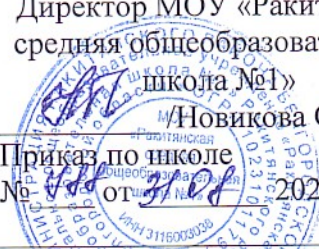


Муниципальное общеобразовательное учреждение
«Ракитянская средняя общеобразовательная школа № 1»
Ракитянского района Белгородской области
Центр поддержки одаренных детей

<p>«Рассмотрено» на заседании Педагогического совета МОУ «Ракитянская СОШ №1» Протокол № <u>1</u> от « <u>29</u> » <u>августа</u> 2023 г.</p>	<p>«Согласовано» Заместитель директора МОУ «Ракитянская средняя общеобразовательная школа №1» <u>Жернова Е.Е.</u> /Жерновая Е.Е./ « <u>29</u> » <u>августа</u> 2023 г.</p>	<p>«Утверждено» Директор МОУ «Ракитянская средняя общеобразовательная школа №1» <u>Новикова О.П.</u> Приказ по школе № <u>488</u> от <u>30.08</u> 2023 г.</p> 
--	---	--

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА
«ПОДГОТОВКА К ОЛИМПИАДЕ ПО ИНФОРМАТИКЕ»
естественнонаучной направленности
на 2023-2024 учебный год
возраст обучающихся - 14-18 лет**

Автор-составитель
Сакова Оксана Анатольевна
(ФИО педагога-разработчика)

п. Ракитное, 2023 г.

Пояснительная записка

Рабочая программа дополнительного образования «Подготовка учащихся к Всероссийской олимпиаде школьников по информатике» разработана на основе:

- Федерального закона Российской Федерации «Об образовании в Российской Федерации» (№ 273-ФЗ от 29.12.2012).
- Федерального компонента государственного образовательного стандарта (Приказ Минобрнауки РФ от 05.03.2004 года № 1089)
- Примерной программы основного общего образования по информатике и информационным технологиям (приказ Минобрнауки РФ № 1312 от 09.03.2004)

Учебная программа «Программа подготовки учащихся ко Всероссийской олимпиаде школьников по информатике» определяет содержание, объем и требования, к изучению программирования.

Программа разработана на основе рабочей программы элективного курса по выбору «Подготовка учащихся к Всероссийской олимпиаде по информатике» (автор – Симонова О.Ю., Кидра Т.В.). Данная рабочая программа имеет изменения, по сравнению с авторской. На изучение дается один час в неделю. Итого вся программа рассчитана на 34 часа.

Цель курса определяется постоянным ростом практической и теоретической значимости программирования в жизни современного общества, глубоким проникновением информационно коммуникационных средств во все сферы человеческой деятельности.

Изучение программирования в общеобразовательных учебных заведениях служит общим целям образования и воспитания гармонично развитой личности, способной решать задачи научно-технического прогресса.

Программа охватывает весь учебный материал базового курса алгоритмизации и программирования из общей программы по информатике. При этом ученики должны не только достичь результатов обучения, предусмотренных программой, но и овладеть соответствующими знаниями, умениями и навыками, на более высоком уровне, что характеризуется, в первую очередь, способностью решать более сложные, нестандартные, олимпиадные задачи.

В программу изучения олимпиадного программирования включены разделы, которые дополняют программу базового курса, имеет целью способствовать достижению учениками высокого уровня подготовки, получению учениками качественного образования с учетом их индивидуальных способностей.

В задачи курса входит:

- Формирование у школьников информационной культуры;
- Разъяснение роли информатики и компьютерной техники в развитии общества и ускорении научно-технического прогресса, открытия отечественных ученых, в развитие компьютерной техники и реализация на этой основе патриотического воспитания;
- Формирование представления об этапах решения задач на компьютере, методы построения математических моделей и алгоритмов, умения пользоваться средами программирования для реализации алгоритмов решения задач;
- Формирование умений разработки и отладки программ языками программирования высокого уровня в средах процедурного и визуального программирования;
- Ознакомление с методами оценки эффективности алгоритмов, классическими алгоритмами, правилами и приемами программирования;
- Развитие познавательного интереса к технике, творческим способностям, подготовка к сознательному выбору профессии на основе тесной связи информатики с жизнью.

Целью курса является формирование теоретической базы знаний учеников по теме алгоритмизация и **программирование**, практических навыков в использовании средств современных систем программирования, а также подготовка к самостоятельной научно-практической работе, участию, в олимпиадных соревнованиях, конкурсах, турнирах, формирование стойкого интереса к программированию и связанной с ним будущей профессиональной деятельности.

Общая характеристика

К *теоретической базе знаний* относятся:

- понятия модели и ее типов, моделирования, как метода познания и основных этапов построения модели;
- принципы оценивания информационной модели, возможности построения и исследования с помощью компьютера информационных моделей, по физике, экономике, экологии, управления и тому подобное;
- понятие языка и среды программирования;
- классификация языков программирования;
- составные языки программирования;
- общие принципы решения задач с помощью компьютера, формулировка проблем и постановка задач;
- понятие алгоритмов и их типов, базовых алгоритмических структур и их свойств;
- принципов оценивания эффективности алгоритмов;
- методы построения алгоритмов для решения задач;
- классические алгоритмы и приемы программирования.

К *практическим навыкам* относятся навыки:

- формирование, описание и реализация алгоритмов и программ, с использованием различных средств языков программирования;
- анализа и использования основных методов построения алгоритмов для решения задач повышенной сложности;
- работы с интегрированными средами процедурного и визуального программирования.

Цель программы достигается через практическое овладение учениками навыками работы с современными системами программирования, ознакомления с основными технологиями решения задач с помощью компьютера, начиная от их постановки и построения соответствующих информационных моделей и заканчивая интерпретацией результатов, полученных с помощью компьютера.

Место курса в учебном плане, среди других учебных дисциплин

Программа курса рассчитана на изучение олимпиадного программирования при условии постоянного доступа учеников к компьютерам с соответствующим программным обеспечением.

Всего курс рассчитан на 34 часа с учётом того, что учащийся может в течение всего курса изучения принимать активное участие в научно-исследовательской проектной деятельности, конкурсах и олимпиадах различного уровня. Часы, при изучении программирования по данной программе, могут варьироваться в зависимости от выбора часов по основной программе и из вариативной части программы. Данная программа может легко быть адаптирована для перехода при изучении различных языков программирования. Я считаю, что результатом изучения данного курса может быть успешная научно-исследовательская проектная деятельность, победы в конкурсах и олимпиадах различного уровня.

Описание ценностных ориентиров содержания учебного предмета

Дополнительная общеразвивающая программа «Подготовка учащихся к Всероссийской олимпиаде по информатике» является более углублённым курсом изучения программирования.

Главная цель - дать основы знаний общеобразовательного курса программирования, сформировать начальные умения и навыки разработки программ языком процедурного программирования. Особенное внимание предоставляется формированию логического (математического, алгоритмического, операционного) мышления учеников, развития их, творческого потенциала. Предусматривается углубление и расширение содержания базового курса, усиления его, научного направления, решения задач повышенной сложности.

Применение школьниками знаний по программированию, при решении задач прикладного характера, играет важную роль в осознанном выборе будущей профессии.

Основной формой учебных занятий в классах с углубленным изучением программирования остаются уроки различных типов, но в первую очередь комбинированные, а также уроки формирования умений и навыков при решении олимпиадных задач, обобщения и систематизации знаний, контроля и коррекции знаний. Во время элективного курса изучения курса программирования предусматриваются такие практические формы занятий: уроки решения олимпиадных задач, лабораторные работы, работы над проектными задачами. Высшей формой организации работы учеников может быть выполнение учениками проектов по разработке программных средств учебного назначения. Такая работа позволяет учащимся принимать участие в конкурсах, турнирах и олимпиадах различного уровня.

Учебно-тематическое планирование

№ п/п	Тема	Количество часов
1.	Модели и алгоритмы	1
2.	Язык и среда программирования	1
3.	Линейные алгоритмы	1
4.	Алгоритмы с разветвлением	1
5.	Циклические алгоритмы	2
6.	Обработка табличных величин	2
7.	Линейные и многомерные массивы	3
8.	Методы сортировки массивов	2
9.	Обработка строковых величин	2
10.	Вспомогательные алгоритмы	2
11.	Средства работы с файлами	2
12.	Анализ и построение алгоритмов	15
	12.1 Методика построения и оценка эффективности алгоритмов	2
	12.2 Элементы вычислительной геометрии	3
	12.3 Алгоритмы на графах	5
	12.4 "Жадные" алгоритмы	2
	12.5 Динамическое программирование	3
13.	Обобщение и систематизация знаний за год	1
	ВСЕГО	34

Содержание курса

Содержание учебного материала	Государственные требования к уровню общеобразовательной подготовки учеников
<p>1. Модели и алгоритмы (1 час) Этапы решения задач с использованием ЭВМ. Понятие модели. Модели материальные и информационные. Основные этапы построения математических моделей. Понятие об алгоритме. Свойства алгоритма. Исполнители алгоритмов. Примеры алгоритмов. Способы описания алгоритмов. Базо-</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие модели; • понятие алгоритма; • базовые алгоритмические структуры; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • виды моделей; • виды алгоритмических структур; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • этапы решения задач с использованием ЭВМ; • этапы построения математических моделей;

<p>вые алгоритмические структуры. Свойства базовых структур алгоритмов. Метод пошаговой детализации. Разработка алгоритма «сверху - вниз».</p>	<ul style="list-style-type: none"> • классификацию моделей; • свойства алгоритмов; • свойства базовых алгоритмических структур; • метод пошаговой детализации; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • материальных и невещественных моделей; • бытовых и вычислительных алгоритмов; • способов описания алгоритмов; • исполнителей алгоритмов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • строить математические модели для решения вычислительных задач; • описывать словесно алгоритмы решения задач; • составлять блок-схемы простых алгоритмов; <p><i>использует:</i></p> <ul style="list-style-type: none"> • блок-схемы для описания алгоритмов решение задач; • метод пошаговой детализации при разработке алгоритмов.
<p style="text-align: center;">2. Язык и среда программирования (1 час)</p> <p>Понятие программы. Языки программирования, их классификация. Характеристика языка и среды программирования. Структура программы на языке программирования.</p> <p>Практическая работа № 1. "Работа в среде программирования. Запуск программы на выполнение".</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие программы; • основные элементы языка программирования; • структуру программы языком программирования; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • понятие языка и среды программирования; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • классификацию языка программирования; • структуру программы; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • языка программирования различных видов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • управлять средой программирования для разработки и отладки программ; <p><i>использует:</i></p> <ul style="list-style-type: none"> • среду программирования для: загрузки, редактирования, отладки, выполнения, и сохранения программ.
<p style="text-align: center;">3. Линейные алгоритмы (1 час)</p> <p>Постоянные и переменные величины. типы и идентификаторы величин. Операции ввода - вывода данных. Указание присвоения. Стандартные математические операторы. Математические функции. Приоритет операций. Запись математических выражений.</p> <p>Практическая работа № 2. "Создание и реализация линейных программ"</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • правила записи идентификаторов; • правила записи математических выражений; • приоритет операций; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • постоянные и переменные величины; • типы величин; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • способы предоставления значений переменной; • особенности операции вывода данных; • особенности операции ввода данных; • особенности указания присвоения; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • математических операторов; • математических функций; • величин и их типов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • корректно выбирать типы переменных величин; • выбирать идентификаторы для переменных величин; • корректно записывать математические выражения;

	<ul style="list-style-type: none"> • разрабатывать и проверять линейные программы; <i>использует:</i> • операции ввода, вывода, и присвоения при разработке линейных программ.
<p align="center">4. Алгоритмы с разветвлением (1 час)</p> <p>Команда разветвления. Условный оператор. Полная и сокращенная форма условного оператора. Логические операции. Простое и составное условие. Логические функции. Оператор множественного выбора. Структура оператора.</p> <p>Практическая работа № 3. «Разработка и выполнение программ с условным оператором и оператором множественного выбора».</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • формат условного оператора; • формат оператора множественного выбора; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • полную и сокращенную формы условного оператора; • задачи на применение условного оператора; • задачи на применение оператора множественного выбора; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • особенности выполнения полной и сокращенной формы условного оператора; • особенности выполнения оператора множественного выбора; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • задач на применение условного оператора; • задач на применение оператора множественного выбора; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • применять условный оператор для решения задач; • корректно выбирать форму условного оператора; • корректно выбирать условие в команде разветвления; • решать и проверять задачи на применение условного оператора; • решать и проверять задачи на применение оператора множественного выбора; <p><i>использует:</i></p> <ul style="list-style-type: none"> • условный оператор и оператор множественного выбора при решении задач.
<p align="center">5. Циклические алгоритмы (2 часа)</p> <p>Указание повторения. Типы циклов. Циклы в языке программирования. Оператор цикла с предусловием и постусловием. Оператор цикла с управляющей переменной. Вложенные циклы.</p> <p>Практическая работа № 4. "Реализация циклических программ"</p> <p>Практическая работа № 5. "Использование вложенных циклов в программах".</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • формат операторов цикла с предусловием, с постусловием, с управляющей переменной; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • команды повторения разных типов; • задачи на применение циклов; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • особенности выполнения циклов разных типов; • особенности выполнения вложенных циклов; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • задач на применение циклов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • применять операторы цикла для решения задач; • корректно выбирать тип цикла; • корректно выбирать условие в цикле; • решать и проверять задачи на применение циклов; • решать и проверять задачи на применение вложенных циклов; <p><i>использует:</i></p> <ul style="list-style-type: none"> • операторы цикла при решении задач.
<p align="center">6. Обработка табличных величин (2 часа)</p> <p>Табличные величины. Одномерные таблицы. Структура описания табличных вели-</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие линейного массива; • структуру описания табличных величин; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • понятие массива, элементы массива индексы и зна-

чин. Правила обращения к данным таблицы. Ввод и вывод элементов массива. Нахождение количества, суммы и произведения элементов массива. Методы поиска. Линейный и бинарный поиск.

Практическая работа № 6.
"Ввод-вывод данных массива".

Практическая работа № 7.
"Разработка и выполнение программ с табличными величинами".

чения элементов массива;

объясняет:

- отличие между простыми и структурированными типами данных;
- правила обращения к элементам таблицы;
- алгоритмы работы с табличными величинами;
- особенности линейного и бинарного поиска в массивах;

приводит примеры:

- прикладных задач, которые требуют применения массивов;

умеет:

- корректно определять тип и описывать табличные величины;
 - осуществлять ввод и вывод табличных величин;
 - находить сумму и произведение элементов массива;
 - осуществлять линейный и бинарный поиск в массивах;
- решать задачи на обработку табличных величин.

7. Линейные и многомерные массивы (3 часа)

Нахождение наибольшего (наименьшего) элемента массива. Формирование массивов. Многомерные массивы. Двухмерный массив. Ввод и вывод элементов двухмерного массива.

Практическая работа № 8.
"Разработка и выполнение программ с массивами чисел".

Ученик описывает:

- понятие линейного и двумерного массива;
- понятие формирования массива;

распознает:

- задачи на использование линейных - одномерных и двухмерных массивов;

объясняет:

- алгоритм нахождения наибольшего (наименьшего) элемента массива;

приводит примеры:

- разных методов формирования массивов;
- задач на применение линейных и двухмерных массивов;

умеет:

- реализовывать алгоритмы поиска наибольшего (наименьшего) элементов массива;
- реализовывать алгоритмы формирования массивов;
- осуществлять ввод и вывод элементов двухмерных массивов;

использует:

- алгоритмы поиска наибольшего (наименьшего) элементов массива при решении задач;
- алгоритмы формирования массивов при решении задач.

8. Методы сортировки массивов (2 часа)

Основные методы формирования: метод "пузырька", прямого выбора вставки, обмена. Поиск и сортировка в одномерном и двухмерном массиве.

Практическая работа № 9.
"Разработка и выполнение программ с сортировкой массива".

Ученик описывает:

- основные методы формирования массива: "пузырёк", прямого выбора, вставки, обмена;

приводит примеры:

- разных методов формирования сортировки массивов;
- задач на применение сортировки линейных и двухмерных массивов;

умеет:

- реализовывать алгоритмы сортировки элементов массива;

использует:

- алгоритмы сортировки элементов линейного и мно-

<p style="text-align: center;">9. Обработка строковых величин (2 часа)</p> <p>Строковые величины. Структура описания величин строкового типа. Операции над строковыми величинами. Стандартные функции для работы с величинами строкового типа.</p> <p>Вставка и исключение элементов строковых величин. Поиск в строке. Превращение строковых и числовых величин.</p> <p>Практическая работа № 10. "Разработка и выполнение программ со строковыми величинами".</p>	<p>гомерного массивов при решении задач;</p> <p>Ученик описывает:</p> <ul style="list-style-type: none"> • понятие строковой величины; • структуру описания величины строкового типа; • стандартные функции для работы с величинами строкового типа; <p>объясняет:</p> <ul style="list-style-type: none"> • особенности выполнения операций со строковыми величинами; • алгоритмы поиска в строке; • специфику превращения строковых и числовых величин; <p>приводит примеры:</p> <ul style="list-style-type: none"> • задач на применение строковых величин; <p>умеет:</p> <ul style="list-style-type: none"> • использовать стандартные функции для работы с величинами строкового типа; • осуществлять поиск в строке; • выполнять превращение строковых и числовых величин; • решать задачи на обработку строковых величин.
<p style="text-align: center;">10. Вспомогательные алгоритмы (2 часа)</p> <p>Понятие и виды вспомогательных алгоритмов. Правила описания и обращение к вспомогательным алгоритмам в языке программирования. Локальные и глобальные переменные. Передача параметров в подпрограмму. Фактические и формальные параметры. Понятие рекурсии и её использование для решения задач.</p> <p>Практическая работа № 11. "Разработка и выполнение программ с использованием вспомогательных алгоритмов".</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • понятие и виды вспомогательных алгоритмов; • понятия локальных и глобальных переменных; • понятия формальных и фактических параметров; • понятие рекурсии; <p>распознает:</p> <ul style="list-style-type: none"> • локальные и глобальные переменные в программах; • формальные и фактические параметры; <p>объясняет:</p> <ul style="list-style-type: none"> • правила описания вспомогательных алгоритмов на языке программирования; • правила обращения к вспомогательным алгоритмам на языке программирования; • методы передачи параметров в подпрограмму; • особенности описания и выполнения рекурсивных функций; <p>приводит примеры:</p> <ul style="list-style-type: none"> • задач на применение вспомогательных алгоритмов; • задач на применение рекурсии; <p>умеет:</p> <ul style="list-style-type: none"> • корректно выделять в задачах локальные и глобальные переменные; • описывать вспомогательные алгоритмы; • обращаться из программы к программе и корректно передавать параметры; • описывать рекурсивные функции и обращаться к ним; <p>использует:</p> <ul style="list-style-type: none"> • подпрограммы при решении задач; • рекурсивные функции при решении задач.
<p style="text-align: center;">11. Средства работы с файлами (2 часа)</p> <p>Понятие файла. Типы файлов. Работа с файловыми структурами данных средствами языка программирования. Осо-</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • понятия файла, типы файлов; • стандартные функции для работы с файлами; <p>объясняет:</p> <ul style="list-style-type: none"> • особенности работы с текстовыми файлами; • правила использования текстовых файлов для ввода и вывода данных;

<p>бенности работы с текстовыми файлами.</p>	<ul style="list-style-type: none"> • преимущества использования файлов для ввода и вывода данных при отладке программы; <p>умеет:</p> <ul style="list-style-type: none"> • применять текстовые файлы для ввода и вывода данных; • решать и проверять задачи с введением и выводением данных во внешние файлы; <p>использует:</p> <p>текстовые файлы для ввода и вывода данных.</p>
<p>12. Анализ и построение алгоритмов (15 часов)</p> <p>12.1 Методика построения и оценка эффективности алгоритмов (2 часа)</p> <p>Выбор метода решения задачи. Анализ эффективности алгоритма решения задачи. Классификация алгоритмов. Поиск оптимального алгоритма решения. Пошаговая детализация, планирование и представление алгоритма. Вспомогательные задачи. Обобщение и анализ экстремальных ситуаций. Подборка системы тестов.</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • классификацию алгоритмов; <p>распознает:</p> <ul style="list-style-type: none"> • принадлежность алгоритма к определенному виду; <p>объясняет:</p> <ul style="list-style-type: none"> • этапы разработки алгоритма; • понятие эффективности и результативности алгоритма; • методы оценивания эффективности алгоритма; <p>приводит примеры:</p> <ul style="list-style-type: none"> • алгоритмов определенных видов; • методов оценивания эффективности алгоритма. <p>умеет:</p> <ul style="list-style-type: none"> • анализировать эффективность алгоритмов; • осуществлять пошаговую детализацию алгоритмов; • анализировать экстремальные ситуации; • выбирать систему тестов для анализа решения задачи; <p>использует:</p> <ul style="list-style-type: none"> • методы оценки эффективности алгоритма при решении задач.
<p>12.2 Элементы вычислительной геометрии (3 часа)</p> <p>Основные формулы аналитической геометрии. Нахождение длины отрезка в N-измеримом пространстве. Расстояние от точки к прямой. Координаты точек пересечения отрезков и прямых. Определение положения точки относительно многоугольника. Нахождение площади многоугольника: метод триангуляции, метод трапеций. Векторная геометрия. Проверка принадлежности точек прямой. Векторное произведение. Направление поворота. Нахождение порядка обхода вершин выпуклого многоугольника. Задачи минимизации в геометрической интерпретации.</p> <p>Практическая работа № 12. "Разработка и выполнение программ с элементами вычислительной геометрии".</p> <p>Практическая работа № 13. "Разработка и выполнение программ с эле-</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • основные формулы вычислительной геометрии; • основные формулы векторной геометрии; <p>объясняет:</p> <ul style="list-style-type: none"> • метод определения положения точки относительно многоугольника; • методы нахождения площади многоугольника: метод триангуляции, метод трапеций; • методы проверки принадлежности точек прямой; • методы нахождения порядка обхода вершин выпуклого многоугольника; • методы определения координат вершин выпуклого многоугольника; <p>приводит примеры:</p> <ul style="list-style-type: none"> • формул из вычислительной и векторной геометрии; <p>умеет:</p> <ul style="list-style-type: none"> • применять формулы из вычислительной и векторной геометрии для решения задач; • реализовывать методы нахождения площади многоугольника; • реализовывать методы проверки принадлежности точек прямой; • реализовывать методы нахождения порядка обхода вершин выпуклого многоугольника; • реализовывать методы определения координат вершин выпуклого Многоугольника • решать и проверять задачи минимизации в геометрической

ментами векторной геометрии".	интерпретации; <i>использует:</i> • формулы и методы из вычислительной и векторной геометрии для решения задач.
-------------------------------	---

Требования к уровню подготовки учащихся

В результате изучения курса ученик должен

знать/понимать:

- основные конструкции языка программирования;
- свойства алгоритмов и основные алгоритмические конструкции; тезис о полноте формализации понятия алгоритма;
- виды и свойства информационных моделей реальных объектов и процессов, методы и средства компьютерной реализации информационных моделей;
- нормы информационной этики и права, информационной безопасности, принципы обеспечения информационной безопасности;
- способы и средства обеспечения надежного функционирования средств ИКТ;

уметь:

- выделять информационный аспект в деятельности человека; информационное взаимодействие в простейших социальных, биологических и технических системах;
- строить информационные модели объектов, систем и процессов, используя для этого типовые средства (язык программирования, таблицы, графики, диаграммы, формулы и т.п.);
- вычислять логическое значение сложного высказывания по известным значениям элементарных высказываний;
- проводить статистическую обработку данных с помощью компьютера;
- интерпретировать результаты, получаемые в ходе моделирования реальных процессов;
- устранять простейшие неисправности, инструктировать пользователей по базовым принципам использования ИКТ;
- оценивать числовые параметры информационных объектов и процессов: объем памяти, необходимый для хранения информации; скорость передачи и обработки информации;
- оперировать информационными объектами, используя имеющиеся знания о возможностях информационных и коммуникационных технологий, в том числе создавать структуры хранения данных; пользоваться справочными системами и другими источниками справочной информации; соблюдать права интеллектуальной собственности на информацию;
- проводить виртуальные эксперименты и самостоятельно создавать простейшие модели в учебных виртуальных лабораториях и моделирующих средах;
- выполнять требования техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информатизации; обеспечение надежного функционирования средств ИКТ;

Календарно-тематический план

№	Дата	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
1. Модели и алгоритмы (1 час)					
1			<p>Этапы решения задач с использованием ЭВМ. Понятие модели. Модели материальные и информационные. Основные этапы построения математических моделей.</p> <p>Понятие об алгоритме. Свойства алгоритма. Исполнители алгоритмов.</p> <p>Способы описания алгоритмов. Базовые алгоритмические структуры. Свойства базовых структур алгоритмов.</p> <p>Метод пошаговой детализации. Разработка алгоритма сверху - вниз.</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие модели; • понятие алгоритма; • базовые алгоритмические структуры; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • виды моделей; • виды алгоритмических структур; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • этапы решения задач с использованием ЭВМ; • этапы построения математических моделей; • классификацию моделей; • свойства алгоритмов; • свойства базовых алгоритмических структур; • метод пошаговой детализации; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • материальных и невещественных моделей; • бытовых и вычислительных алгоритмов; • способов описания алгоритмов; • исполнителей алгоритмов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • строить математические модели для решения вычислительных задач; • описывать словесно алгоритмы решения задач; • составлять блок-схемы простых алгоритмов; <p><i>использует:</i></p> <ul style="list-style-type: none"> • блок-схемы для описания алгоритмов решение задач; • метод пошаговой детализации при разработке алгоритмов. 	
2. Язык и среда программирования (1 час)					
2			<p>Понятие программы. Языки программирования, их классификация.</p> <p>Характеристика языка и среды программирования. Структура программы на языке программирования.</p> <p><i>Практическая работа № 1.</i> "Работа в среде программирования. Запуск программ на выполнение".</p>	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие программы; • основные элементы языка программирования; • структуру программы языком программирования; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • понятие языка и среды программирования; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • классификацию языка программирования; • структуру программы; <p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • языка программирования различных 	
№	Дата	Корр	Раздел, тема	Требования к уровню подго-	Приме

			Товки учащихся	чание
			<p>видов;</p> <p>умеет: управлять средой программирования для разработки и отладки программ;</p> <p>использует:</p> <ul style="list-style-type: none"> • среду программирования для: загрузки, редактирования, отладки, выполнения, и сохранения программ. 	
3. Линейные алгоритмы (1 час)				
3		<p>Постоянные и переменные величины. Типы и идентификаторы величин.</p> <p>Операция вывода данных.</p> <p>Операция данных.</p> <p>Указание присвоения. Стандартные математические операторы.</p> <p>Математические функции. Приоритет операций.</p> <p>Запись математических выражений.</p> <p>Разработка линейных программ.</p> <p>Практическая работа № 2.</p> <p>"Создание и реализация линейных программ".</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • правила записи идентификаторов; • правила записи математических выражений; • приоритет операций; <p>распознает:</p> <ul style="list-style-type: none"> • постоянные и переменные величины; • типы величин; <p>объясняет:</p> <ul style="list-style-type: none"> • способы предоставления значений переменной; • особенности операции вывода данных; • особенности операции ввода данных; • особенности указания присвоения; <p>приводит примеры:</p> <ul style="list-style-type: none"> • математических операторов; • математических функций; • величин и их типов; <p>умеет:</p> <ul style="list-style-type: none"> • корректно выбирать типы переменных величин; • выбирать идентификаторы для переменных величин; • корректно записывать математические выражения; • разрабатывать и проверять линейные программы; <p>использует:</p> <ul style="list-style-type: none"> • операции ввода, вывода, и присвоения при разработке линейных программ. 	
4. Алгоритмы с разветвлением (1 часа)				
4		<p>Команда разветвления. Условный оператор.</p> <p>Полная и сокращенная форма условного оператора.</p> <p>Логические операции.</p> <p>Простое и составное условие. Логические функции.</p> <p>Разработка и выполнение программы с условным оператором.</p> <p>Оператор множественного выбора. Структура оператора.</p> <p>Разработка и выполнение про-</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • формат условного оператора; • формат оператора множественного выбора; <p>распознает:</p> <ul style="list-style-type: none"> • полную и сокращенную формы условного оператора; • задачи на применение условного оператора; • задачи на применение оператора множественного выбора; <p>объясняет:</p> <ul style="list-style-type: none"> • особенности выполнения полной и сокращенной формы условного оператора; <p>Требования к уровню подготовки учащихся</p>	
№	Дата	Корр	Раздел, тема	Приме

				чение	
			<p>граммы с оператором множественного выбора.</p> <p>Реализация программы с оператором множественного выбора.</p> <p>Практическая работа № 3.</p> <p>"Разработка и выполнение программы с условным оператором и оператором множественного выбора».</p>	<ul style="list-style-type: none"> • особенности выполнения оператора множественного выбора; приводит примеры: • задач на применение условного оператора; • задач на применение оператора множественного выбора; умеет: • применять условный оператор для решения задач; • корректно выбирать форму условного оператора; • корректно выбирать условие в команде разветвления; • решать и проверять задачи на применение условного оператора; • решать и проверять задачи на применение оператора множественного выбора; использует: • условный оператор и оператор множественного выбора при решении задач. 	
<p>5. Циклические алгоритмы (2 часа)</p>					
5			<p>Указание повторения. Типы циклов. Циклы в языке программирования.</p> <p>Оператор цикла с предусловием.</p> <p>Разработка и выполнение циклических программ.</p> <p>Оператор цикла постусловием.</p> <p>Реализация циклических программ.</p> <p>Практическая работа № 4.</p> <p>"Реализация циклических программ".</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • формат операторов цикла с предусловием, с постусловием, с управляющей переменной; <p>распознает:</p> <ul style="list-style-type: none"> • команды повторения разных типов; • задачи на применение циклов; <p>объясняет:</p> <ul style="list-style-type: none"> • особенности выполнения циклов разных типов; • особенности выполнения вложенных циклов; <p>приводит примеры:</p> <ul style="list-style-type: none"> • задач на применение циклов; <p>умеет:</p> <ul style="list-style-type: none"> • применять операторы цикла для решения задач; 	
6			<p>Оператор цикла управляющей переменной.</p> <p>Реализация циклических программ.</p> <p>Вложенные циклы.</p> <p>Использование вложенных циклов в программах.</p> <p>Реализация программы с вложенными циклами.</p> <p>Практическая работа № 5. "Использование вложенных циклов в программах".</p>	<ul style="list-style-type: none"> • применять операторы цикла для решения задач; • корректно выбирать тип цикла; • корректно выбирать условие в цикле; • решать и проверять задачи на применение циклов; • решать и проверять задачи на применение вложенных циклов; использует: • операторы цикла при решении задач. 	

№	Дата	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
6. Обработка табличных величин (2 часа)					
7			Табличные величины. Одномерные таблицы. Структура описания табличных величин. Правила обращения к элементам таблицы. Ввод и вывод элементов массива. Алгоритмы работы с табличными величинами. Нахождение суммы и произведения элементов массива. Практическая работа № 6. "Ввод-вывод данных массива".	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие линейного массива; • структуру описания табличных величин; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • понятие массива, элементы массива индексы и значения элементов массива; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • отличие между простыми и структурированными типами данных; • правила обращения к элементам таблицы; • алгоритмы работы с табличными величинами; • особенности линейного и бинарного поиска в массивах; 	
8			Нахождение количества элементов массива. Разработка и выполнение программ с табличными величинами. Методы поиска. Линейный и бинарный поиск. Практическая работа № 7. "Разработка и выполнение программы с табличными величинами".	<p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • прикладных задач, которые требуют применения массивов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • корректно определять тип и описывать табличные величины; • осуществлять ввод и вывод табличных величин; • находить сумму и произведение элементов массива; • осуществлять линейный и бинарный поиск в массивах; • решать задачи на обработку табличных величин. 	
7. Линейные и многомерные массивы (3 часа)					
9			Нахождение наибольшего (наименьшего) элемента массива. Основные методы упорядочения массивов: прямого выбора, вставка, обмен.	<p><i>Ученик описывает:</i></p> <ul style="list-style-type: none"> • понятие линейного массива; • структуру описания табличных величин; <p><i>распознает:</i></p> <ul style="list-style-type: none"> • понятие массива, элементы массива индексы и значения элементов массива; <p><i>объясняет:</i></p> <ul style="list-style-type: none"> • отличие между простыми и структурированными типами данных; 	
10			Многомерные массивы. Двухмерный массив.	<p><i>приводит примеры:</i></p> <ul style="list-style-type: none"> • прикладных задач, которые требуют применения массивов; <p><i>умеет:</i></p> <ul style="list-style-type: none"> • корректно определять тип и описывать табличные величины; • находить сумму и произведение элементов массива; • осуществлять линейный и бинарный поиск в массивах; 	
11			Ввод и вывод элементов двумерного массива. Практическая работа № 8. "Разработка и выполнение программ с массивами чисел".		

№	Дата	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
8. Методы сортировки в массиве (2 часа)					
12			Основные методы формирования: метод "пузырька", прямого выбора вставки, обмена.	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • основные методы формирования массива: "пузырёк", прямого выбора, вставки, обмена; <p>приводит примеры:</p> <ul style="list-style-type: none"> • разных методов формирования сортировки массивов; • задач на применение сортировки линейных и двумерных массивов; <p>умеет:</p> <ul style="list-style-type: none"> • реализовывать алгоритмы сортировки элементов массива; <p>использует:</p> <ul style="list-style-type: none"> • алгоритмы сортировки элементов линейного и многомерного массивов при решении задач 	
13			Поиск и сортировка в одномерном и двумерном массиве. Практическая работа №9. "Разработка и выполнение программ с сортировкой массива".		
9. Обработка строчных величин (2 часа)					
14			Строковые величины. Структура описания величин строчного типа. Операции над строчными величинами. Стандартные функции для работы с величинами строчного типа. Вставка и исключение элементов строчных величин.	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • понятие строковой величины; • структуру описания величины строкового типа; • стандартные функции для работы с величинами строкового типа; <p>объясняет:</p> <ul style="list-style-type: none"> • особенности выполнения операций со строковыми величинами; • алгоритмы поиска в строке; • специфику превращения строковых и числовых величин; <p>приводит примеры:</p> <ul style="list-style-type: none"> • задач на применение строковых величин; <p>умеет:</p> <ul style="list-style-type: none"> • использовать стандартные функции для работы с величинами строкового типа; • осуществлять поиск в строке; • выполнять превращение строковых и числовых величин; • решать задачи на обработку строковых величин. 	
15			Поиск в строке. Преобразование строчных и числовых величин. Практическая работа № 10. "Разработка и выполнение программы со строчными величинами"		
10. Вспомогательные алгоритмы (2 часа)					
16			Понятие и виды вспомогательных алгоритмов. Правила описания и обращения к вспомогательным алгоритмам в языке программирования. Локальные и глобальные переменные. Передача параметров в подпрограмму. Фактические и формальные параметры.	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • понятие и виды вспомогательных алгоритмов; • понятия локальных и глобальных переменных; • понятия формальных и фактических параметров; • понятие рекурсии; <p>распознает:</p> <ul style="list-style-type: none"> • локальные и глобальные переменные в программах; • формальные и фактические параметры; 	

№	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
17		<p>Понятие рекурсии и её использование для решения задач.</p> <p>Практическая работа № 11. "Разработка и выполнение программы со вспомогательными алгоритмами".</p>	<p>объясняет:</p> <ul style="list-style-type: none"> • правила описания вспомогательных алгоритмов на языке программирования; • методы передачи параметров в подпрограмму; • особенности описания и выполнения рекурсивных функций; <p>умеет:</p> <ul style="list-style-type: none"> • корректно выделять в задачах локальные и глобальные переменные; • описывать вспомогательные алгоритмы; • описывать рекурсивные функции и обращаться к ним; <p>использует:</p> <ul style="list-style-type: none"> • подпрограммы при решении задач; • рекурсивные функции при решении задач. 	
11. Средства работы с файлами (2 часа)				
18		<p>Понятие файла. Типы файлов</p> <p>Работа с файловыми структурами данных средствами языка программирования.</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • понятия файла, типы файлов; • стандартные функции для работы с файлами; <p>объясняет:</p>	
19		<p>Особенности работы с текстовыми файлами.</p> <p>Разработка программ с применением файловых структур данных.</p>	<ul style="list-style-type: none"> • особенности работы с текстовыми файлами; • правила использования текстовых файлов для ввода и вывода данных; • преимущества использования файлов для ввода и вывода данных при отладке программы; <p>умеет:</p> <ul style="list-style-type: none"> • применять текстовые файлы для ввода и вывода данных; • решать и проверять задачи с введением и выводением данных во внешние файлы; <p>использует: текстовые файлы для ввода и вывода данных.</p>	
12. Анализ и построение алгоритмов (15 часов)				
12.1 Методика построения и оценка эффективности алгоритмов (2 часа)				
20		<p>Выбор метода решения задачи.</p> <p>Анализ эффективности алгоритма решения задачи.</p> <p>Классификация алгоритмов. Поиск оптимального алгоритма решения.</p>	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • классификацию алгоритмов; <p>распознает:</p> <ul style="list-style-type: none"> • принадлежность алгоритма к определенному виду; <p>объясняет:</p> <ul style="list-style-type: none"> • этапы разработки алгоритма; • понятие эффективности и результативности алгоритма; 	

№	Дата	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
21			Пошаговая детализация, планирование, и представление алгоритма. Вспомогательные задачи. Обобщение и анализ экстремальных ситуаций. Подборка системы тестов.	<p>методы оценивания эффективности алгоритма;</p> <p>приводит примеры:</p> <ul style="list-style-type: none"> • алгоритмов определенных видов; • методов оценивания эффективности алгоритма. <p>умеет:</p> <ul style="list-style-type: none"> • анализировать эффективность алгоритмов; • осуществлять пошаговую детализацию алгоритмов; • анализировать экстремальные ситуации; • выбирать систему тестов для анализа решения задачи; <p>использует:</p> <ul style="list-style-type: none"> • методы оценки эффективности алгоритма при решении задач. 	
12.2 Элементы вычислительной геометрии (3 часа)					
22			Основные формулы аналитической геометрии. Определение длины отрезка в N-мерном пространстве. Расстояние от точки до прямой. Координаты точек пересечения отрезков и прямых. Определение положения точки относительно многоугольника.	<p>Ученик описывает:</p> <ul style="list-style-type: none"> • основные формулы вычислительной геометрии; • основные формулы векторной геометрии; <p>объясняет:</p> <ul style="list-style-type: none"> • метод определения положения точки относительно многоугольника; • методы нахождения площади многоугольника: метод триангуляции, метод трапеций; 	
23			Нахождение площади многоугольника: метод триангуляции, метод трапеций. Практическая работа № 12. "Разработка и выполнение программ с элементами вычислительной геометрии".	<ul style="list-style-type: none"> • методы проверки принадлежности точек прямой; • методы нахождения порядка обхода вершин выпуклого многоугольника; • методы определения координат вершин выпуклого многоугольника; <p>приводит примеры:</p> <ul style="list-style-type: none"> • формул из вычислительной и векторной геометрии; 	
24			Векторная геометрия. Проверка принадлежности точек прямой. Векторное произведение. Направление поворота. Определение порядка обхода вершин выпуклого многоугольника. Задачи минимизации в геометрической интерпретации. Практическая работа № 13. "Разработка и выполнение программы с элементами векторной геометрии".	<p>умеет:</p> <ul style="list-style-type: none"> • применять формулы из вычислительной и векторной геометрии для решения задач; • реализовывать методы нахождения площади многоугольника; • реализовывать методы проверки принадлежности точек прямой; • реализовывать методы нахождения порядка обхода вершин выпуклого многоугольника; • реализовывать методы определения координат вершин выпуклого Многоугольника 	

№	Дата	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
				<ul style="list-style-type: none"> решать и проверять задачи минимизации в геометрической интерпретации; использует: <ul style="list-style-type: none"> формулы и методы из вычислительной и векторной геометрии для решения задач. 	
12.3 Алгоритмы на графах (5 часов)					
25			Основные понятия теории графов. Матричное представление графов. Матрица связи, матрица инцидентности и матрица расстояний на графе.	Ученик описывает: <ul style="list-style-type: none"> основные понятия теории графов; распознает: <ul style="list-style-type: none"> основные элементы графа; матрицу связности, инцидентности, и расстояний графа; объясняет: <ul style="list-style-type: none"> основные способы представления графов; <ul style="list-style-type: none"> алгоритмы поиска в ширину и глубину, построения остового дерева минимальной длины, определения кратчайшего пути в графе проверки связности по графу нахождения наименьшего общего кратного; приводит примеры: <ul style="list-style-type: none"> графов и соответствующих им матрицы связности, инцидентности и расстояний; задач на применение графов; умеет: <ul style="list-style-type: none"> строить соответствующие графу матрицы связности, инцидентности и расстояний; реализовывать алгоритмы поиска в ширину и в глубину, построения центрального дерева минимальной длины, определения кратчайшего пути в графе, проверки связности к графу, нахождения наименьшего общего кратного; решать и проверять задачи с использованием представления данных в виде графов; использует: <ul style="list-style-type: none"> представление данных в виде графов для решения задач. 	
26			Поиск в ширину Поиск в глубину.		
27			Проверка связности графов Поиск кратчайших путей и оптимальных маршрутов в графах. Алгоритм «Дейкстри».		
28			Нахождение минимального центрального дерева графа по алгоритму «Прима-Краскала». Алгоритм «Тарьяна» нахождения наименьшего общего кратного.		
29			Практическая работа № 14. "Разработка и выполнение программы с элементами теории графов".		
12.4 "Жадные" алгоритмы (2 часа)					
30			Понятие "жадного" алгоритма. Теоретические основы "жадных" алгоритмов. Преимущества и недостатки "жадных" алгоритмов. Классические примеры "жадных" алгоритмов.	Ученик описывает: <ul style="list-style-type: none"> сущность "жадных" алгоритмов; объясняет: <ul style="list-style-type: none"> основные принципы построения "жадных" алгоритмов; критерии применения "жадных" алгоритмов; преимущества и недостатки "жадных" алгоритмов; <ul style="list-style-type: none"> идеи решения классических задач на применение "жадных" алгоритмов; приводит примеры:	
31			Решение задач с применением "жадных" алгоритмов. Геометрические, транспортные, экономические задачи.		

№	Дата	Корр	Раздел, тема	Требования к уровню подготовки учащихся	Примечание
			<i>Практическая работа № 15.</i> "Решение задач с помощью "жадных" алгоритмов".	<ul style="list-style-type: none"> • задач на применение "жадных" алгоритмов; умеет: <ul style="list-style-type: none"> • оценивать возможности применения "жадных" алгоритмов к Решению задач • реализовывать "жадные" алгоритмы; • решать и проверять задачи с применением "жадных" алгоритмов; использует: <ul style="list-style-type: none"> • "жадные" алгоритмы для решения задач. 	
12.5 Динамическое программирование (3 часа)					
32			Основные понятия задач динамического программирования. Критерии применения метода динамического программирования. Решение задач методом динамического программирования. Матричное счисление. Перемножение нескольких матриц.	Ученик описывает: <ul style="list-style-type: none"> • основные понятия задач динамического программирования; объясняет: <ul style="list-style-type: none"> • критерии применения Динамического программирования; • этапы построения алгоритмов основанных на принципах динамического программирования; 	
33			Нахождение наибольшей общей подпоследовательности множества. Определение оптимальной триангуляции многоугольника.	<ul style="list-style-type: none"> • понятие независимых подзадач и подзадач, которые перекрываются; • разницу в использовании динамического программирования и "жадных" алгоритмов; • разницу в использовании полнопреодолимого метода, метода "разделяй и властвуй" и динамического программирования; приводит примеры: <ul style="list-style-type: none"> • задач на применение динамического программирования; 	
34			<i>Практическая работа № 16.</i> "Решение задач методом динамического программирования".	умеет: <ul style="list-style-type: none"> • оценивать эффективность алгоритмов динамического программирования; • разбивать задачу на шаги строить рекуррентное соотношение между параметрами подзадач • составлять и реализовывать алгоритмы классических задач динамического программирования; • решать задачи методом динамического программирования. использует: <ul style="list-style-type: none"> • динамическое программирование для решения задач. 	

Перечень учебно-методической литературы

1. Окулов с.М., Программирование в алгоритмах. - БИНОМ, 2002.
2. Кнут д.З. Искусство программирования, том 1. Основные алгоритмы, 3-е изд.: Пер. с англ.: Уч. пос. - М.: Издательский дом "Вильямс", 2000.
3. Кристофидес Н. Теория графов. Алгоритмический подход. -М.: Мир, 1978.
4. Кормен Т., Лейзерсон Ч., Ривест Г. Алгоритмы: построение и анализ. - М: МЦНМО, 2000

5. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы.: Пер. с англ.: Уч. пос. — М.: Издательский дом “Вильямс”, 2000. — 384 с., ил.
6. Вентцель Е.С. Исследование операций: задачи, принципы, методология. — 2-е изд., стер. — М.: Наука. Гл. ред. физ.-мат. лит., 1988. — 208 с. — (Пробл. науки и техн. прогресса)
7. Вирт Н. Алгоритмы и структуры данных: Пер. с англ. — М.: Мир, 1989. — 360 с., ил.
8. Липский В. Комбинаторика для программистов: Пер. с польск. — М.: Мир, 1988. — 213 с., ил.
9. Кормен Т., Лейзерсон Ч., Ривест Р. Алгоритмы: построение и анализ. М.: МЦНМО, 2001. — 960 с., 263 ил.