



Федеральное государственное бюджетное образовательное учреждение высшего образования
«ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ ИМЕНИ ДВАЖДЫ ГЕРОЯ
СОВЕТСКОГО СОЮЗА, ЛЕТЧИКА-КОСМОНАВТА А.А. ЛЕОНОВА»

УТВЕРЖДАЮ

И. о. проректора

А.В. Троицкий

«__» _____ 2023 г.

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

РАБОЧАЯ ПРОГРАММА

ДИСЦИПЛИНЫ

«Основы алгоритмизации и программирования»

Направление подготовки: 09.03.02 Информационные системы и технологии

Профиль: высокопроизводительные вычислительные и
телекоммуникационные интеллектуальные системы и комплексы

Уровень высшего образования: бакалавриат

Форма обучения: очная, заочная

Королев
2023

Рабочая программа является составной частью основной профессиональной образовательной программы и проходит рецензирование со стороны работодателей в составе основной профессиональной образовательной программы. Рабочая программа актуализируется и корректируется ежегодно.

Автор: Парафейников И.В. Рабочая программа дисциплины: Основы алгоритмизации и программирования – Королев МО: «Технологический университет», 2023 г.

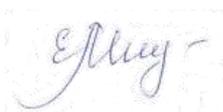
Рецензент: к.т.н., доцент Баранова О.М.

Рабочая программа составлена в соответствии с требованиями федерального государственного образовательного стандарта высшего образования (ФГОС ВО) по направлению подготовки бакалавров 09.03.02 Информационные системы и технологии и Учебного плана, утвержденного Ученым советом Университета протокол № 9 от 11.04.2023 г.

Рабочая программа рассмотрена и одобрена на заседании кафедры:

Заведующий кафедрой (ФИО, ученая степень, звание, подпись)	Баранова О.М., к.т.н., доцент			
Год утверждения (переутверждения)	2023			
Номер и дата протокола заседания кафедры	№ 7 от 03.04.2023			

Рабочая программа согласована:

Руководитель ОПОП  к.т.н., доцент Макарова Е.Г.

Рабочая программа рекомендована на заседании УМС:

Год утверждения (переутверждения)	2023			
Номер и дата протокола заседания УМС	№ 5 от 11.04.2023 г.			

1. Перечень планируемых результатов обучения по дисциплине, соотнесенных с планируемыми результатами освоения ОПОП

Целью изучения дисциплины является приобретение теоретических и практических навыков в области технологии алгоритмизации и программирования на языках высокого уровня для решения прикладных задач в профессиональной деятельности.

Содержание дисциплины охватывает круг вопросов, связанных с формированием у бакалавров представления о современном подходе к решению различных задач прикладной области с помощью ЭВМ; подготовка бакалавров к использованию современных компьютерных технологий и алгоритмов в будущей производственно-технической, проектно-конструкторской и исследовательской деятельности.

В процессе обучения студент приобретает и совершенствует следующие компетенции:

общефессиональные компетенции:

- Способен понимать принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, и использовать их при решении задач профессиональной деятельности (ОПК-2);
- Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий (ОПК-6);

профессиональные компетенции (ПК):

- Способность выполнять логическую и функциональную работу по созданию комплекса программ (ПК-14).

Основными **задачами** дисциплины является:

1. формирование знаний по алгоритмизации и программированию, о стиле написания программ, о рациональных методах их разработки и оптимизации, о стратегии отладки и тестирования программ;
2. изучение структур данных в памяти и в файлах и алгоритмов работы с ними;
3. знакомство с основными принципами организации хранения и поиска данных, алгоритмами сортировки и поиска;
4. изучение основ технологии программирования и методов решения вычислительных задач и задач обработки символьных данных;
5. приобретение навыков использования базового набора фрагментов и алгоритмов в процессе разработки программ, навыков анализа и «чтения» программ.

После завершения освоения данной дисциплины студент должен:

Необходимые знания:

- исследует принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности;

- исследует методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий;
- знает логические и функциональные уровни организации информационных систем.

Необходимые умения:

- умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности;
- умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий;
- умеет выполнять логическую и функциональную работу по созданию комплекса программ.

Трудовые действия:

- имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности;
- имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач;
- владеет способами создания программного и информационного обеспечения.

2. Место дисциплины в структуре ОПОП ВО

Дисциплина «Основы алгоритмизации и программирования» относится к обязательной части основной образовательной программы подготовки бакалавров по направлению подготовки 09.03.02 Информационные системы и технологии.

Дисциплина реализуется кафедрой прикладного программного обеспечения.

Изучение данной дисциплины базируется на ранее изученных дисциплинах: «Информатика» и компетенциях: ОПК-2, ОПК-6.

Основные положения дисциплины должны быть использованы в дальнейшем при изучении дисциплины «Информационные технологии проектирования».

3. Объем дисциплины и виды учебной работы

Общая трудоемкость освоения дисциплины составляет 6 зачетных единиц, 216 часа. Преподавание дисциплины ведется на 1-ом и 2-ом курсе, во 2-ом и 3-ем семестрах соответственно.

Таблица 1

Виды занятий	Всего часов	Семестр 2	Семестр 3
Общая трудоемкость	216	108	108
ОЧНАЯ ФОРМА ОБУЧЕНИЯ			
Аудиторные занятия	96	48	48
Лекции (Л)	32	16	16
Практические занятия (ПЗ)	64	32	32
Лабораторные работы (ЛР)	–	–	–
Самостоятельная работа	120	60	60
Курсовые работы (проекты)	+	+	-
Расчетно-графические работы	–	–	-
Контрольная работа, домашнее задание	+ –	+ –	+ -
Текущий контроль знаний (7 - 8, 14 - 15 недели)	Тест	+	+
Вид итогового контроля	экзамен	Экзамен	Экзамен

ЗАОЧНАЯ ФОРМА ОБУЧЕНИЯ

Виды занятий	Всего часов	Курс 1	Курс 2
Общая трудоемкость	216	108	108
Аудиторные занятия	24	12	12
Лекции (Л)	8	4	4
Практические занятия (ПЗ)	16	8	8
Лабораторные работы (ЛР)	–	–	–
Самостоятельная работа	192	96	96
Курсовые работы (проекты)	-	-	-
Расчетно-графические работы	–	–	-
Контрольная работа, домашнее задание	+ –	+ –	+ -
Текущий контроль знаний (7 - 8, 14 - 15 недели)	Тест	+	+
Вид итогового контроля	Экзамен	Экзамен	Экзамен

4. Содержание дисциплины

4.1. Темы дисциплины и виды занятий

Таблица 2

Наименование тем	Лекции, очн/заочн час	Практические занятия, очн/заочн час	Занятия в интерактивной форме, очн/заочн час	Практическая подготовка, очн/заочн час.	Код компетенций
Тема 1. Основы алгоритмизации	2/0,5	4/1	2/1	2/1	ОПК-2
Тема 2. Основы языков программирования	2/0,5	4/1	2/1	2/1	ОПК-2
Тема 3. Выражения и операторы. Структурное программирование	4/1	8/2	2/2	2/2	ОПК-6
Тема 4. Массивы	4/1	8/2	2/2	2/2	ОПК-6
Тема 5. Циклические конструкции	4/1	8/2	2/2	2/2	ОПК-6
Тема 6. Строки	4/1	8/2	2/2	2/2	ОПК-6
Тема 7. Функции	4/1	8/2	4/2	4/2	ОПК-6
Тема 8. Алгоритмы сортировки	4/1	8/2	4/2	4/2	ПК-14
Тема 9. Структуры данных	4/1	8/2	4/2	4/2	ПК-14
Итого:	32/8	64/16	24/16	24/16	

4.2. Содержание тем дисциплины

Тема 1. Основы алгоритмизации

Алгоритм решения задачи. Способы записи алгоритмов. Эквивалентность способов записи. Перевод алгоритма в программу (кодирование). Языки программирования. Трансляция и интерпретация. Технологии программирования.

Тема 2. Основы языков программирования

Алфавит и синтаксис языка программирования. Типы данных, идентификаторы, переменные, константы, выражения. Приведение типов и преобразование типов в выражениях. Операторы. Служебные слова. Подпрограммы и функции. Локальные и глобальные переменные. Формальные параметры. Структура программы. Понятие о стиле программирования.

Тема 3. Выражения и операторы. Структурное программирование

Арифметические и логические операции, операции сравнения, битовые операции. Приоритет операций. Оператор присваивания. Управление ходом выполнения программы: виды условных операторов, оператор выбора, циклы. Алгоритмы с условиями. Методы отладки программ.

Тема 4. Массивы

Массивы. Характерные ошибки при работе с массивами и методы защиты от них. Передача массива в функцию.

Тема 5. Циклические конструкции

Повторяющиеся действия в алгоритмах. Разбор циклических алгоритмов. Виды операторов цикла. Работа с переменными в цикле. Ошибки при работе с циклами. Операторы break, continue. Файловый ввод и вывод.

Тема 6. Строки

Основные алгоритмы работы со строками. Различные способы организации строковых данных. Работа с отдельными символами. Наиболее употребительные функции для работы со строками.

Тема 7. Функции

Функции в программе, их назначение и использование. Объявление и определение функции. Вызов функции. Область видимости функции. Обмен данными с функциями. Механизмы передачи значений в функцию. Возврат значений функциями. Возврат управления из функции. Рекурсия. Перегрузка функций и использование аргументов по умолчанию.

Тема 8. Алгоритмы сортировки

Основные алгоритмы сортировки массива. Понятие эффективности алгоритма сортировки.

Тема 9. Структуры данных

Структуры данных. Доступ к членам структуры. Ссылки на структуры. Стеки, списки, очереди, множества.

5. Перечень учебно-методического обеспечения для самостоятельной работы по дисциплине

«Методические указания для обучающихся по освоению дисциплины»

6. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Структура фонда оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине приведена в Приложении 1 к настоящей Рабочей программе.

7. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Белов, В. В. Алгоритмы и структуры данных : учебник / В. В. Белов, В. И. Чистякова. - Москва : КУРС : ИНФРА-М, 2020. - 240 с. - (Бакалавриат). - ISBN 978-5-906818-25-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1057212> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
2. Пруцков, А. В. Математическая логика и теория алгоритмов: Учебник / Пруцков А.В., Волкова Л.Л. - Москва :КУРС, НИЦ ИНФРА-М, 2018. - 152 с.: - (Бакалавриат). - ISBN 978-5-906818-74-4. - Текст : электронный. - URL: <https://znanium.com/catalog/product/956763> (дата обращения: 23.10.2020). – Режим доступа: по подписке.

Дополнительная литература:

1. Канцедал, С. А. Алгоритмизация и программирование : учебное пособие / С.А. Канцедал. — Москва : ФОРУМ : ИНФРА-М, 2021. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189320> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
2. Игнашева, Е. П. Системы счисления, алгоритмизация и программирование : учебное пособие / Е.П. Игнашева. — Москва : ИНФРА-М, 2020. — 224 с. — (Военное образование). - ISBN 978-5-16-015295-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1078360> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
3. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-553-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1042452> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
4. Колдаев В.Д.; Под ред. Гагариной Л.Г. Численные методы и программирование: Учебное пособие - М.: ИД ФОРУМ, НИЦ ИНФРА-М, 2016. - 336 с. ISBN 978-5-8199-0333-9 / ЭБС «Знаниум». <http://znanium.com/bookread2.php?book=546692>

Электронные книги:

1. Бабенко, М. А. Введение в теорию алгоритмов и структур данных: Краткий учебный курс / Бабенко М.А., Левин М.В., - 3-е изд. - Москва :МЦНМО, 2016. - 144 с.: ISBN 978-5-4439-2396-3. - Текст :

- электронный. - URL: <https://znanium.com/catalog/product/958593> (дата обращения: 23.10.2020). – Режим доступа: по подписке
2. Воронцова Е.А. Программирование на С++ с погружением: практические задания и примеры кода - М.: НИЦ ИНФРА-М, 2016. - 80 с. / ЭБС «Знаниум». <http://znanium.com/bookread2.php?book=563294>
 3. Корнеев В.И., Гагарина Л.Г., Корнеева М.В. Программирование графики на С++. Теория и примеры : учеб. пособие — М.: ИД «ФОРУМ»: ИНФРА-М, 2017. — 517 с. / ЭБС «Знаниум» <http://znanium.com/bookread2.php?book=562914>

8. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

Интернет-ресурсы:

<http://www.intuit.ru/studies/courses/555/411/info> – Алгоритмы и теория вычислений.

<http://www.intuit.ru/studies/courses/1106/146/info> – Основы теории вычислимых функций

<http://www.infojournal.ru> – Научно-образовательный портал,

<http://www.interface.ru/> – Научно-образовательный портал.

<http://www.codeblocks.org/>

<https://code.visualstudio.com/>

<http://www.cplusplus.com/> - Сеть ресурсов, посвященных языку С++

<http://pythontutor.ru/> - Курс по программированию на языке Python

<https://www.python.org/> – Сайт разработчиков Python

<https://jupyter.org/>

<https://colab.research.google.com/>

9. Методические указания для обучающихся по освоению дисциплины

Методические указания для обучающихся по освоению дисциплины приведены в Приложении 2 к настоящему Положению.

10. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине

Перечень программного обеспечения:

LibreOffice, VS Code, GCC + MinGW, Python 3, PyCharm

Информационные справочные системы:

1. Электронные ресурсы образовательной среды Технологического университета.

2. Рабочая программа и методическое обеспечение по дисциплине «Основы алгоритмизации и программирования»

11. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

Лекционные занятия:

- аудитория, оснащенная презентационной техникой (проектор, экран);
- комплект электронных презентаций/слайдов.

Практические занятия:

- учебный класс, оснащенный вычислительной техникой (ПК) и доступом к Интернет-ресурсам.
- рабочее место преподавателя, оснащенное компьютером с доступом в Интернет;
- рабочие места студентов, оснащенные компьютерами с доступом в Интернет.

Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ДЛЯ ПРОВЕДЕНИЯ
ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ОБУЧАЮЩИХСЯ ПО
ДИСЦИПЛИНЕ**

«Основы алгоритмизации и программирования»

(Приложение 1 к рабочей программе)

Направление подготовки: 09.03.02 Информационные системы и технологии

**Профиль: высокопроизводительные вычислительные и
телекоммуникационные интеллектуальные системы и комплексы**

Уровень высшего образования: бакалавриат

Форма обучения: очная, заочная

1. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы

№ п/п	Индекс компетенции	Содержание компетенции (или ее части)*	Раздел дисциплины, обеспечивающий формирование компетенции (или ее части)	В результате изучения раздела дисциплины, обеспечивающего формирование компетенции (или ее части), обучающийся должен:		
				Необходимые знания	Необходимые умения	Трудовые действия
1	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>Тема 1. Основы алгоритмизации</p> <p>Тема 2. Основы языков программирования</p> <p>Тема 3. Выражения и операторы. Структурное программирование</p>	исследует принципы работы современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности	умеет выбирать современные информационные технологии и программные средства, в том числе отечественного производства, при решении задач профессиональной деятельности	имеет навыки применения современных информационных технологий и программных средств, в том числе отечественного производства, при решении задач профессиональной деятельности
2	ОПК-6	Способен разрабатывать алгоритмы и программы, пригодные для практического применения в области информационных систем и технологий	<p>Тема 4. Массивы</p> <p>Тема 5. Циклические конструкции</p> <p>Тема 6. Строки</p> <p>Тема 7. Функции</p> <p>Тема 8. Алгоритмы сортировки</p> <p>Тема 9. Структуры данных</p>	исследует методы алгоритмизации, языки и технологии программирования, пригодные для практического применения в области информационных систем и технологий	умеет применять методы алгоритмизации, языки и технологии программирования при решении профессиональных задач в области информационных систем и технологий	имеет навыки программирования, отладки и тестирования прототипов программно-технических комплексов задач
3.	ПК-14	Способность выполнять логическую и функциональную работу по созданию комплекса программ.	<p>Тема 8. Алгоритмы сортировки</p> <p>Тема 9. Структуры данных</p>	логические и функциональные уровни организации информационных систем	умеет выполнять логическую и функциональную работу по созданию комплекса программ	владеет способами создания программного и информационного обеспечения

2. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

<i>Код компетенции</i>	<i>Инструмент, оценивающий сформированность компетенции</i>	<i>Показатель оценивания компетенции</i>	<i>Критерии оценки</i>
ОПК-2	Письменное задание	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (компетенция <u>не сформирована</u>) – 2 и менее баллов</p>	<p>1. Проводится в форме письменной работы</p> <p>2.Время, отведенное на процедуру – 10 - 15 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <p>1.Соответствие ответа заявленной тематике (0-5 баллов).</p> <p>Максимальная сумма баллов - 5 баллов.</p>
ОПК-6	Практическое задание	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> уровне – 3 балла; <p>В) не сформирована (компетенция <u>не сформирована</u>) – 2 и менее баллов</p>	<p>1. Проводится в форме практического задания</p> <p>2.Время, отведенное на процедуру – 20-30 мин.</p> <p>Неявка – 0.</p> <p>Критерии оценки:</p> <p>1.Соответствие ответа заявленной тематике (0-5 баллов).</p> <p>Максимальная сумма баллов - 5 баллов.</p> <p>.</p>
ПК-14	Контрольная работа	<p>А) полностью сформирована (компетенция освоена на <u>высоком</u> уровне) – 5 баллов</p> <p>Б) частично сформирована:</p> <ul style="list-style-type: none"> • компетенция освоена на <u>продвинутом</u> уровне – 4 балла; • компетенция освоена на <u>базовом</u> 	<p>1. Проводится в форме письменной работы</p> <p>Неявка на защиту контрольной работы – 0.</p> <p>Критерии оценки:</p> <p>1.Соответствие содержания контрольной работы заявленной тематике (1 балл).</p>

		<p>уровне – 3 балла; В) не сформирована (компетенция не сформирована) – 2 и менее баллов</p>	<p>2. Качество источников и их количество при подготовке работы (1 балл). 3. Владение информацией и способность отвечать на вопросы аудитории (1 балл). 4. Качество самой представленной работы (1 балл). 5. Использование специализированного программного обеспечения (0,5 балл). 6. Оригинальность подхода и всестороннее раскрытие выбранной тематики (0,5 балл). Максимальная сумма баллов - 5 баллов.</p>
--	--	---	--

3. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Примерная тематика письменных заданий:

1. Составить алгоритм нахождения суммы элементов прямоугольной таблицы размером $[n:m]$
2. Составить алгоритм нахождения максимального элемента прямоугольной таблицы размером $[n:m]$.
3. Составить алгоритм нахождения максимального элемента таблицы и количество элементов.
4. Составить алгоритм на определение количества слов в предложении.
5. Составить алгоритм нахождения количества различных чисел в одномерной таблице.
6. Составить алгоритм нахождения суммы четных чисел от 2 до 200.
7. Составить алгоритм нахождения суммы цифр числа.
8. Составить алгоритм нахождения двузначного числа, сумма кубов цифр которого равна n .
9. Получить из слова a , вычеркивание некоторого количества букв, слово b .

10. Составить алгоритм записи положительных элементов таблицы А в таблицу В, а отрицательных элементов таблицы А в таблицу С.
11. Составить алгоритм решения системы уравнений $ax+by+c=0$ и $a_1x+b_1y+c_1=0$.
12. Составить алгоритм определения площади и периметра треугольника.
13. Составить алгоритм вычисления значения функции по заданному значению переменных:

$$e^x - \frac{y^2 + 12xy - 3x^2}{18y - 1} 2ctg(3x) - \frac{\ln \cos x}{\ln(1 + x^2)}$$

14. Записать арифметическое выражение на алгоритмическом языке и указать порядок выполнения операций:

$$D = \frac{\sqrt{x + b - a} + \ln y}{arctg(b + a)}$$

15. Составить алгоритм, реализующую решение задачи: дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
16. Составить схему алгоритма вычисления таблицы значений функции $f(x)$ на промежутке от -2 до 3 с шагом 0,5:

$$f(x) = \sum_{k=1}^5 \frac{(-2)^{k+1} (k+1)!}{(x + 2.5)^{k+1}}$$

Примерная тематика практических заданий:

1. Составить программу вычисления произведения нечетных чисел от 1 до 100.
2. Составить программу, определяющую количество четных и нечетных элементов одномерного массива.
3. Составить программу нахождения максимального элемента одномерного массива.
4. Составить программу нахождения номера минимального элемента одномерного массива.
5. Составить программу упорядочивания элементов одномерного массива по возрастанию.
6. Составить программу нахождения суммы элементов на главной диагонали матрицы.
7. Составить программу нахождения суммы двух двумерных массивов.
8. Составить программу вычисления числа сочетаний из n по m с использованием подпрограммы-функции.
9. Составить программу решения квадратного уравнения с использованием процедуры.
10. Составить блок-схему и написать программу, которая при вводе числа в диапазоне от 1 до 99 добавляет к нему слово "копейка" в правильной форме. Например, 1 копейка, 5 копеек, 42 копейки.
11. Составить блок-схему и написать программу с использованием оператора выбора варианта для решения задачи: дано целое число k ($1 < k < 365$).

Определить, каким днем недели (понедельник, вторник, воскресенье) является k-день невысокосного года, если 1 января – понедельник.

12. Составить блок-схему и написать программу для решения задачи: дана действительная матрица размером $n \times m$. Требуется преобразовать матрицу: поэлементно вычесть последнюю строку из всех строк, кроме последней.
13. Составить структурную схему алгоритма и проект программы для решения задачи: (исходные данные ввести с клавиатуры, результат вывести на экран) четырехугольник задан координатами своих вершин. Найти его периметр. Вершины ввести с клавиатуры

Примерная тематика заданий на контрольную работу:

Составить программу решения задачи с использованием файла данных и выполнить ее на ПК.

Вариант	Условие задачи
1	Создать файл, содержащий массив целых чисел. Найти наименьшее из модулей отклонения чисел от их среднего значения.
2	Создать файл, содержащий массив вещественных чисел. Определить количество чисел, меньших среднего арифметического значения всех чисел массива.
3	Создать файл, содержащий фамилии студентов и их возраст. Вывести фамилии студентов, имеющих наименьший возраст.
4	Создать файл, содержащий названия всех месяцев года. Вывести сначала летние месяцы, а затем - зимние.
5	Создать файл, содержащий фамилии студентов. Вывести список студентов, имеющих фамилии наибольшей длины.
6	Создать файл, содержащий произвольные текстовые строки. Подсчитать количество строк, начинающихся с буквы 'А'.
7	Создать файл, содержащий текстовые строки произвольной длины. Дополнить все строки символом 'X' до самой длинной строки.
8	Создать текстовый файл с произвольным числом строк. Подсчитать общее количество символов в каждой строке.
9	Переписать в файл выходных данных таблицу значений функций: $y = \sin x$ и $z = \operatorname{tg} x$, рассчитанных для всех x на отрезке $[0, 3]$ с шагом 0.1. Значения x записать в файле с одной цифрой в дробной части, значения y - с двумя цифрами в дробной части, значения z - в форме числа с порядком.
10	Создать файл, содержащий текстовые строки произвольной длины. Вывести строки с наименьшей длиной.
11	Создать файл, содержащий ФИО студентов и их экзаменационные оценки по трем дисциплинам. Вывести ФИО студентов, успешно сдавших все экзамены.
12	Создать файл, содержащий ФИО студентов и их экзаменационные оценки по трем дисциплинам. Выбрать студентов, имеющих хотя бы одну задолженности. Вывести ФИО этих студентов и количество несданных экзаменов.

13	Создать файл F1, содержащий ФИО студентов. Переписать его в файл F2 таким образом, чтобы перед ФИО студента стоял его порядковый номер.
14	Создать файл, состоящий из девяти строк. В первой строке записать одну цифру '1', во второй строке 1 две цифры '2', ..., в девятой строке - девять цифр '9'.
15	Создать файл, содержащий ФИО рабочих и их среднемесячный заработок. Вывести ФИО рабочих, имеющих наибольший заработок.
16	Создать файл, содержащий ФИО студентов и их экзаменационные оценки по пяти дисциплинам. Вывести ФИО студентов, сдавших сессию на «хорошо» и «отлично».
17	Создать файл, содержащий ФИО абонентов и их номера телефонов. Составить программу, которая по ФИО абонента выводит его номер телефона.
18	Создать файл, содержащий номер рейса самолета и количество свободных мест. Составить программу, которая по вводимому номеру рейса выводит сведения о количестве свободных мест на этот рейс.
19	Создать файл, содержащий названия детских игрушек и их стоимость. Вывести наименование и стоимость самой дорогой игрушки.
20	Создать файл, содержащий массив целых чисел произвольного размера. Определить количество элементов массива. Переменной L присвоить значение TRUE, если в массиве нечетное число элементов, и значение FALSE - в противном случае.

4. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

<i>Неделя текущей/промежуточно-го контроля</i>	<i>Вид оценочного средства</i>	<i>Код компетенций, оценивающих знания, умения, навыки</i>	<i>Содержание оценочного средства</i>	<i>Требования к выполнению</i>	<i>Срок сдачи (неделя семестра)</i>	<i>Критерии оценки по содержанию и качеству с указанием баллов</i>

Согласно графику учебного процесса	Тестирование	ОПК-2, ОПК-6 ПК-14	25 вопросов	Компьютерное тестирование. Время отведенное на процедуру – 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка – 0 Удовлетворительно – от 51% правильных ответов. Хорошо – от 70% правильных ответов. Отлично – от 90% правильных ответов.
Согласно графику учебного процесса	Тестирование	ОПК-2, ОПК-6 ПК-14	25 вопросов	Компьютерное тестирование. Время отведенное на процедуру – 30 минут	Результаты тестирования предоставляются в день проведения процедуры	Критерии оценки определяются процентным соотношением. Не явка -0 Удовлетворительно - от 51% правильных ответов. Хорошо - от 70%. Отлично – от 90%.
Согласно графику учебного процесса	Экзамен	ОПК-2, ОПК-6 ПК-14	2 вопроса, 1 практическое задание	Экзамен проводится в 2 этапа: устной форме, путем ответа на вопросы и демонстрация результатов выполнения практического задания. Время отведенное на процедуру – 30 минут.	Результаты предоставляются в день проведения экзамена	Критерии оценки: «Отлично»: <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответ на вопросы билета. «Хорошо»: <ul style="list-style-type: none"> • знание основных понятий предмета; • умение использовать и применять полученные знания на практике; • работа на практических занятиях; • знание основных научных теорий, изучаемых предметов; • ответы на вопросы билета неполные «Удовлетворительно»: <ul style="list-style-type: none"> • демонстрирует частичные знания по темам дисциплин; • незнание неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; «Неудовлетворительно»: <ul style="list-style-type: none"> • демонстрирует частичные знания по темам

						дисциплин; • незнание основных понятий предмета; • неумение использовать и применять полученные знания на практике; • не работал на практических занятиях; не отвечает на вопросы.
--	--	--	--	--	--	--

Примерная тематика тестовых заданий:

Процессор выполняет команды алгоритма, записанные ...

1. На алгоритмическом языке
2. На командном языке
3. В виде блок-схемы
4. На естественном языке
5. На машинном языке (в двоичном коде)

С фамилией какого из древних ученых связано происхождение слова «алгоритм»?

1. Евклид
2. Аль-Хорезми
3. Аль-Хайсама
4. Аль-Коши
5. Пифагор

Автором самого древнего алгоритма считается...

1. Евклид
2. Пифагор
3. Декарт
4. Аль-Хорезми
5. Аль-Коши

Какой из документов является алгоритмом?

1. Классный журнал
2. Расписание звонков
3. Правила техники безопасности
4. Инструкция по получению денег в банкомате
5. Список класса

Для представления алгоритма в графическом виде используют...

1. Геометрические фигуры
2. Линии, точки
3. Формулы
4. Все ответы правильны
5. Графики функции

Свойство алгоритма «дискретность» означает:

1. Предлагаемые действия должны быть понятными и единственно возможными

2. Способность алгоритма давать правильные результаты решения задач
3. Пригодность алгоритма для решения однотипных задач
4. Решение задач должно быть получено за определенное число шагов
5. Непрерывность алгоритмического процесса

Алгоритм — это...

1. последовательность команд для ЭВМ
2. совокупность требований к программе
3. набор определений и правил для исполнителя
4. отдельные указания исполнителю выполнить некоторые законченные действия
5. совокупность понятных и точных указаний о том, какие действия и в какой последовательности выполнять для решения любой задачи из заданного класса за конечное число шагов

Последовательность нескольких команд алгоритма, выполняемых одна за другой, называется...

1. Командой
2. Программой
3. Служебные слова
4. Серия
5. Система программ

Аргументами называются величины...

1. не являющиеся исходными данными для алгоритма
2. являющиеся результатами для алгоритма
3. используемые для обозначения
4. являющиеся исходными данными для алгоритма
5. являющиеся заголовком для алгоритма

Алгоритмы, целиком используемые в составе других алгоритмов, называются...

1. Линейными
2. Определенными
3. Разветвляющимися
4. Вспомогательными
5. Циклическими

В каком году разработан алгоритм Брезенхейма

1. 1983
2. 1957
3. 1978
4. 1965

Какой алгоритм заливки произвольной области с отверстиями более сложен

1. Рекурсивный
2. Рекурсивный на основе серий пикселей
3. Алгоритм с предварительным выделением границы
4. Примерно равны

Какой алгоритм заливки произвольной области с отверстиями выигрывает в скорости при многократной заливке

1. Рекурсивный
2. Рекурсивный на основе серий пикселей
3. Алгоритм с предварительным выделением границы
4. Примерно равны

Какая из структур данных используется для оптимизации скорости выполнения расщепления полигона на два по хорде

1. Двусвязный список
2. Двусвязный циклический список
3. Циклический список
4. Динамический массив

Какая структура данных используется для оптимизации скорости выполнения алгоритмов обработки полигонов

1. Двусвязный список
2. Двусвязный циклический список
3. Циклический список
4. Динамический массив

Какой из методов имеет аппаратную реализацию

1. Метод трассировки лучей
2. Метод z-буфера
3. Метод двоичного разбиения пространства
4. Метод плавающего горизонта

Какой из алгоритмов позволяет отображать графики функций от двух переменных

1. Метод трассировки лучей
2. Метод z-буфера
3. Метод двоичного разбиения пространства
4. Метод плавающего горизонта

Циклический алгоритм используется при вычислении ...

1. суммы всех чисел от 1 до 100
2. площади трапеции
3. корня квадратного уравнения
4. суммы двух чисел, введенных с клавиатуры
5. среднего арифметического всех двухзначных чисел

Разветвляющийся алгоритм – это ...

1. присутствие в алгоритме хотя бы одного условия
2. набор команд, которые выполняются последовательно друг за другом
3. многократное выполнение одних и тех же действий
4. алгоритм, использующий подпрограммы

Свойство алгоритма «дискретность» обозначает ...

1. что команды должны следовать друг за другом
2. что каждая команда должна быть описана в расчёте на конкретного исполнителя
3. разбиение алгоритма на конечное число простых шагов

4. обязательное наличие завершающих инструкций
5. последовательность выполнения команд алгоритма

Свойство алгоритма, указывающее, что каждое правило алгоритма должно быть чётким,

1. однозначным и не оставлять места для произвола, называется ...
2. дискретность
3. понятность
4. определённости
5. результативность
6. массовость

Понятность, массовость – это свойства ...

1. алгоритма
2. информации
3. кодирования
4. кибернетики
5. программы

Языками высокого уровня являются ...

1. машинный язык
2. язык ассемблера
3. процедурный язык
4. объектно-ориентированный язык
5. логический язык

Ошибки, которые обнаруживаются транслятором: ...

1. неверное образование имён переменных
2. пропуск знака пунктуации
3. неверное написание служебных слов
4. несогласованность скобок
5. неверное определение порядка арифметических действий
6. неполный учёт возможных условий
7. неправильно составленный алгоритм решения задачи

Верными утверждениями являются: ...

1. различают два вида трансляторов: компиляторы и интерпретаторы
2. различают два вида компиляторов: трансляторы и интерпретаторы
3. различают два вида интерпретаторов: компиляторы и трансляторы
4. транслятор – программа, переводящая текст программы на языке высокого
5. уровня в эквивалентную программу на машинном языке
6. после того, как программа откомпилирована, ни сама исходная программа, ни компилятор более не нужны

Типовые вопросы, выносимые на экзамен

- 1) Дать определения следующим понятиям, привести примеры: а) алгоритм; б) алгоритмический язык; в) язык программирования; г) синтаксическое правило; д) семантика;

- 2) Дать определения следующим понятиям, привести примеры: а) машинный код; б) ассемблер и язык ассемблера; в) язык высокого уровня; г) парадигма программирования; д) декларативный язык; е) императивный язык;
- 3) Дать определения следующим понятиям, привести примеры: а) структурное программирование; б) подпрограмма (функция); в) компилятор; г) интерпретатор; д) лексема; е) алфавит языка; ё) идентификатор; ж) ключевое слово; з) числовая константа; к) комментарий;
- 4) Дать определения следующим понятиям, привести примеры: а) переменная; б) именованная константа; в) инициализатор; г) тип данных; д) интегрированная среда разработки; е) форматная строка; ё) L-значение.
- 5) Структура программы на языке C/C++?
- 6) Из каких символов может состоять идентификатор?
- 7) Зачем в начале программы используют директиву #include?
- 8) Какая функция предназначена для вывода информации на экран?
- 9) Какая функция используется для ввода информации с клавиатуры?
- 10) Отличия ввода-вывода в языке C/C++? В чём оно заключается?
- 11) Какие условные операторы имеются в C/C++?
- 12) Какие операторы цикла имеются в C/C++?
- 13) Какие операторы управления в C/C++ вы знаете?
- 14) В каких случаях используется оператор break?
- 15) Что такое указатель и ссылка?
- 16) Каким образом создаются динамические массивы в языках C/C++?
- 17) Какие методы сортировки массива вы знаете? Прокомментируйте их особенности?
- 18) Какие пользовательские типы данных вы знаете?
- 19) Что такое тип данных «перечисление» (приведите пример)?
- 20) Что такое тип данных «структура» (приведите пример)?
- 21) Что такое тип данных «объединение» (приведите пример)?
- 22) Каков порядок работы с файлами?
- 23) Что является основой модульного программирования?
- 24) Что такое функция и рекурсивная функция?
- 25) Что такое встраиваемая функция и в каких случаях она используется?
- 26) Что понимается под перегрузкой функций?
- 27) Что такое стек и как он используется в C/C++?
- 28) Какие способы передачи параметров функций вы знаете и каковы их особенности?
- 29) основополагающие принципы объектно-ориентированного программирования?
- 30) Что такое инкапсуляция, наследование и полиморфизм?
- 31) Каким образом определяется тип данных «класс» (приведите пример)?
- 32) Что такое конструктор класса и для чего он предназначен?
- 33) Основные свойства конструктора?
- 34) Что такое деструктор класса и для чего он предназначен?

- 35) Что такое указатель `this`?
- 36) Что понимается под перегрузкой операций (приведите примеры)?
- 37) Какие операции не подлежат перегрузки?
- 38) Что такое дружественные функции?
- 39) Каким образом помечаются дружественные функции?
- 40) Что понимается под наследованием классов?
- 41) Что понимается под общедоступным наследованием?
- 42) Что такое шаблоны типа?
- 43) Что такое контейнерный класс (приведите примеры)?
- 44) Каким образом задаются шаблоны типа?
- 45) Что такое библиотека STL (каков её состав)?
- 46) Что такое полиморфизм и для чего он предназначен?
- 47) Что понимается под ранним и поздним связыванием?
- 48) Каким механизмом языка C++ обеспечивается динамическое связывание?
- 49) Что такое виртуальные функции и для чего они предназначены?
- 50) Что означает служебное слово `virtual` в описании функции или класса?
- 51) Что понимается под абстрактным классом?
- 52) Что понимается под интерфейсом абстракции?
- 53) Какие виды абстракций вам известны. Приведите примеры?
- 54) Абстракция массив и ее характеристики (свойства)?
- 55) Абстракция стек и ее характеристики (свойства)
- 56) Абстракция множество и ее характеристики (свойства)?
- 57) Абстракция отображение и ее характеристики (свойства)?
- 58) Что такое бинарное дерево поиска?
- 59) Каковы алгоритмы поиска, вставки и удаления элемента?
- 60) Что такое вырожденное бинарное дерево и какие методы борьбы с этим существуют?
- 61) Что такое граф?
- 62) Что понимается под степенью входа и выхода вершины?
- 63) В каком случае граф является полным?
- 64) Что такое петля на графе?
- 65) Какая классификация графов вам известна, что она означает?
- 66) Какие способы представления графа вы знаете?
- 67) Что такое матрица смежности?
- 68) Что такое матрица инцидентности?
- 69) Что такое список смежности (инцидентности)?
- 70) Что такое список рёбер?
- 71) Основные алгоритмы обхода графа?
- 72) В чём сущность алгоритма поиска в ширину?
- 73) В чём сущность алгоритма поиска в глубину?
- 74) Основные применения алгоритмов поиска в ширину и глубину?
- 75) Что такое алгоритм Дейкстры, его сущность?
- 76) Какие встроенные типы представления данных имеются в Python?
- 77) Чем характеризуется объект в Python?
- 78) Что такое идентификатор в Python?

- 79) Какие типы объектов являются изменяемые, а какие не изменяемые?
- 80) Что такое последовательность?
- 81) Что представляет из себя строка, список, кортеж и словарь?
- 82) Что такое контейнер (коллекция)?
- 83) Что из себя представляет программа на Python?
- 84) Перечислите основные алгоритмические конструкции в Python?
- 85) Какие типы циклов имеются в Python?
- 86) Что такое функция и как она определяется в Python?
- 87) Что такое исключения в Python?
- 88) Перечислите порядок вычислений операндов в выражениях в Python.
- 89) Какие ограничения существуют в Python для имен (идентификаторов)?
- 90) Какие пространства имен существуют в Python?
- 91) Использование встроенных типов данных (int и long)?
- 92) Использование встроенных типов данных (float и complex)?
- 93) Использование встроенных типов данных (bool)?
- 94) Использование встроенных типов данных (string и unicode)?
- 95) Использование встроенных типов данных (tuple)?
- 96) Использование встроенных типов данных (list)?
- 97) Что означает и примеры взятие элемента по индексу и срезы?
- 98) Использование встроенных типов данных (dict)?
- 99) Использование встроенных типов данных (file)?
- 100) Что такое генератор в Python?
- 101) В чём заключается сущность генераторов в Python?
- 102) Что такое сопрограммы в Python?
- 103) Для чего используются сопрограммы?
- 104) Как в Python задаются классы?
- 105) Классы. Правила видимости и наследование?
- 106) Что означает стиль программирования?
- 107) Перечислите наиболее существенные положения стиля программирования на языке Python.
- 108) Разработайте алгоритм и программу вывода таблицы умножения на языке Python.
- 109) Что такое модуль в Python?
- 110) Модули стандартной библиотеки?
- 111) Назначение и содержание модуля sys?
- 112) Назначение и содержание модуля copy?
- 113) Назначение и содержание модуля math и cmath?
- 114) Назначение и содержание модуля random?
- 115) Назначение и содержание модуля os?
- 116) Назначение и содержание модуля shelve?
- 117) Назначение и содержание модуля csv?
- 118) Назначение и содержание модуля Numeric?
- 119) Каким образом передаются параметры командной строки в Python?
- 120) Какие параметры передаются в командной строке в Python?
- 121) Какой модуль Python обеспечивает работу (методы) с командной строкой?

- 122) Как осуществляется доступ к переменным окружения?
- 123) Какие аргументы имеются в функции `open()` в Python (прокомментируйте их назначение)?
- 124) Какие методы и объекты файлов вы знаете?
- 125) Каким образом в Python можно вставлять переменные в строки при выдаче?
- 126) Какие методы реализуются модулем `pickle`?
- 127) Каково назначение модуля `re`?
- 128) Каков синтаксис регулярных выражений в Python?
- 129) Основные методы модуля `re` (их назначение)?
- 130) Какой пакет Python реализует высокоуровневый интерфейс для взаимодействий с серверами HTTP, FTP и доступа к локальным файлам?
- 131) Каким образом в Python достигается множественное наследование?
- 132) Каков порядок поиска атрибутов при множественном наследовании?
- 133) Что понимается под динамическом связыванием и динамической типизацией?
- 134) Что понимается под статическими методами, для чего они используются?
- 135) Что понимается под методами класса, каким образом они определяются и чем отличаются от методов экземпляров?
- 136) Что такое свойство атрибута в классе Python?
- 137) Что такое дескриптор в классе Python?
- 138) Что понимается под инкапсуляцией данных в Python?
- 139) Что понимается под управлением памятью объектов?
- 140) Каким образом атрибуты объектов реализованы в Python и каким образом производится связывание атрибутов?
- 141) Что означает определение в классе переменной `__slots__`?
- 142) В чём существо перегрузки операторов в Python?
- 143) Какие функции используются для проверки принадлежности к классу?
- 144) Что такое абстрактный базовый класс и каким образом он определяется?
- 145) Что понимается под метаклассом, в чём его сущность?
- 146) Какие форматы используются для передачи данных через Интернет?
- 147) Какие модули стандартной библиотеки Python обеспечивают поддержку наиболее популярных форматов?
- 148) Какой функционал обеспечивает модуль `HTMLParser`?
- 149) Что из себя представляет формат JSON и какой модуль стандартной библиотеки Python обеспечивают его поддержку?
- 150) Какой функционал обеспечивает модуль `json`?
- 151) Что из себя представляет формат XML?
- 152) Какие виды поддержки формата XML обеспечивает Python?
- 153) Какие модули стандартной библиотеки Python обеспечивают поддержку парсинга XML-документов?
- 154) Какой функционал обеспечивает модуль `xml.dom.minidom`?
- 155) Какой функционал обеспечивает модуль `xml.etree.ElementTree`?

- 156) Какую функциональность реализует модуль tkinter?
- 157) Что является главным элементом GUI tkinter?
- 158) Какой класс является базовым для любого tkinter приложения?
- 159) Что такое виджет и какие виды виджетов существуют в tkinter?
- 160) Tkinter. Приведите примеры виджетов и функционал ими реализуемый.
- 161) Tkinter. Какими общими свойствами обладают все виджеты?
- 162) Tkinter. Что такое упаковщики, каков их функционал?
- 163) Какие упаковщики существуют в tkinter, в чём их отличия?
- 164) Tkinter. Что такое привязка событий к виджету, каким образом это реализуется?
- 165) Какие классы в tkinter обеспечивают работу с изображениями?
- 166) Что такое ttk и чем виджеты tkinter отличаются от ttk?
- 167) Tkinter. Какую функциональность реализует класс Canvas?
- 168) Tkinter. Какую функциональность реализуют методы create_line, create_rectangle, create_polygon?
- 169) Tkinter. Какую функциональность реализуют методы create_oval, create_arc, create_text?
- 170) Canvas. Идентификаторы, теги и анимация?
- 171) Модуль messagebox – стандартные диалоговые окна?
- 172) Модуль filedialog – диалоговые окна открытия и сохранения файлов?
- 173) Tkinter. Виджет Menu?
- 174) Tkinter. Каким образом создаются в Menu вложенные меню?
- 175) Функциональность и особенности модуля tkinter.ttk?
- 176) Каково назначение опции activeforeground и activebackground в виджетах tkinter.ttk?

Методические указания для обучающихся по освоению дисциплины

**ИНСТИТУТ ИНФОКОММУНИКАЦИОННЫХ СИСТЕМ И
ТЕХНОЛОГИЙ
КАФЕДРА ПРИКЛАДНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ОБУЧАЮЩИХСЯ ПО
ОСВОЕНИЮ ДИСЦИПЛИНЫ**

«Основы алгоритмизации и программирования»

(Приложение 2 к рабочей программе)

Направление подготовки: 09.03.02 Информационные системы и технологии

Профиль: высокопроизводительные вычислительные и телекоммуникационные интеллектуальные системы и комплексы

Уровень высшего образования: бакалавриат

Форма обучения: очная, заочная.

1. Общие положения

Целью изучения дисциплины является приобретение теоретических и практических навыков в области технологии алгоритмизации и программирования на языках высокого уровня для решения прикладных задач в профессиональной деятельности.

Задачи дисциплины:

1. формирование знаний по алгоритмизации и программированию, о стиле написания программ, о рациональных методах их разработки и оптимизации, о стратегии отладки и тестирования программ;
2. изучение структур данных в памяти и в файлах и алгоритмов работы с ними;
3. знакомство с основными принципами организации хранения и поиска данных, алгоритмами сортировки и поиска;
4. изучение основ технологии программирования и методов решения вычислительных задач и задач обработки символьных данных;
5. приобретение навыков использования базового набора фрагментов и алгоритмов в процессе разработки программ, навыков анализа и «чтения» программ.

2. Указания по проведению практических (семинарских) занятий

Практическое занятие 1.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа

Тема и содержание практического занятия: Создание простого консольного приложения. Ввод с клавиатуры, вывод на консоль. Определение функции и ее вызов. Формальные и фактические параметры функции. Функции, возвращающие значение.

Продолжительность занятия – 4/1 ч

Практическое занятие 2.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: групповая дискуссия

Тема и содержание практического занятия: Реализация алгоритмов работы с последовательностью: нахождение суммы элементов, среднего значения. Реализация алгоритмов с условием: нахождение минимального (максимального) значения и т.д.

Продолжительность занятия – 4/1 ч

Практическое занятие 3.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа

Тема и содержание практического занятия: Реализация циклических алгоритмов. Заполнение массива. Генерация случайных значений. Передача

массива в функцию. Подпрограммы обработки элементов массива. Вложенные циклы.

Продолжительность занятия – 8/2 ч

Практическое занятие 4.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа

Тема и содержание практического занятия: Работа с файлами. Понятие потока. Открытие и закрытие потоков. Считывание из потока и вывод в поток.

Продолжительность занятия – 8/2 ч

Практическое занятие 5.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа

Тема и содержание практического занятия: Алгоритмы работы со строками. Разбиение и объединение строк, поиск и извлечение подстроки, удаление и вставка подстроки.

Продолжительность занятия – 8/2 ч

Практическое занятие 6.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: групповая дискуссия

Тема и содержание практического занятия: Реализация синтаксического анализа простого выражения методом рекурсивного спуска.

Продолжительность занятия – 8/2 ч

Практическое занятие 7.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: групповая дискуссия

Тема и содержание практического занятия: Реализация и сравнение рекурсивных и итеративных алгоритмов.

Продолжительность занятия – 8/2 ч

Практическое занятие 8.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: групповая дискуссия

Тема и содержание практического занятия: Реализация простых алгоритмов сортировки массива (выбором, вставками, пузырьком).

Продолжительность занятия – 8/2 ч

Практическое занятие 9.

Вид практического занятия: смешанная форма практического занятия.

Образовательные технологии: командная работа

Тема и содержание практического занятия: Реализация стеков и других структур данных на основе массивов. Введение в обработку списков.

Продолжительность занятия – 8/2 ч

3. Указания по проведению лабораторного практикума

Не предусмотрено учебным планом.

Указания по курсовой работе.

Не предусмотрено учебным планом.

4. Указания по проведению самостоятельной работы студентов

№ п/п	Наименование блока (раздела) дисциплины	Виды СРС
1	Тема 1. Основы алгоритмизации	<p>Самостоятельное изучение тем, выполнение практических заданий.</p> <p>Примерная тематика практических заданий:</p> <ol style="list-style-type: none">1. Составить алгоритм и написать программу вычисления значения функции по заданному значению переменных: $e^x - \frac{y^2 + 12xy - 3x^2}{18y - 1}$$2ctg(3x) - \frac{\ln \cos x}{\ln(1 + x^2)}$2. Записать арифметическое выражение на алгоритмическом языке и указать порядок выполнения операций: $D = \frac{\sqrt{x + b - a} + \ln y}{arctg(b + a)}$3. Составить алгоритм и написать программу, реализующую решение задачи: дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.4. Составить программу вычисления объема цилиндра и конуса, которые имеют одинаковую высоту H и одинаковый радиус основания R.
2	Тема 2. Основы языков программирования	<p>Самостоятельное изучение тем, выполнение практических заданий.</p> <p>Примерная тематика практических заданий:</p> <ol style="list-style-type: none">1. Составить структурную схему алгоритма и проект программы для решения задачи: (исходные данные ввести с клавиатуры, результат вывести на экран) четырехугольник задан координатами своих вершин. Найти его периметр. Вершины ввести с клавиатуры.2. Написать программу определения катета и площади прямоугольного треугольника, по заданным катету и гипотенузе. Длины катета и гипотенузы ввести с экрана.3. Написать программу для решения задачи нахождения корней квадратного уравнения, введя с экрана коэффициенты a, b, c (коэффициент a не равен 0), для которых дискриминант положителен
3	Тема 3.	Самостоятельное изучение тем, выполнение практических

	Выражения и операторы. Структурное программирование	заданий. Примерная тематика практических заданий: 1. Написать программу, которая по введенному номеру единицы измерения (1 – дециметр, 2 – километр, 3 – метр, 4 – миллиметр, 5 – сантиметр) или длине отрезка L выдавала бы соответствующее значение длины отрезка в метрах. 2. Составить блок-схему и написать программу, которая при вводе числа в диапазоне от 1 до 99 добавляет к нему слово "копейка" в правильной форме. Например, 1 копейка, 5 копеек, 42 копейки. 3. Написать блок-схему и программу для решения задачи: дано трехзначное число. Определить: а) верно ли, что все его цифры одинаковые? б) есть ли среди его цифр одинаковые? 4. Составить блок-схему и написать программу с использованием оператора выбора варианта для решения задачи: дано целое число k ($1 < k < 365$). Определить, каким днем недели (понедельник, вторник, воскресенье) является k-день невысокосного года, если 1 января – понедельник.
4	Тема 4. Массивы	Самостоятельное изучение тем, выполнение практических заданий. Примерная тематика практических заданий: 1. Составить блок-схему и написать программу для решения задачи: дана действительная матрица размером $n \times m$. Требуется преобразовать матрицу: поэлементно вычесть последнюю строку из всех строк, кроме последней. 2. Составить программу, которая заполняет квадратную матрицу порядка n натуральными числами 1, 2, 3, ..., n^2 , записывая их «по спирали». 3. Написать блок-схему и программу для решения задачи: все массивы в заданиях объявить, как динамические. Ввести массив из 11 вещественных чисел. Создать новый массив из элементов исходного, не превышающих среднее арифметическое массива. Определить наименьший положительный элемент нового массива.
5	Тема 5. Циклические конструкции	Самостоятельное изучение тем, выполнение практических заданий. Примерная тематика практических заданий: 1. Написать блок-схему и составить программу, которая запрашивает пароль (четырёхзначное число) до тех пор, пока количество неверных ответов не превысит три. 2. Составить программу перевода данного натурального числа из десятичной системы счисления в двоичную. 3. Написать программу поиска произведения последовательности чисел, вводимых с клавиатуры, предшествующих первому введенному нулю. Контрольный пример: 1,2,3,-4,5,-2,0.
6	Тема 6. Строки	Самостоятельное изучение тем, выполнение практических заданий.

		<p>Примерная тематика практических заданий:</p> <ol style="list-style-type: none"> 1. Написать программу, которая во введенной с клавиатуры строке преобразует буквы русского алфавита в латинские. 2. Написать программу, которая во введенной с клавиатуры строке меняет четные слова с нечетными. 3. Напишите программу, в которой с клавиатуры вводятся две строки символов. К строке с наибольшей длиной добавить текст, содержащийся в другой строке. 4. Напишите программу, в которой с клавиатуры вводятся две строки символов. К строке с наименьшей длиной добавить текст, содержащийся в другой строке.
7	Тема 7. Функции	<p>Самостоятельное изучение тем, выполнение практических заданий.</p> <p>Примерная тематика практических заданий:</p> <ol style="list-style-type: none"> 1. Составить схему алгоритма и написать программу вычисления таблицы значений функции $f(x)$ на промежутке от -2 до 3 с шагом 0,5: $f(x) = \sum_{k=1}^5 \frac{(-2)^{k+1}(k+1)!}{x+2.5)^{k+1}}$ 2. Составить схему алгоритма и написать программу вычисления таблицы значений функции $f(x)$ на промежутке от -2 до 2 с шагом 0,4: $f(x) = \begin{cases} \sqrt[5]{x^3}, & \text{если } x > 0 \\ x^2 + \sum_{k=1}^3 \frac{x^k}{k}, & \text{если } x \leq 0 \end{cases}$
8	Тема 8. Алгоритмы сортировки	<p>Самостоятельное изучение тем, выполнение практических заданий.</p> <p>Примерная тематика практических заданий:</p> <ol style="list-style-type: none"> 1. Дана последовательность чисел a_1, a_2, \dots, a_n. Требуется расположить элементы по убыванию. Для этого в массиве, начиная с первого, выбирается наибольший элемент и ставится на первое место, а первый – на место наибольшего. Написать алгоритм сортировки выбором. 2. Дана последовательность чисел a_1, a_2, \dots, a_n. Требуется представить числа в порядке возрастания. Для этого сравниваются два соседних числа a_i и a_{i+1}. Если $a_i > a_{i+1}$, то делается перестановка. Так продолжается до тех пор, пока все элементы не будут расположены в порядке возрастания. Составить алгоритм сортировки обменами, подсчитывая при этом количество перестановок. 3. Дана последовательность чисел a_1, a_2, \dots, a_n. Требуется представить числа в порядке возрастания. Делается это следующим образом. Пусть a_1, a_2, \dots, a_i – упорядоченная последовательность, т.е. $a_1 \leq a_2 \leq \dots \leq a_i$. Берется следующее число a_{i+1} и вставляется в последовательность так, чтобы новая последовательность была тоже возрастающей. Процесс производится до тех пор, пока все элементы от $i+1$ до n не будут перебраны. Составить

		<p>алгоритм сортировки обменами.</p> <p>4. Дан массив n действительных чисел. Требуется упорядочить его по возрастанию. Делается это следующим образом: сравниваются два соседних элемента a_i и a_{i+1}. Если $a_i \leq a_{i+1}$, то продвигаются на один элемент вперед. Если $a_i > a_{i+1}$, то производится перестановка и сдвигаются на один элемент назад. Составить алгоритм сортировки Шелла.</p>
9	Тема 9. Структуры данных	<p>Самостоятельное изучение тем, выполнение практических заданий.</p> <p>Примерная тематика практических заданий:</p> <p>1. Дан набор из 10 чисел. Создать две очереди: первая должна содержать числа из исходного набора с нечетными номерами (1, 3, ..., 9), а вторая — с четными (2, 4, ..., 10); порядок чисел в каждой очереди должен совпадать с порядком чисел в исходном наборе. Вывести указатели на начало и конец первой, а затем второй очереди.</p> <p>2. Создать стек целочисленных значений, для реализации используя односвязные списки. Реализовать операции добавления (push) и удаления (pop) элемента из стека. Добавьте в стек числа 4, 3, 1, 2, 4 и распечатайте содержимое стека. Удалите один элемент из стека, и распечатайте содержимое стека еще раз. Найдите минимальный элемент, принадлежащий стеку.</p> <p>3. Описать структуру с именем STUDENT, содержащую поля: Name – фамилия и инициалы, Kurs – курс, SES – успеваемость (массив из пяти элементов). Написать программу с использованием структуры, выполняющую:</p> <ul style="list-style-type: none"> – ввод с клавиатуры данных в массив STUD, состоящий из 10 структур типа STUDENT, записи должны быть упорядочены по алфавиту; – вывод на экран записей, упорядоченного списка студентов, средний бал которых превышает общий средний бал; – если таких студентов нет – выдать сообщение

5. Перечень основной и дополнительной учебной литературы

Основная литература:

1. Белов, В. В. Алгоритмы и структуры данных : учебник / В. В. Белов, В. И. Чистякова. - Москва : КУРС : ИНФРА-М, 2020. - 240 с. - (Бакалавриат). - ISBN 978-5-906818-25-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1057212> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
2. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В. Д. Колдаев ; под ред. проф. Л. Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1151517> (дата обращения: 27.09.2021). – Режим доступа: по подписке.

Дополнительная литература:

1. Канцедал, С. А. Алгоритмизация и программирование : учебное пособие / С.А. Канцедал. — Москва : ФОРУМ : ИНФРА-М, 2021. — 352 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0727-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1189320> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
2. Игнашева, Е. П. Системы счисления, алгоритмизация и программирование : учебное пособие / Е.П. Игнашева. — Москва : ИНФРА-М, 2020. — 224 с. — (Военное образование). - ISBN 978-5-16-015295-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1078360> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
3. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 343 с. — (Среднее профессиональное образование). - ISBN 978-5-00091-553-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1042452> (дата обращения: 23.10.2020). – Режим доступа: по подписке.
4. Нидхем, М. Графовые алгоритмы : практическое руководство / М. Нидхем, Э. Ходлер ; пер. с англ. В. С. Яценкова. - Москва : ДМК Пресс, 2020. - 258 с. - ISBN 978-5-97060-799-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1094928> (дата обращения: 27.09.2021). – Режим доступа: по подписке.

Электронные книги:

1. Колдаев, В. Д. Основы алгоритмизации и программирования : учебное пособие / В. Д. Колдаев ; под ред. проф. Л. Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2021. — 414 с. — (Среднее профессиональное образование). - ISBN 978-5-8199-0733-7. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1151517> (дата обращения: 27.09.2021). – Режим доступа: по подписке.
2. Мишенин, А. И. Сборник задач по программированию : учебное пособие / А. И. Мишенин. - Москва : Финансы и Статистика, 2021. - 224 с. - ISBN 978-5-00184-039-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1541948> (дата обращения: 27.09.2021). – Режим доступа: по подписке.
3. Долгов, А. И. Алгоритмизация прикладных задач : учебное пособие / А. И. Долгов. - 3-е изд., стер. - Москва : Флинта, 2021. - 136 с. - ISBN 978-5-9765-0086-2. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1843168> (дата обращения: 27.09.2021). – Режим доступа: по подписке.

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

Интернет-ресурсы:

<http://www.intuit.ru/studies/courses/555/411/info> – Алгоритмы и теория вычислений.

<http://www.intuit.ru/studies/courses/1106/146/info> – Основы теории вычислимых функций

<http://www.infojournal.ru> – Научно-образовательный портал,

<http://www.interface.ru/> – Научно-образовательный портал.

<http://www.codeblocks.org/>

<https://code.visualstudio.com/>

<http://www.cplusplus.com/> - Сеть ресурсов, посвященных языку C++

<http://pythontutor.ru/> - Курс по программированию на языке Python

<https://www.python.org/> – Сайт разработчиков Python

<https://jupyter.org/>

<https://colab.research.google.com/>

7. Перечень информационных технологий

Перечень программного обеспечения:

LibreOffice, VS Code, GCC + MinGW, Python 3, PyCharm

Информационные справочные системы:

1. Электронные ресурсы образовательной среды Технологический университет.
2. Рабочая программа и методическое обеспечение по дисциплине «Основы алгоритмизации и программирования»