 09.03.02, 09.03.03, Конова Е.А.) https://edu.susu.ru/course/view.php?id=140830

Перечень используемого программного обеспечения:

1. Microsoft-Windows(бессрочно)
2. Microsoft-Office(бессрочно)
3. ФГАОУ ВО "ЮУрГУ (НИУ)"-Портал "Электронный ЮУрГУ" (<https://edu.susu.ru>)(бессрочно)
4. Microsoft-Visual Studio(бессрочно)

Перечень используемых профессиональных баз данных и информационных справочных систем:

1. ООО "ГарантУралСервис"-Гарант(бессрочно)

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Лекции	203 (3г)	Проектор
Самостоятельная работа студента	447 (Л.к.)	Компьютерный класс 30 раб. станций. Установленное ПО: браузер, Visual Studio.
Контроль самостоятельной работы	447 (Л.к.)	Компьютерный класс 30 раб. станций. Установленное ПО: браузер, Visual Studio.
Практические занятия и семинары	447 (Л.к.)	Компьютерный класс 30 раб. станций. Установленное ПО: браузер, Visual Studio.
Экзамен	447 (Л.к.)	Компьютерный класс 30 раб. станций. Установленное ПО: браузер, Visual Studio.