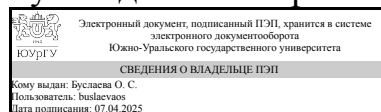


ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

УТВЕРЖДАЮ:
Руководитель направления



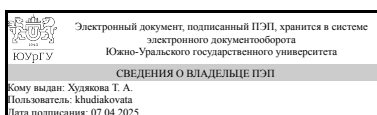
О. С. Буслаева

РАБОЧАЯ ПРОГРАММА

дисциплины 1.О.15.01 Основы программирования
для направления 09.03.02 Информационные системы и технологии
уровень Бакалавриат
форма обучения очная
кафедра-разработчик Цифровая экономика и информационные технологии

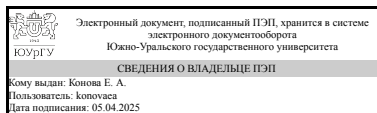
Рабочая программа составлена в соответствии с ФГОС ВО по направлению подготовки 09.03.02 Информационные системы и технологии, утверждённым приказом Минобрнауки от 19.09.2017 № 926

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



Т. А. Худякова

Разработчик программы,
старший преподаватель



Е. А. Конова

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-6 Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	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

3. Место дисциплины в структуре ОП ВО

Перечень предшествующих дисциплин, видов работ учебного плана	Перечень последующих дисциплин, видов работ
Нет	1.О.15.02 Программирование на языках высокого уровня, 1.О.14 Архитектура информационных систем, 1.О.12 Структуры данных и прикладные алгоритмы, 1.О.16 Базы данных, 1.О.11 Объектно-ориентированное программирование, 1.О.13 Информационные технологии, 1.О.17 Операционные системы, 1.О.22 Пакеты прикладных программ, 1.О.21 Математическая логика и теория алгоритмов, Учебная практика (ознакомительная) (2 семестр), Учебная практика (научно-исследовательская работа, получение первичных навыков научно-исследовательской работы) (4 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 трудоемкость дисциплины составляет 4 з.е., 144 ч., 74,5 ч. контактной работы

Вид учебной работы	Всего часов	Распределение по семестрам в часах
		Номер семестра
		1
Общая трудоёмкость дисциплины	144	144
<i>Аудиторные занятия:</i>	64	64
Лекции (Л)	16	16
Практические занятия, семинары и (или) другие виды аудиторных занятий (ПЗ)	48	48
Лабораторные работы (ЛР)	0	0
<i>Самостоятельная работа (СРС)</i>	69,5	69,5
Подготовка к экзамену.	38	38
Решение задач на самостоятельную разработку	21	21
Изучение теоретического материала	10,5	10,5
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.	6	2	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	2	6	0
4	Раздел 4. Функции: синтаксические правила и проектирование алгоритмов.	8	2	6	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	2	10	0
6	Раздел 6. Работа с текстом. Объектный тип std::string.	6	2	4	0
7	Раздел 7. Работа с двумерными массивами. Обмен данными с файлами.	6	2	4	0
8	Раздел 8. Парадигма структурного программирования. Концепция типов данных: структура как тип. Модульный стиль проектирования приложений.	10	2	8	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. Операции над данными. Правила записи и семантика арифметических и логических выражений. Краткое введение в алгоритмы.	2
2	2	Лекция 2. Операторы в формальных языках. Классификация операторов C++. Условный оператор. Проектирование алгоритмов с ветвлением.	2
3	3	Лекция 3. Циклические алгоритмы: общие вопросы. Операторы цикла C++ как инструменты для проектирования циклических алгоритмов. Классификация циклических алгоритмов. Циклы со счётчиком, циклы по событию, итерационные циклы. Проектирование циклических алгоритмов.	2
4	4	Лекция 4. Функции: синтаксис и семантика описания и обращения. Формальные и фактические параметры. Принцип модульности кратко. Перегрузка функций. Два механизма передачи параметров. Область действия и время жизни имен. Проектирование и алгоритмы.	2
5	5	Лекция 5. Массивы как структура хранения данных. Семантика массива. Алгоритмы работы с массивами. Указатели: теоретическое введение и практическое использование. Полезности: динамические массивы, массивы как параметры функций, функции, возвращающие указатель, параметры по адресу.	2
6	6	Лекция 6. Работа с текстовыми данными. Объект std::string. Описание	2

		объектного типа. Кратко: свойства и методы string.	
7	7	Лекция 7. Хранение данных приложения в файлах. Работа с файлами данных: чтение данных из файлов, выгрузка в файл.	2
8	8	Лекция 8. Парадигма структурного программирования. 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
7,8	3	Практические занятия 7 и 8. Проектирование циклических алгоритмов. К выполнению три упражнения для кодирования итерационного и сложного циклических алгоритмов.	4
9	4	Практическое занятие 9. Функции. Изучение принципов модульной 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.	2
10,11	4	Практические занятия 10 и 11. 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.	4
12,13	5	Практические занятия 12 и 13. 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.	4
14	5	Практическое занятие 14. Практическое использование указателей.	2

		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	
15,16	5	Практические занятия 15 и 16. 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.	4
17,18	6	Практические занятия 17 и 18. Работа с текстовыми данными. Объект <code>std::string</code> . Цель работы. Использование методов <code>string</code> для работы с текстом. Использование функций для реализации алгоритмов работы со строками. К выполнению три упражнения и задание для самостоятельной работы.	4
19,20	7	Практические занятия 19 и 20. Двумерные массивы. Обмен данными с файлами	4
21,22,23	8	Практические занятия 21, 22, 23. Case: проектирование приложения в модульном стиле. Задание является итоговой интегрированной работой по дисциплине "Основы Программирования".	6
24	8	Практическое занятие 24. Структуры как тип данных. К выполнению упражнение на использование структурного типа.	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
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
Изучение теоретического материала	Электронный ЮУрГУ. Конспект лекций, по разделам. Основная и дополнительная литература, по разделам.	1	10,5

6. Фонд оценочных средств для проведения текущего контроля успеваемости, промежуточной аттестации

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 баллов – 6. Весовой коэффициент мероприятия – 1.	экзамен
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	экзамен

						следующих показателей: - алгоритмы эффективны и правильно работают - 2 балла, проведено тестирование – 1 балл. За две задачи максимальное количество баллов – 6. Весовой коэффициент мероприятия – 1.	
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: - 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.	экзамен
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	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: - алгоритмы эффективны и правильно работают - по 2б. за задачу, проведено тестирование – 1 балл. За две задачи максимальное количество баллов – 5. Весовой коэффициент мероприятия – 1.	
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 <code>std::string</code>	1	4	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: - алгоритм эффективный и правильно работает - 2 балла, проведено тестирование – 2 балла. Максимальное количество баллов – 4. Весовой коэффициент мероприятия – 1.	экзамен
9	1	Текущий контроль	Задание для самостоятельной работы: Структуры	1	10	Защита работы осуществляется в индивидуальном порядке. Студент предоставляет отлаженный код.	экзамен

			как тип данных. Обновление: чтение данных из файла			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.	
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 функций - 5б, соблюдены принципы модульности и реализован сценарий, данные подгружаются из файлов – 5б. Максимальное количество баллов – 20. Весовой коэффициент мероприятия – 1.	экзамен
11	1	Текущий контроль	Контрольная работа	1	20	Контрольная работа нацелена на проверку сформированности компетенций по теоретической части дисциплины. Выполняется письменно, включает 10 вопросов из разных разделов курса. На подготовку ответов студенту дается 60 минут. Правильный ответ на вопрос соответствует 2 баллам,	экзамен

		форме	
1	Основная литература	Электронный каталог ЮУрГУ	Конова Е. А. Информатика и программирование [Текст] : учеб. пособие к практ. 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:

Нет

8. Материально-техническое обеспечение дисциплины

Вид занятий	№ ауд.	Основное оборудование, стенды, макеты, компьютерная техника, предустановленное программное обеспечение, используемое для различных видов занятий
Лекции	203 (Зг)	Проектор
Экзамен	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.
Самостоятельная работа студента	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.