

**Муниципальное бюджетное учреждение
дополнительного образования
«ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ»**

Рассмотрена и рекомендована к
утверждению Методическим
советом МБУ ДО «Центр
дополнительного образования»
Протокол от 28.08.2023г. №1

Утверждена
Директором МБУ ДО «Центр
дополнительного образования»
Г.Ф.Войтошенко
Приказ от 28.08.2023г. № 75



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
технической направленности
«ШКОЛА ПРОГРАММИРОВАНИЯ»**

Возраст обучающихся 9-17 лет
Срок реализации программы 4 года
Количество модулей - 4

Разработчик:

Котова Юлия Николаевна,
педагог дополнительного образования

Каменск-Уральский
2023

Пояснительная записка

Направленность: дополнительная общеобразовательная общеразвивающая программа «Школа программирования» предназначена для работы в системе дополнительного образования и имеет **техническую** направленность.

Программа «Школа программирования» разработана в соответствии с

- Федеральным законом РФ от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;

- Постановлением Главного государственного санитарного врача РФ от 28 сентября 2020 г. № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

- Приказом Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

- Приказом Министерства просвещения Российской Федерации от 30 сентября 2020 г. № 533 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам, утвержденный приказом Министерства просвещения Российской Федерации 09.11.2018 г. №196»;

- Письмом Минобрнауки России № 09-3242 от 18.11.2015 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»);

- Приказом Министерства образования и молодежной политики Свердловской области от 30.03.2018 №162-Д «Об утверждении Концепции развития образования на территории Свердловской области на период до 2035 года».

- Уставом и нормативными документами МБУ ДО «Центр дополнительного образования»

Общая характеристика программы

В настоящее время какую бы сферу жизнедеятельности человека мы не взяли: медицину, проектирование зданий, машин, образование и любую другую – без применения компьютерных технологий нигде в современном мире не обходится. Для каждой из этих областей разрабатываются соответствующие IT-программы. Следовательно, сегодня является чрезвычайно востребованной такая профессия, как программист: навыки программирования пользуются высоким спросом, должность программиста хорошо оплачивается. Даже за пределами IT-мира знание хотя бы одного языка программирования – это серьезный плюс в резюме. Наш город нуждается в хороших программистах и очень важно мотивировать учащихся на получение необходимых нашему городу IT-профессий.

Программа «Школа программирования» направлена в первую очередь на то, чтобы развить у учащихся алгоритмическое мышление и научить использовать стандартные и нестандартные алгоритмы для решения задач повышенного уровня сложности.

Данная программа предполагает углубленное изучение языков программирования и подготовку учащихся к турнирам и конкурсам по программированию различного уровня.

По форме организации образовательного процесса программа является модульной и предполагает сроки освоения от 3 до 4 лет в зависимости от возраста, в котором ребенок начинает обучение. Модули ориентированы на различные возрастные группы и предусматривают обучение с усложнением содержания материала по принципу восходящей и расширяющейся дидактической спирали.

По уровню освоения программа общеразвивающая, разноуровневая (стартовый, базовый, продвинутый уровни). Обеспечивает возможность обучения детей с различным стартовым уровнем готовности к её освоению, в том числе, одарённых детей.

Программа состоит из 4 модулей.

Модуль	Уровень	Возраст учащихся	3 года обучения	4 года обучения
I. «Алгоритмы и исполнители»	Стартовый	9-10 лет (3-4 класс)	–	+
II. «Школа юного программиста»	Базовый	11-13 лет (5-7 класс)	+	+
III. «Школа олимпиадного программирования»	Продвинутый	13-15 лет (7-9 класс)	+	+
IV. «Спортивное программирование»	Продвинутый	15-17 лет (9-11 класс)	+	+

Особенности целевой группы

I модуль (стартовый уровень) «Алгоритмы и исполнители» предназначен для мальчиков и девочек 9-10 лет (3-4 класса), имеющих хорошие математические способности и склонность к логическому мышлению. Изучение психолого-педагогической литературы дало основание сделать вывод, что, хотя проблема организации формирования и развития логико-алгоритмического мышления в педагогической и психологической теории до сих пор не нашла единого решения, практически все исследователи единодушны в том, что работа по формированию и развитию логико-алгоритмического мышления детей младшего школьного возраста очень эффективна. При этом сама система заданий должна учитывать специфику восприятия и мышления детей младшего школьного возраста. Программа модуля позволяет в ходе логических игр и построения алгоритмов для различных

исполнителей познакомить детей с различными алгоритмическими структурами, выявить обучающихся, имеющих нестандартное мышление и способности логически мыслить. Группа обучающихся не более 14 человек.

II модуль (базовый уровень) «Школа юного программиста» направлен на обучение детей 11-13 лет (5-7 класс), имеющих способности в области математики, имеющих хорошее логическое мышление или учащихся, которые успешно завершили I модуль данной программы и показали умение логически мыслить. В этом возрасте учащиеся могут освоить несложный язык программирования. Модуль дает учащимся возможность познакомиться с основами программирования на основе изучения языка программирования Питон. Данный язык программирования имеет ряд преимуществ. Он минималистичен и имеет легкий синтаксис языка. При обучении языку программирования Питон можно познакомить учащихся со всеми основными алгоритмическими структурами. Группа обучающихся не более 16 человек.

III модуль (Продвинутый уровень) «Школа олимпиадного программирования» предполагает изучение основных алгоритмов олимпиадного программирования. Модуль рассчитан на детей 13-15 лет, которые успешно завершили II модуль данного курса и имеющих склонность к программированию. Так же данный модуль могут выбрать учащиеся 13-15 лет, не обучавшиеся ранее по данной программе, но участвовавшие в турнирах и олимпиадах по программированию и желающие повысить свой уровень программирования. Группа обучающихся не более 14 человек.

IV модуль (Продвинутый уровень) «Спортивное программирование» дает возможность учащимся освоить сложные алгоритмы, позволяющие решать задачи олимпиадного уровня. Модуль рассчитан на учащихся 9-11 классов. Учащиеся 8 классов, успешно освоившие предыдущие II и III модули так же могут выбрать обучение на этом уровне. Так как ежегодно растет уровень олимпиадных заданий, появляются новые технологии и способы решений задач, то для более успешной подготовки к олимпиадам и турнирам по программированию данный модуль по желанию учащиеся могут пройти несколько раз, каждый раз повышая свое мастерство и уровень своего участия в мероприятиях различного масштаба. Группа обучающихся не более 14 человек.

Актуальность:

В настоящее время в нашем городе Каменске-Уральском нет курсов, обучающих спортивному (олимпиадному) программированию. Однако учащихся, способных к программированию есть не мало. Уровень заданий, предлагаемых на всевозможных турнирах, конкурсах и олимпиадах по программированию, заметно выше того, что изучают учащиеся массовых школ на уроках. При подготовке к турнирам необходимо не только знакомить с методами олимпиадного программирования, но и учить детей правильно воспринимать задания

нестандартного характера повышенной трудности и преодолевать психологическую нагрузку при работе в незнакомой обстановке. И чем раньше начать такую работу, тем это будет эффективнее. Участвуя в различных олимпиадах и турнирах, дети проявляют в полной мере свои способности, удовлетворяют здоровые амбиции, развивают самостоятельность в получении новых знаний и умений находить верное решение.

Изучение основ программирования связано с развитием умений и навыков, которые носят общеинтеллектуальный характер. Изучение языков программирования развивает мышление, способствует формированию у учащихся многих приемов умственной деятельности, таких как многоаспектный анализ, планомерность, организованность деятельности, умение логически грамотно рассуждать, четко формулировать свои мысли. Эти качества необходимы для подготовки хороших специалистов IT-технологий, в которых в настоящее время нуждаются предприятия нашего города и региона в целом.

Изучая технологии программирования, обучающиеся усваивают основы алгоритмизации, приобщаются к алгоритмической культуре, познают азы профессии программиста.

Данный курс актуален, т.к. не только формирует общеучебные и общеинтеллектуальные умения и навыки, но и позволяет подготовить одаренных учащихся для успешного выступления на различных олимпиадах и турнирах по программированию. Так же данный курс можно считать профориентационным, так как он формирует у учащихся интерес к профессиям, связанных с программированием.

Формы и методы обучения:

При изучении данного курса особый акцент делается не только на приобретение новых знаний, но и на развитие способностей приобретать знания самостоятельно, на умение анализировать ситуацию, выбирать самый удобный способ составления программ для решения задач, аргументировано отстаивать свою точку зрения.

Эффективность работы обеспечивается сочетанием фронтальной, групповой и индивидуальной форм работы на занятиях, а также правильно организованной самостоятельной работой.

Ведущие виды занятий - практические занятия, лекции.

Формы подведения результатов – практические работы, самостоятельные работы, миниолимпиады, тестирование, творческие работы, контроль он-лайн (на сайтах для решения задач онлайн, результаты участия в различных конкурсах, турнирах и олимпиадах городского, регионального и федерального уровня).

Режим занятий

I-й модуль – 1 раз в неделю по два учебных часу (2 занятия по 45 минут с перерывом между уроками 10 минут). Всего 72 учебных часа в год.

II-й модуль – 1 раз в неделю по два учебных часа (2 занятия по 45 минут с перерывом между уроками 10 минут). Всего 72 учебных часа в год.

III-й модуль – 1 раз в неделю по два учебных часа (2 занятия по 45 минут с перерывом между уроками 10 минут). Всего 72 учебных часа в год.

IV-й модуль – 1 раз в неделю по два учебных часа (2 занятия по 45 минут с перерывом между уроками 10 минут). Всего 72 учебных часа в год.

Общее количество - 288 учебных часов.

Цель и задачи общеразвивающей программы

Цель: Создать условия для развития у детей интереса к алгоритмизации и программированию, подготовить учащихся к участию в олимпиадах и турнирах различного уровня по программированию, формирование творческой личности, владеющей знаниями, умениями и навыками в области программирования.

Задачи:

Обучающие:

- обучение основам алгоритмизации, знакомство с основными алгоритмическими структурами;
- знакомство с несколькими языками программирования;
- обучение основным алгоритмическим структурам программирования;
- знакомство учащихся с алгоритмами повышенной сложности, позволяющим решать задачи олимпиадного уровня сложности;
- обучение учащихся принципам спортивного программирования;
- создание интереса к информационным технологиям и программированию.

Развивающие:

- воспитание всесторонне развитой личности;
- развитие у учащихся логического мышления;
- развитие умения применять стандартные алгоритмы к нестандартным задачам;
- развитие находчивости, изобретательности, умения довести решение задачи до конца;
- развитие личностных свойств и качеств: самостоятельности, саморегуляции, саморганизации, аккуратности, терпения, настойчивости в достижении цели;
- формирование интереса к интеллектуальной, научной деятельности, программированию.

Воспитательные:

- воспитание целеустремленности и трудолюбия;
- формирование навыков здоровьесбережения при работе на ПК (правильная осанка, охрана зрения, самоконтроль времени работы за ПК);
- формирование активной жизненной позиции, коммуникативных компетенций, адекватной самооценки, умения анализировать свою и общую деятельность через участие учащихся в турнирах и конкурсах;
- воспитание коммуникативного общения в группе, мотивированное на достижение высокой результативности.

**Учебно-тематический план.
1 модуль. «Алгоритмы и исполнители»**

№ п/п	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		Всего	Теория	Практика	
1	Раздел 1. Введение. Правила техники безопасности. Входное тестирование.	2	1	1	
2	Раздел 2. Понятие алгоритма и исполнителя.	4	2	2	
2.1	Что такое алгоритм? Словесные алгоритмы.	2	1	1	
2.2	Какими бывают исполнители. Я – исполнитель.	2	1	1	Проект «Оригами»
3	Раздел 3. Алгоритмы и исполнители пакета «Роботландия».	6	3	3	
3.1	Алгоритмические этюды.	2	1	1	
3.2	Исполнители.	2	1	1	Практическая работа
3.3	Черные ящики: Буквоед.	2	1	1	
4	Раздел 4. Алгоритмический язык «Кукарача».	12	5	7	
4.1	Знакомство с «Кукарачей». Поля. Система команд. Первая программа. Редактирование текста программ.	2	1	1	
4.2	Линейные программы.	2	1	1	Практическая работа
4.3	Подпрограммы.	2	1	1	
4.4	Циклы.	2	1	1	Практическая работа
4.5	Команда «Если».	2	1	1	
4.6	Мини-олимпиада «Кукарача».	2		2	Мини- олимпиада
5	Раздел 5. Исполнитель: Транспортер.	8	4	4	
5.1	Знакомство с ПО «Мир информатики».	2	1	1	
5.2	Исполнитель Транспортер. Система команд. Редактирование программы. Линейные программы.	2	1	1	
5.3	Ветвления.	2	1	1	Практическая работа
5.4	Циклические программы.	2	1	1	Практическая работа

6	Раздел 6. Некоторые математические понятия.	4	2	2	
6.1	Система координат. Точка. Отрезок. Координатный диктант.	2	1	1	Практическая работа
6.2	Угол. Значение угла в градусах. Использование транспортира.	2	1	1	
7	Раздел 7. studio.code.org - Студия программирования. Курс 2.	18	7	11	
7.1	Последовательности.	4	2	2	
7.2	Циклы.	4	2	2	Практическая работа
7.3	Отладка текста программ.	4	2	2	Практическая работа
7.4	Ветвления.	2	1	1	
7.5	Лаборатория игр: Создай историю.	2	1	1	Практическая работа
7.6	Творческая работа по теме «Студия программирования. Курс 2».	2	1	1	Творческая работа
8	Раздел 8. Логические задачи.	4	2	2	
8.1	Решение логических задач с помощью таблиц.	2	1	1	
8.2	Задача Энштейна. Доски Шерлока.	2	1	1	Практическая работа
9	Раздел 9. studio.code.org Час кода с Minecraft.	4	1	3	
9.1	Minecraft: водное приключение.	2	1	1	
9.2	Minecraft: водное приключение.	2		2	Творческая работа
10	Раздел 10. studio.code.org Час кода с Анной и Эльзой.	2		2	Творческая работа
11	Раздел 11. Итоговый контроль.	6	2	4	
11.1	Подготовка к мини-олимпиаде	2	2		
11.2	Итоговая мини-олимпиада	2		2	Мини-олимпиада
11.3	Итоговое тестирование.	2		2	Тест Равена Тест «Точки» Тест «Числовые ряды»
12	Резервное время учителя.	2	2		
	Итого часов	72	31	41	

2 модуль. Школа юного программиста.

№ п/п	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		Всего	Теория	Практика	
1	Раздел 1. Введение. Техника безопасности.	2	1	1	
2	Раздел 2. Моделирование и формализация. Основы алгоритмизации. Математические модели.	4	2	2	
2.1	Постановка задачи. Формализация. Языки программирования.	2	1	1	Самоконтроль
2.2	Виды моделей. Математические модели.	2	1	1	Практическая работа
3	Раздел 3. Начала программирования. Python 3.	10	4	6	
3.1	Структура программы. Операторы присваивания и вывода данных.	2	2		
3.2	Типы данных. Переменные. Ввод данных с клавиатуры.	2	1	1	Самоконтроль
3.3	Целочисленная арифметика.	2	1	1	
3.4	Линейный алгоритм.	2		2	Практическая работа
3.5	Мини-олимпиада «Линейный алгоритм».	2		2	Мини-олимпиада
4	Раздел 4. Основные алгоритмические структуры на языке Python.	32	16	16	
4.1	Условная инструкция if	2	1	1	
4.2	Полный синтаксис условного оператора	2	1	1	
4.3	Тип данных bool, логические операции	2	1	1	Практическая работа
4.4	Вложенные условные инструкции	2	1	1	Самоконтроль
4.5	Виды циклов. Цикл for.	2	2		
4.6	Функция range	2	1	1	
4.7	Варианты применения цикла for	2	1	1	Практическая работа
4.8	Цикл с предусловием.	2	1	1	Практическая работа
4.9	Вложенные циклы.	2	1	1	
4.10	Строки.	2	1	1	
4.11	Методы и функции работы со	2	1	1	Практическая

	строками.				работа
4.12	Разбор задач повышенной сложности.	2	2		
4.13	Списки. Работа с элементами списков.	2	1	1	
4.14	Методы и функции для работы со списками.	2	1	1	
4.15	Линейный поиск в списке.	2	1	1	
4.16	Мини-олимпиада «Начало программирования».	2		2	Мини-олимпиада
5	Раздел 5. Графический модуль.	8	3	5	
5.1	Графические примитивы.	2	1	1	Самоконтроль
5.2	Работа с цветом.	2	1	1	Самоконтроль
5.3	Графика и циклы.	2	1	1	Практическая работа
5.4	Творческая работа «Графические операторы».	2		2	Творческая работа
6	Раздел 6. Системы счисления.	10	5	5	
6.1	Обзор систем счисления.	2	2		
6.2	Арифметические операции в различных системах счисления.	2	1	1	
6.3	Перевод из одной системы счисления в другую.	2	1	1	
6.4	Машинные системы счисления.	2	1	1	Творческая работа
6.5	Программы перевода различных систем счисления.	2		2	Практическая работа
7	Раздел 7. Подготовка к мини-олимпиаде. Мини-олимпиада.	4	1	3	
7.1	Подготовка к мини-олимпиаде.	2	1	1	Самоконтроль
7.2	Мини-олимпиада	2		2	Мини-олимпиада
8	Раздел 8. Резервное время педагога.	2		2	
Итого часов		72	31	41	

3 модуль. Школа олимпиадного программирования.

№ п/п	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		Всего	Теория	Практика	
1	Раздел 1. Введение. Правила техники безопасности.	2	1	1	
2	Раздел 2. Восстановление навыков написания программ для стандартных и нестандартных алгоритмов.	6	3	3	
2.1	Линейный алгоритм. Ветвления.	2	1	1	
2.2	Циклы. Графика.	2	1	1	
2.3	Символьные переменные.	2	1	1	
3	Раздел 3. Списки.	8	4	4	
3.1	Способы ввода и вывода списков. Срезы.	2	1	1	Контроль онлайн
3.2	Операции по спискам.	2	1	1	Контроль онлайн
3.3	Двумерные массивы	2	1	1	Контроль онлайн
3.4	Работа с двумерными массивами.	2	1	1	Контроль онлайн
4	Раздел 4. Структуры данных.	8	4	4	
4.1	Множества.	2	1	1	Контроль онлайн
4.2	Операции с множествами.	2	1	1	Контроль онлайн
4.3	Словари.	2	1	1	Контроль онлайн
4.4	Работа с элементами словаря.	2	1	1	Контроль онлайн
5	Раздел 5. Введение в олимпиадное программирование.	4	2	2	
5.1	Специфика олимпиадных задач.	2	2		
5.2	Решение простых олимпиадных задач. Подбор тестов.	2		2	
6	Раздел 6. Целочисленная арифметика .	4	2	2	
6.1	Делимость. Простые числа. Арифметика остатков.	2	1	1	
6.2	Алгоритм Евклида. НОД, НОК.	2	1	1	Практическая работа
7	Раздел 7. Процедуры и функции.	6	3	3	
7.1	Процедуры с параметром. Локальные и глобальные переменные.	2	1	1	
7.2	Функции. Применение функций для	2	1	1	Практическая

	упрощения текста программы.				работа
7.3	Рекурсии.	2	1	1	
8	Раздел 8. Алгоритмы для обработки массивов. Динамическое программирование.	12	7	5	
8.1	Вспомогательные массивы. Сохранение эффективности алгоритма.	2	1	1	
8.2	Сортировки пузырьком, выбором, перестановками.	2	2		
8.3	Быстрая сортировка.	2	1	1	Контроль он-лайн
8.4	Префиксные суммы.	2	1	1	Самоконтроль
8.5	Последовательность Фибоначчи. Рекуррентные отношения.	2	1	1	
8.6	Динамическое программирование. Задача о кузнечике.	2	1	1	Контроль-онлайн
9	Раздел 9. Миниолимпиада.	2		2	Мини-олимпиада
10	Раздел 10. Вычислительная геометрия.	6	3	3	
10.1	Прямые. Точки. Отрезки. Длина отрезка.	2	1	1	
10.2	Параллельность и ортогональность прямых.	2	1	1	
10.3	Многоугольники.	2	1	1	
11	Раздел 11. Основы логики.	8	4	4	
11.1	Алгебра высказываний. Текстовые задачи.	2	1	1	Самоконтроль
11.2	Логические операции. Таблицы истинности.	2	1	1	Взаимоконтроль
11.3	Решение логических задач.	2	1	1	Творческая работа
11.4	Задача Энштейна. Доски Шерлока.	2	1	1	
12	Раздел 12. Итоговая работа.	2		2	Итоговая работа
	Резервное время педагога.	4		4	
	Итого часов	72	33	39	

4 модуль. Спортивное программирование.

№ п/п	Название раздела, темы	Количество часов			Формы аттестации (контроля)
		Всего	Теория	Практика	
1	Раздел 1. Введение. Правила техники безопасности.	2	1	1	
2	Раздел 2. Знакомство с языком программирования C++.	8	4	4	
2.1	Структура программы. Линейный алгоритм. Ветвления.	2	1	1	Контроль онлайн
2.2	Циклы.	2	1	1	Контроль онлайн
2.3	Строки.	2	1	1	Контроль онлайн
2.4	Массивы.	2	1	1	Контроль онлайн
3	Раздел 3. Процедуры и функции. Нисходящее пошаговое структурное проектирование.	4	2	2	
3.1	Процедуры и функции.	2	1	1	Самоконтроль
3.2	Рекурсии.	2	1	1	
4	Раздел 4. Динамическое программирование.	10	4	6	
4.1	Динамическое программирование. Задачи о лесенке и кузнечике.	2	1	1	
4.2	Последовательности.	2	1	1	Самоконтроль
4.3	Одномерная динамика.	2	1	1	Практическая работа
4.4	Двумерная динамика на таблицах.	2	1	1	Самоконтроль
4.5	Решение задач по теме «Динамическое программирование».	2		2	Контроль онлайн
5	Раздел 5. Перебор с возвратом.	6	3	3	
5.1	Стек. Организация очереди.	2	1	1	Практическая работа
5.2	Использование стека и рекурсии.	4	2	2	Практическая работа
6	Раздел 6. Записи (словари). Повторение.	6	3	3	
6.1	Объединение разнородных элементов. Поля.	2	1	1	Самоконтроль
6.2	Операции над записями.	4	2	2	Творческая работа

7	Раздел 7. Жадные алгоритмы.	6	3	3	
7.1	Принцип жадного алгоритма. Задача о рюкзаке.	2	1	1	Самоконтроль
7.2	Решение задач на жадные алгоритмы.	4	2	2	Взаимоконтроль
8	Раздел 8. Теория графов.	10	6	4	
8.1	Графы и деревья: определения и примеры.	2	2		
8.2	Матрицы смежности.	2	1	1	Самоконтроль
8.3	Алгоритмы на деревьях и графах.	6	3	3	Практическая работа
9	Раздел 9. Длинная арифметика.	4	2	2	
9.1	Запись числа в массив.	2	1	1	Самоконтроль
9.2	Операции с длинными числами.	2	1	1	Взаимоконтроль
10	Раздел 10. Решение задач по информатике.	8	4	4	
10.1	Количество информации.	2	1	1	
10.2	Чтение и анализ кода программы.	2		2	Практическая работа
10.3	Разбор ошибок при решении теоретических заданий.	2	2		
10.4	Комбинаторика.	2	1	1	
11	Раздел 11. Подготовка к выпускной работе. Выпускная работа.	4	1	3	
11.1	Подготовка к выпускной работе.	2	1	1	
11.2	Выпускная работа.	2		2	Творческая работа
	Резервное время учителя.	4		4	
Итого часов		72	32	40	

Содержание материала 1 модуля. «Алгоритмы и исполнители»

1. Введение. Правила техники безопасности. Входное тестирование.

Теория. Правила поведения в компьютерном классе. Введение в курс.

Практика. Тестирование по методикам: Тест Равена. Тест «Точки» Тест «Числовые ряды».

2. Понятие алгоритма и исполнителя.

2.1 Что такое алгоритм? Словесные алгоритмы.

Теория. Знакомство с понятием алгоритма. Простейшие словесные алгоритмы. Алгоритмы в нашей жизни.

Практика. Написание алгоритма рисования картинка на выбор. Творческая командная игра «Алгоритмика». Игра «Мудрый крот».

2.2 Какими бывают исполнители. Я – исполнитель.

Теория. Понятие исполнителя. Каким может быть исполнитель. Система команд исполнителя.

Практика. Действие по готовому алгоритму. Творческая работа «Оригами». Игра «Морской бой».

3. Алгоритмы и исполнители пакета «Роботландия».

3.1 Алгоритмические этюды.

Теория. Понятие алгоритмического этюда. Известные и неизвестные алгоритмические этюды.

Практика. Знакомство с ПО «Роботландия». Составление алгоритмов: Перевозчик. Монах. Конюх. Переливашка.

3.2 Исполнители.

Теория. Исполнители - программы. Система команд исполнителей. Справочная система ПО «Роботландия».

Практика. Исполнители: Плюсик. Машинист. Автомат. Плюсик.

3.3 Черные ящики: Буквоед.

Теория. Черные ящики – понятие. Эксперимент «Разгадай алгоритм». Творческая игра «Ввод и вывод». Загрузка программы Буквоед и правила разгадывания алгоритмов.

Практика. Буквоед.

4. Алгоритмический язык «Кукарача».

4.1 Знакомство с «Кукарачей».

Теория. Загрузка программной среды «Кукарача». Рабочие поля. Система команд. Ввод команд. Первая программа. Имя программы. Редактирование текста программ.

Практика. Написание простых программ для Кукарачи.

4.2 Линейные программы.

Теория. Понятие линейности алгоритма. Возможность использования команды Повтори. Создание и упрощение текста программы. Удаление строк из текста программы.

Практика. Практическая работа: «линейный алгоритм».

4.3 Подпрограммы.

Теория. Вспомогательные программы Правила написания подпрограмм. Использование подпрограмм для уменьшения кода программы и удобства исправления ошибок.

Практика. Решение задач с использованием подпрограмм.

4.4 Циклы.

Теория. Что такое цикл. Циклы в нашей жизни. Циклы для Кукарачи. Организация цикла с помощью подпрограмм.

Практика. Практическая работа на тему «Циклы».

4.5 Команда «Если».

Теория. Условия. Выбор в нашей жизни. Разветвляющиеся алгоритмы.

Практика. Использование команды «Если» для решения сложных задач для Кукарачи.

4.6 Зачет по теме «Кукарача».

Практика. Зачетная работа по теме алгоритмический язык «Кукарача».

5. Исполнитель: Транспортёр.

5.1 Знакомство с ПО «Мир информатики».

Теория. Загрузка ПО «Мир информатики». Разделы. Ознакомление с разделами «Конструирование», «Исполнители».

Практика. Выполнение заданий из разделов «Конструирование», «План и правила», «Логика».

5.2 Исполнитель Транспортёр. Система команд. Редактирование программы. Линейные программы.

Теория. Знакомство с исполнителем Транспортёр: загрузка, система команд.

Практика. Выполнение заданий для транспортёра из пакета 1 года обучения ПО «Мир информатики». Игра «Мудрый крот».

5.3 Ветвления.

Теория. Ветвления. Формы ветвлений. Команда ветвления.

Практика. Творческая игра «Если б я...». Выполнение заданий для транспортёра из пакета 2 года обучения ПО «Мир информатики». Игра «Морской бой».

5.4 Циклические программы.

Теория. Циклы. Возможности цикла.

Практика. Творческая игра «Движение по циклу». Выполнение заданий для транспортёра из пакета 3-4 года обучения ПО «Мир информатики».

6. Некоторые математические понятия.

6.1 Система координат. Точка. Отрезок. Координатный диктант.

Теория. Система координат. Координаты точки. Отрезок. Обход точек.

Практика. Координатный диктант. Игра «Морской бой». Игра «Максит».

6.2 Угол. Значение угла в градусах. Использование транспортира.

Теория. Угол. Виды углов. Сколько градусов в круге. Определение значения угла «на глаз». Определение значения угла с помощью транспортира.

Практика. Практическая работа «Рисунок из углов».

7. studio.code.org - Студия программирования. Курс 2.

7.1 Последовательность.

Теория. Регистрация на сайте. Знакомство с разделами сайта. Система команд в заданиях «Лабиринт». Перетаскивание блоков программы. Составление алгоритма. Запуск. Система команд в заданиях «Художник». Поворот «налево». Поворот «направо».

Практика. Прохождение 5 этапов раздела «Лабиринт: последовательность». Прохождение 8 уровней раздела «Художник: последовательность».

7.2 Циклы.

Теория. Организация цикла в заданиях «Лабиринт», «Художник», «Пчела». Возможность организации вложенных циклов. Возможность включения в цикл нескольких действий.

Практика. Прохождение по 5-8 заданий в каждом исполнителе «Лабиринт», «Художник», «Пчела».

7.3 Отладка текста программ.

Теория. Понятие «Код программы». Тестирование кода. Понятие отладки кода. Обзор комплекса заданий «Пчела:отладка», «Художник:отладка».

Практика. Прохождение по 5-8 заданий в каждом комплексе.

7.4 Ветвления.

Теория. Ветвления. Структура блоков с условием. Организация условия для исполнителя «Пчела».

Практика. Прохождение задания в комплексе «Пчела: условные команды».

7.5 Лаборатория игр: Создай историю.

Теория. Игровые технологии. Как создаются игры.

Практика. Создать собственную игру, используя раздел «Лаборатория игр: Создай историю».

7.6 Творческая работа по теме «Студия программирования. Курс 2» .

Практика. Творческая работа в онлайн курсе studio.code.org.

8. Логические задачи.

8.1 Решение логических задач с помощью таблиц.

Теория. Логика. Суждения. Текстовые задачи. Табличный способ решения логических задач.

Практика. Задания раздела «Логика. Суждения» из ПО «Мир информатики». Решение задач с помощью таблиц.

8.2 Задача Эйнштейна. Доски Шерлока.

Теория. Эйнштейн. Истинность суждений. Загадки Эйнштейна. Знакомство с программой «Шерлок».

Практика. Разгадывание двух «досок» из программы «Шерлок».

9. studio.code.org Час кода с Minecraft.

9.1 Minecraft: водное приключение.

Теория. Известные ИТ-игры. Игромания – болезнь или нет? Могут ли игры быть полезными? Обзор заданий курса «Minecraft: водное приключение».

Практика. Прохождение заданий курса «Minecraft: водное приключение».

9.2 Minecraft: водное приключение.

Практика. Прохождение заданий курса «Minecraft: водное приключение».

10 studio.code.org Час кода с Анной и Эльзой.

Теория. Компьютерная анимация. Мультипликация. Повторение темы «Значения угла в градусах». Обзор заданий курса «Час кода с Анной и Эльзой».

Практика. Прохождение заданий курса «Час кода с Анной и Эльзой».

11 Итоговый контроль.

11.1 Подготовка к мини-олимпиаде.

Теория. Повторение тем «Алгоритмы. Исполнители». Исполнитель «Кукарача». Логика.

Практика. Повторение решения заданий для исполнителей «Машинист», «Кукарача», «Транспортер».

11.2 Итоговая мини-олимпиада.

Практика. Итоговая мини-олимпиада.

11.3 Итоговое тестирование.

Практика. Тестирование по методикам: Тест Равена. Тест «Точки» Тест «Числовые ряды».

12 Резервное время учителя.

Содержание материала 2 модуля. «Школа юного программиста»

1. Введение.

Теория. Правила техники безопасности. Правила поведения в компьютерном классе. Освоение клавиатуры.

Практика. Клавиатурный тренажер.

2. Моделирование и формализация. Основы алгоритмизации. Математические модели.

2.1. Постановка задачи. Формализация. Языки программирования.

Теория. История языков программирования. Хорошо и «плохо» поставленные задачи. Проблема адекватности постановки задачи. Этапы решения задачи.

Практика. Решение задачи «Конюх». Решение задачи «Монах». Логическая игра «Крот».

Тема 2.2. Виды моделей. Математические модели.

Теория. Виды моделей. Информационные модели. Математические модели.

Практика. Практическая работа «Построение математических моделей».

3. Начала программирования. Python 3.

3.1. Структура программы. Операторы присваивания и вывода данных.

Теория. Структура, алфавит, окно и редактор, компиляция и исполнение программы. Арифметические операции.

Оформление простейшей программы.

3.2. Типы данных. Переменные. Ввод данных с клавиатуры.

Теория. Типы данных. Имена переменных. Способы ввода значений переменных.

Практика. Создания простейших программ. Программы-диалоги.

3.3. Целочисленная арифметика.

Теория. Целый тип данных. Арифметические операции для целых чисел. Определения остатка.

Практика. Решение задач с целыми числами.

3.4. Линейный алгоритм.

Практика. Практическая работа «Составление программ по готовым математическим моделям».

3.5. Мини-олимпиада: «Линейный алгоритм».

Практика. Мини-олимпиада: «Линейный алгоритм».

4. Основные алгоритмические структуры на языке Python.

4.1. Условная инструкция if

Теория. Оператор ветвления на Питоне. Полная форма ветвления. Отступы.

Практика. Решение задач на использование условного оператора. Использование оператора ветвление при работе с целыми числами.

4.2. Полный синтаксис условного оператора.

Теория. Полная и Неполная форма ветвления. Сложный оператор. Применение инструкции *elif*.

Практика. Решение задач с использованием условного оператора. Нахождения минимума и максимума.

4.3. Тип данных *bool*, логические операции.

Теория. Логический тип данных. Значения переменных типа *bool*. Логические операции *and*, *or*, *not*.

Практика. Применение логических операций для решения задач на тему «Шахматные этюды».

4.4. Вложенные условные инструкции.

Теория. Возможность использования вложенных условий. Правильная организация отступов. Примеры решения задач.

Практика. Решение задач на использование вложенных условных инструкций.

4.5. Виды циклов. Цикл *for*.

Теория. Циклические алгоритмы. Применение разных видов циклов к решению одной задачи. Синтаксис цикла *For*.

4.6. Функция *range*.

Теория. Счетчик. Генератор списка натуральных чисел *range*. Обратный цикл. Использование параметра в теле цикла.

Практика. Применение цикла с параметром для решения задач над рядом натуральных чисел.

4.7. Варианты применения цикла *for*.

Теория. Условие выхода из цикла. Примеры решения задач с использованием цикла *for*.

Практика. Решение задач на тему «Цикл с параметром». Практическая работа «Циклы».

4.8. Цикл с предусловием.

Теория. Синтаксис цикла *while*. Работа с процентами.

Практика. Практическая работа «Применение цикла с предусловием».

4.9. Вложенные циклы.

Теория. Вложенные циклы. Степень вложенности.

Практика. Решение задач на использование нескольких циклов.

4.10. Строки.

Теория. Строки. Срезы. Ввод строк.

Практика. Решение задач на обработку строк.

4.11. Методы и функции работы со строками.

Теория. Методы *find*, *rfind*. Методы классификации и изменения строк. Функции *max*, *min*, *sorted* в применении к строкам.

Практика. Решение задач на обработку строк.

4.12. Разбор задач повышенной сложности.

Теория. Разбор решения задач школьного и муниципального уровней ВСОШ.

Практика. Решение задач олимпиады муниципального уровня для 7 класса.

4.13. Списки. Работа с элементами списков.

Теория. Понятие списка. Организация и заполнение списка. Работа с элементами списка.

Практика. Решение задач на преобразование списка.

4.14. Методы и функции для работы со списками.

Теория. Замена, удаление, вставка, подсчет элементов.

Практика. Решение задач на обработку элементов списка.

4.15. Линейный поиск в списке.

Теория. Обзор методов поиска элементов в списке. Линейный поиск.

Практика. Решение задач на поиск элементов списка.

4.16. Мини-олимпиада «Начало программирования».

Практика. Мини-олимпиада «Начала программирования».

5. Графический модуль.

5.1. Графические примитивы.

Теория. Подключение графического модуля. Процедуры рисования точки, отрезка, прямоугольника, окружности, круга, овала.

Практика. Применение графических процедур для создания предложенного рисунка.

5.2. Работа с цветом.

Теория. Кодирование цветов. Процедуры закрашивания. Параметры графических операторов, отвечающих за цвет.

Практика. Доработка рисунка с применением заливки цветом.

5.3. Графика и циклы.

Теория. Применение цикла для рисования. Случайные числа.

Практика. Создание рамки из квадратов, окружностей. Рисование звездного неба. Рисование мишени.

5.4. Творческая работа «Графические операторы».

Практика. Создание открытки на тему (по выбору): «8 марта», «День космонавтики», «День леса», «День Балтийского моря».

6.1. Обзор систем счисления.

Теория. История систем счисления. Позиционные и непозиционные системы счисления. Римская система счисления.

Практика. Перевод из римской системы счисления в десятичную. Написание программы перевода десятичного числа <1000 в римскую систему.

6.2. Арифметические операции в различных системах счисления.

Теория. Алфавит позиционной системы счисления. Принцип перехода в следующую позицию числа. Двоичная система. Арифметические операции в двоичной системе. Арифметические операции в других системах счисления.

Практика. Вычисление арифметических примеров в различных системах счисления.

6.3. Перевод из одной системы счисления в другую.

Теория. Принцип степеней основания системы счисления в формировании позиционного числа. Перевод из двоичной системы счисления в десятичную. Перевод из десятичной системы в двоичную. Перевод из других систем счисления в десятичную и обратно.

Практика. Решение задач на перевод из одной системой счисления в другую.

6.4. Машинные системы счисления.

Теория. Восьмеричная и шестнадцатеричная системы счисления. Связь двоичной системы счисления с восьмеричной и шестнадцатеричной систем счисления. Триады, тетрады.

Практика. Самостоятельная работа «Системы счисления»

6.5. Программы перевода различных систем счисления.

Теория. Алгоритм перевода из десятичной системы в двоичную. Алгоритм перевода из двоичной системы в десятичную.

Практика. Написание программы сложения двух двоичных чисел. Написание программы перевода из n -ичной системы счисления в m -ичную.

7. Подготовка к мини-олимпиаде. Мини-олимпиада.

7.1. Решение задач.

Теория. Обзор задач повышенного уровня сложности.

Практика. Решение задач по теме «Программирование на языке Питон».

7.2. Мини-олимпиада.

Практика. Мини-олимпиада.

8. Резервное время учителя.

Содержание материала 3 модуля. Школа олимпиадного программирования.

1. Введение.

Теория. Правила техники безопасности. Обзор материала на новый учебный год. Обзор языков программирования наиболее удобных и часто используемых для решения олимпиадных задач.

Практика. Клавиатурный тренажер. Логические игры. Доски Шерлока.

2. Восстановление навыков написания программ для стандартных и нестандартных алгоритмов.

2.1. Линейный алгоритм. Ветвления.

Теория. Основные процедуры и функции.

Практика. Решение задач на линейный алгоритм и ветвления.

2.2. Циклы. Графика.

Теория. Виды циклов. Графические процедуры.

Практика. Решение задач на использование циклических алгоритмов.

2.3. Символьные переменные.

Теория. Работа со строками и символьным типом.

Практика. Решение задач на обработку строк.

3. Списки.

3.1. Способы ввода и вывода списков. Срезы.

Теория. Структура «список». Доступ к элементам списка. Базовые операции со строками (конкатенация, дублирование, длина). Срезы.

Практика. Решения простых задач с сайта astr.ru на тему «Списки».

3.2. Операции по спискам.

Теория. Методы работы со строками (классификация строк, изменение строк).

Практика. Решения задач с сайта astr.ru на тему «Массивы».

3.3. Двумерные массивы.

Теория. Задание массива. Встроенные библиотеки. Ввод массива. Вывод массива. Заполнение массива по условию. Работа с элементами массива Вложенные списки. Доступ к элементам двумерного массива. Поиск в двумерном массиве.

Практика. Решения задач с сайта astr.ru на тему «двумерные массивы».

3.4. Работа с двумерными массивами.

Теория. Вложенные списки. Доступ к элементам двумерного массива. Поиск в двумерном массиве. Преобразование массива.

Практика. Решения задач с сайта astr.ru по теме «Двумерные массивы».

4. Структуры данных.

4.1. Множества.

Теория. Задание множества. Работа с элементами множеств. Преобразование списка во множество и множества в список.

Практика. Регистрация на сайте pythontutor.ru. Решения задач на сайте pythontutor.ru из темы «Множество».

4.2. Операции с множествами.

Теория. Операции с множествами. Сравнение множеств. Методы `discard` и `remove`.

Практика. Решения задач на сайте pythontutor.ru из темы «Множество».

4.3. Словари.

Теория. Когда нужно использовать словари. Создание словаря. Работа с элементами словаря.

Практика. Решения задач на сайте pythontutor.ru из темы «Словари».

4.4. Работа с элементами словаря.

Теория. Перебор элементов словаря. Методы `keys`, `values`.

Практика. Решения задач на сайте pythontutor.ru из темы «Словари».

5. Введение в олимпиадное программирование.

5.1. Специфика олимпиадных задач.

Теория. Специфика олимпиадных задач. Ограничение по реализации. Требования к эффективности решения. Система тестов.

5.2. Решение простых олимпиадных задач. Подбор тестов.

Практика. Контекст из 3х задач сайта acm.ru.

6. Целочисленная арифметика.

6.1. Делимость. Простые числа. Арифметика остатков.

Теория. Делимость и нахождение остатка. Поиск делителей. Определение простого числа. Проверка числа на простоту.

Практика. Решение задач на делимость. Вывод всех простых чисел из числового интервала.

6.2. Алгоритм Евклида. НОД. НОК.

Теория. Версии алгоритма Евклида – оригинальная, программистская. Нахождение НОД. Формула нахождения НОК.

Практика. Решение задач с сайта acm.ru на использование НОД и НОК.

7. Процедуры и функции.

7.1. Процедуры с параметром. Локальные и глобальные переменные.

Теория. Описание и вызов процедуры. Формальные и фактические параметры. Применение пользовательских процедур.

Практика. Использование процедур при решении задач.

7.2. Пользовательские функции.

Теория. Описание и вызов функции. Виды параметров. Глобальные и локальные объекты. Отличие процедуры от функции.

Практика. Применение пользовательских функций для решения задач.

7.3. Рекурсии.

Теория. Понятие рекурсии. Примеры рекурсивных программ.

Практика. Применение рекурсий для вычисления факториала, нахождения числа E .

8. Алгоритмы для обработки массивов. Динамическое программирование.

8.1. Вспомогательные массивы. Сохранение эффективности алгоритма.

Теория. Использование вспомогательных алгоритмов для ограничения памяти. Подсчет количества одинаковых элементов в массиве.

Практика. Решение задач с использованием вспомогательного массива.

8.2. Сортировки пузырьком, выбором, перестановками.

Теория. Различные методы сортировки. Эффективность сортировок. Количество итераций.

8.3. Быстрая сортировка.

Теория. Принцип рекурсивной сортировки. Возможности языков программирования для сортировки массива.

Практика. Решение задач на сортировку с сайта astr.ru.

8.4. Префиксные суммы.

Теория. Понятие частичных сумм. Формирование массива префиксных сумм. Сокращение времени исполнения алгоритма при использовании префиксных сумм.

Практика. Решение задачи 1081 с сайта astr.ru.

8.5. Рекуррентные отношения. Числа Фибоначчи.

Теория. Рекуррентные отношения. Последовательности. Числа Фибоначчи.

Практика. Нахождение n -ного числа Фибоначчи с помощью составления последовательности и с помощью рекурсии.

8.6. Динамическое программирование. Задача о Кузнечике.

Теория. Понятие динамического программирования. Принципы динамического программирования. Нахождении формулы для формирования очередного элемента массива. Разбор задачи о Кузнечике.

Практика. Задачи на динамическое программирование с сайта astr.ru.

9. Мини-олимпиада.

Практика. Мини-олимпиада.

10. Вычислительная геометрия.

10.1. Прямые. Точки. Отрезки. Длина отрезка.

Теория. Координатная плоскость. Координаты точки. Нахождение длины отрезка. Нахождение площади треугольника.

Практика. Задачи с сайта astr.ru.

10.1. Параллельность и ортогональность прямых.

Теория. Понятия параллельности и ортогональности прямых. Формулы определения параллельности и ортогональности. Вхождение точки в некоторую область.

Практика. Задача о пересечении границы.

10.1. Многоугольники.

Теория. Выпуклые и вогнутые многоугольники. Пересечение сторон многоугольника. Точка центра тяжести. Нахождение площади многоугольника.

Практика. Задача о столе и стекле.

11. Основы логики.

11.1. Алгебра высказываний. Текстовые задачи.

Теория. Высказывания. Истинность. Исторические логические задачи. История логики.

Практика. Решение текстовых задач.

11.2. Логические операции. Таблицы истинности.

Теория. Конъюнкция. Дизъюнкция. Отрицание. Импликация. Эквивалентность. Таблицы истинности. Законы формальной логики.

Практика. Составление таблиц истинности. Определение истинности выражения.

11.3. Решение логических задач.

Теория. Решение комбинаторных задач с помощью алгебры логики.

Практика. Практическая работа на тему «Алгебра логики»

11.4. Задача Эйнштейна. Доски Шерлока.

Теория. Эйнштейн и его открытия. Логическая задача, предложенная Эйнштейном. Современная IT-интерпретация задачи Эйнштейна.

Практика. Программа «Шерлок». Разобрать 3-4 «доски».

12. Итоговое тестирование.

Практика. Итоговое тестирование.

13. Резервное время учителя.

Содержание материала 4 модуля. «Спортивное программирование».

1. Введение.

2. Знакомство с языком программирования C++.

2.1. Структура программы. Линейный алгоритм. Ветвления.

Теория. Алфавит языка C++. Структура программы. Типы данных. Основные функции. Арифметические операции. Оператор присваивания. Оператор ветвления.

Практика. Решения простых задач с сайта aстр.ru на тему «Ветвления».

2.2. Циклы.

Теория. Цикл `for`. Параметры цикла. Возможности цикла. Цикл `While`.

Практика. Решения простых задач с сайта aстр.ru на тему «Циклы».

2.3. Строки.

Теория. Библиотека `string`. Ввод строк. Функции и методы работы со строками.

Практика. Решения простых задач с сайта aстр.ru на тему «Строки».

2.4. Массивы.

Теория. Задание массива. Встроенные библиотеки. Ввод массива. Вывод массива. Заполнение массива по условию. Работа с элементами массива.

Практика. Решения простых задач с сайта aстр.ru по теме «Двумерные массивы».

3. Процедуры и функции. Нисходящее пошаговое структурное проектирование.

3.1. Процедуры и функции.

Теория. Описание. Структура. Механизм передачи параметров. Область действия параметров. Нисходящее пошаговое структурное проектирование.

Практика. Решение задач с использованием процедур и функций.

3.2. Рекурсии.

Теория. Понятие рекурсии. Имитация работы цикла с помощью рекурсии. Быстрые сортировки.

Практика. Разбор задач на рекурсии. Построение драконовой кривой. Примеры рекурсивных алгоритмов.

4. Динамическое программирование.

4.1. Динамическое программирование. Задачи о камнях и рюкзаке.

Теория. Динамическое программирование. Задачи о камнях и рюкзаке. *Практика.* Реализация задач о камнях и рюкзаке.

4.2. Последовательности.

Теория. Последовательности. Построение последовательности в массиве. Задачи на подпоследовательности.

Практика. Реализация построения последовательности по заданной рекурсивной формуле с помощью рекурсии и без рекурсии. Решение задач на последовательности.

4.3. Одномерная динамика.

Теория. Маршруты. Количество маршрутов. Полидромы.

Практика. Решение задач на поиск и количество маршрутов.

4.4. Двумерная динамика на таблицах.

Теория. Задачи о маршрутах на прямоугольном поле.

Практика. Решение задачи «Фермер» с сайта aстр.ru.

4.5. Решение задач по теме «Динамическое программирование».

Теория. Разбор задач на динамическое программирование.

Практика. Решение задач на динамическое программирование с сайта aст.timus.ru.

5. Перебор с возвратом.

5.1. Стек. Организация очереди.

Теория. Понятие стека. Методы изменения стека. Организация очереди с помощью стека и с помощью массива. Сравнение алгоритмов.

Практика. Решение задачи на организацию очереди.

5.2. Использование стека и рекурсии.

Теория. Разбор решений олимпиадных задач с использованием стека.

Практика. Решение задач с использованием стека.

6. Записи.

6.1. Объединение разнородных элементов. Поля.

Теория. Комбинированный тип данных. Запись. Описание записи. Доступ к полям записи. Примеры решения задач

Практика. Ввод данных типа Запись.

6.2. Операции над записями.

Теория. Эффективное использование записей в программе. Использование записей для описания таблиц. Операции над записями.

Практика. Решение задачи 27 из ЕГЭ на умение использовать записи.

7. Жадные алгоритмы.

7.1. Принцип жадного алгоритма. Задача о расписании.

Теория. Знакомство с понятием Жадный алгоритм. Принципы жадных алгоритмов. Разбор решения задачи о расписании.

Практика. Написание программы к задаче о расписании.

7.2. Решение задач на жадные алгоритмы.

Теория. Разбор решений олимпиадных задач с использованием жадных алгоритмов.

Практика. Решение задач на жадные алгоритмы.

8. Теория графов.

8.1. Графы и деревья: определения и примеры

Теория. История. Неориентированные графы. Ребра. Вершины. Петли. Степени вершины. Путь в графе. Связные и несвязные графы. Расстояние между вершинами. Ориентированный граф. Взвешенный граф.

8.2. Матрицы смежности.

Теория. Матрицы смежности. Списки смежности.

Практика. Построение матрицы смежности для графа.

8.3. Алгоритмы на деревьях и графах.

Теория. Обходы деревьев и графов. Нахождение кратчайшего пути, метод «волны», нахождение k кратчайших путей в графе, обход препятствий, трассировка вокруг препятствия, поиск в ширину.

Практика. Решение задач на графы на применение рассмотренных алгоритмов графам.

9. Длинная арифметика.

9.1. Запись числа в массив.

Теория. Длинная арифметика. Числа, выходящие за пределы типа длинного целого. Выделение цифр и запись их в массив.

Практика. Написание программы, считывающей длинное число и помещение его в массив.

9.2. Операции с длинными числами.

Теория. Сложение двух длинных чисел. Произведение длинного числа и числа типа *byte*.

Практика. Написание программы произведения длинного целого на двузначное число.

10. Решение задач по информатике.

10.1. Кодирование информации.

Теория. Единицы измерения информации. Формула Шеннона. Условие Фано. Разбор заданий на тему «Кодирование информации».

Практика. Решение задач на тему «Кодирование информации».

10.2. Чтение и анализ кода программы.

Практика. Анализ текста программ, нахождение и исправление ошибок.

10.3. Разбор ошибок при решении теоретических заданий.

Теория. Разбор решения заданий предыдущего занятия и рассмотрение ошибок, возникших при решении этих задач. Разбор возможных ошибок при решении подобных заданий.

10.4. Комбинаторика.

Теория. Перестановки. Размещения. Сочетания. Применение языка программирования для решения задач на комбинаторику.

Практика. Решение задач на комбинаторику.

11. Подготовка к выпускной работе. Выпускная работа.

11.1. Подготовка к выпускной работе.

Теория. Рассмотрение вопросов тестовой части.

Практика. Решение примерных задач.

11.2. Выпускная работа.

Практика. Выпускная работа.

12. Резервное время учителя.

Планируемые результаты обучения по окончании четырех уровней

Предметные результаты:

Моделирование и формализация. Математические модели.

- приводить примеры моделирования;
- приводить примеры формализации;
- перечислять этапы информационной технологии решения задач с использованием компьютера;
- уметь строить простейшие информационные модели и исследовать их на компьютере;
- уметь строить математические модели для стандартных задач.

Основы алгоритмизации.

- объяснять сущность алгоритма, его основных свойств, иллюстрировать их на конкретных примерах алгоритмов;
- знать свойства алгоритма и понимать их;
- определять возможность применения исполнителя для решения конкретной задачи по системе его команд;
- объяснять структуру основных алгоритмических конструкций и уметь использовать их для построения алгоритмов.

Языки программирования.

- знать основные типы данных и стандартные функции для каждого типа данных;
- знать стандартные процедуры и уметь их использовать;
- знать основные алгоритмические структуры, в том числе ветвления и циклические;
- уметь читать код программы на разных языках программирования;
- знать, что такое система тестов и уметь составлять тесты для анализа программы;
- уметь использовать разные виды циклов, вложенные циклы;
- знать функции и процедуры работы с символьными переменными и уметь применять их для решения задач;
- знать процедуры графического модуля, уметь использовать их для реализации движения в графике;
- уметь разрабатывать и записывать на языке программирования типовые алгоритмы для работы с массивами;
- уметь разрабатывать сложные алгоритмы для задач повышенной трудности и олимпиадных задач.

Системы счисления.

- приводить примеры записи чисел в позиционных и непозиционных системах счисления;
- объяснять позиционный принцип представления чисел в системах счисления;
- описывать правила выполнения арифметических операций в двоичной системе счисления;

- уметь записывать числа в шестнадцатеричной и восьмеричной системах счисления;
- уметь переводить числа из одной системы счисления в другую.

Основы логики.

- знать основные понятия формальной логики;
- уметь применять основные логические операции (инверсия, конъюнкция, дизъюнкция, импликация, эквивалентность);
- представлять логические выражения в виде формул и таблиц истинности.

Длинная арифметика

- знать принцип работы с длинными числами;
- уметь выполнять элементарные арифметические операции с числами превосходящими пределы типа long.

Теория графов

- знать как составляется граф, основные понятия графа;
- уметь преобразовывать граф в табличную форму, уметь использовать алгоритмы работы с графами.

Динамическое программирование и жадные алгоритмы.

- знать принципы рекуррентных соотношений и знать принципы применения жадных алгоритмов;
- уметь решать задачи с использованием динамического программирования и жадных алгоритмов.

Стеки и очереди.

- знать, как организован стек и очередь;
- уметь вносить элементы в стек и извлекать и удалять элементы из стека.

Личностные результаты.

- иметь способность к самообразованию;
- уметь приводить состоятельные аргументы в пользу предложенных технических решений;
- иметь активную жизненную позицию, коммуникативные компетенции, адекватную самооценку, умение анализировать свою и общую деятельность;
- уметь планировать рабочее время.

Метапредметные результаты:

- владеть общепредметными понятиями «объект», «система», «модель», «алгоритм», «исполнитель»;
- уметь самостоятельно планировать путь достижения цели;
- уметь преобразовывать объект из чувственной формы в знаково-символическую модель;
- уметь строить разнообразные информационные структуры для описания объектов.

Комплекс организационно-педагогических условий.

Условия реализации программы:

Материально-техническое обеспечение:

- учебный класс с 14-ю рабочими местами;
- 14 компьютеров с необходимым программным обеспечением (см. в разделе «информационное обеспечение»);
- белая маркерная доска, маркеры для белой доски;
- принтер, бумага для печати;
- проектор.

Информационное обеспечение

Необходимое ПО:

- ОС Windows;
- пакет «Роботландия»;
- пакеты «Мир информатики» 1-2 год обучения и 3-4 год обучения;
- PascalABC.Net;
- Python 3.7;
- Visual Studio C++ или Code blocks;
- логическая игра-головоломка «Sherlock»;
- графический редактор Paint;
- браузер, поддерживающий Flash.

Видео:

- техника безопасности в компьютерном классе;
- анимация и мультипликация;
- история систем счисления;
- машинные системы счисления;
- обзор олимпиадных языков программирования;
- методы сортировки: «пузырек», вставками, быстрая сортировка;
- логические операции;
- динамическое программирование;
- графы: основные понятия.

Презентации

- Линейный алгоритм, Ветвления, Циклы;
- история языков программирования;
- графика в Паскале;
- логические задачи;
- задача Эйнштейна.

Интернет источники:

- studio.code.org – Студия программирования (графические исполнители);
- Phytontutor.ru – он-лайн учебник;
- Acmp.ru – «Школа программиста» он-лайн задачи;
- Codeforces.ru – он-лайн олимпиадные задачи, проведение олимпиад;

- www.onlinegdb.com – он-лайн компиляторы;
- Acst.timus.ru – ол-лайн олимпиадные задачи с турниров УрФУ.

Кадровое обеспечение – педагог ДО Котова Юлия Николаевна, высшая квалификационная категория

Оценочные материалы

1 модуль. «Алгоритмы и исполнители»

Мини-олимпиада по теме «Кукарача»:

6 задач: 1 задача на тему «Линейный алгоритм». 2-3 задача на тему «Повтори». 4-5 задача на тему «Циклы». 6 задача на тему «Ветвление».

Оценивание по занятому месту:

Решено 6 задач – 1 место (оценка 5)

Решено 5 задач – 2 место (оценка 5)

Решено 4 задачи – 3 место (оценка 4)

Решено 3 задачи – без места (оценка 3)

Творческая работа по теме «Студия программирования. Курс 2»

Оценивание по 5-бальной системе:

В каждом из разделов курса решено не менее 10 заданий – 5

В каждом разделе курса решено не менее 7 заданий – 4

В каждом разделе решено менее 7 заданий или не все разделы рассмотрены – 3

Рассмотрено не более 3-х разделов – 2

Итоговая мини-олимпиада.

5 задач: 1 задача из раздела «Кукарача». 2 задача из раздела «Транспортер» 3 задача из раздела «Логические задачи». 4 задача из раздела «Алгоритмы». 5 задача из раздела «Математические понятия».

Оценивание по 5-бальной системе:

Решено 5 задач – 1-2 место (оценка 5). 1 место получает тот, кто первым выполнит все задания.

Решено 4 задачи – 3 место (оценка 4)

Решено 3 задачи – без места (оценка 3)

Решено 1-2 задачи – без места (оценка 2).

Итоговое тестирование:

Тесты, разработанные совместно с психологами для определения уровня логического мышления, внимания и математической логики. Созданы программные продукты для компьютерного тестирования.

Тест Равена:

Полученный результат	Степень	Итог
----------------------	---------	------

95% и больше	1 степень	Особо высокоразвитый интеллект испытуемого
75-94%	2 степень	Незаурядный интеллект испытуемого
25-74%	3 степень	Средний интеллект испытуемого
6-24%	4 степень	Интеллект испытуемого ниже среднего
5% и меньше	5 степень	Дефектная интеллектуальная способность испытуемого

Тест Точки:

Полученный результат	Степень	Итог
8-9 карточек	1 степень	Высоко развитая память
6-7 карточек	2 степень	Средне развитая память
5 и менее	3 степень	Память ниже среднего уровня

Тест Числовой ряд:

Полученный результат	Степень	Итог
11-14 продолженных рядов	1 степень	Отличная математическая логика
8-10 продолженных рядов	2 степень	Хорошая математическая логика
5-7 продолженных рядов	3 степень	Средне развитая математическая логика
менее 5 продолженных рядов	4 степень	Низкая математическая логика

2 модуль. «Школа юного программиста»

Мини-олимпиада по теме «Начала программирования»:

5 задач: 1 задача на тему «Линейный алгоритм». 2 задача на тему «Ветвления». 3 задача на тему «Циклы». 4 задача на тему «Строки». 5 задача на тему «Массивы».

Оценивание по 5-бальной системе:

Решено 5 задач – 1-2 место (оценка 5). Первое место занимает тот, кто первым выполнит все задания.

Решено 4 задачи – 3 место (оценка 4)

Решено 3 задачи – без места (оценка 3)

Решено 1-2 задачи – без места (оценка 2).

Мини-олимпиада:

1. Теоретический тест (9 вопросов).

Темы вопросов: функции и процедуры языка Паскаль, типы данных, целочисленная арифметика, виды циклов, системы счисления.

Оценивание по 5-бальной системе:

Дано верных ответов 9 – 5

Дано верных ответов 7-8 – 4

Дано верных ответов 5-6 – 3

Дано верных ответов менее 5 – 2.

2. 15 вариантов заданий по 3 задачи в каждом.

1 задача на тему «Ветвление». 2 задача на тему «Строки». 3 задача на темы: «Графика», «Массивы», «Циклы».

Оценивание каждой задачи:

Решено все верно – 5

Сдано решение с одной ошибкой или не проходит один тест – 4

Сдано решение с ошибками не более трех (в том числе и синтаксических) – 3

Задача не решена или решена с более чем 3 ошибками – 2.

4 оценки складываются в общий балл. Итоговая оценка:

18-20 баллов – 5 (1 место)

15-17 баллов – 4 (2 место)

11-14 баллов – 3 (3 место)

Менее 11 баллов – работа не выполнена.

3 модуль. «Школа олимпиадного программирования»

Итоговая работа:

1. Тест 2 варианта (5 вопросов).

Темы вопросов: Системы счисления, изучение кода программы на разных языках программирования, логические выражения, таблицы истинности.

Оценивание по 5-бальной системе:

Дано верных ответов 5 – 5

Дано верных ответов 4 – 4

Дано верных ответов 3 – 3

Дано верных ответов 1-2 – 2.

2. 16 вариантов заданий по 3 задачи в каждом.

Темы задач: «Сложные ветвления», «Вложенные циклы», «Сортировка массива», «Последовательности», «Целочисленная арифметика», «Префиксные суммы», «Рекурсии».

Оценивание каждой задачи:

Решено все верно – 5

Сдано решение с одной ошибкой или не проходит один тест – 4

Сдано решение с ошибками не более трех (в том числе и синтаксических) – 3

Задача не решена или решена с более чем 3 ошибками – 2.

4 оценки складываются в общий балл. Итоговая оценка:

18-20 баллов – 5 (1 место)

15-17 баллов – 4 (2 место)

11-14 баллов – 3 (3 место)

Менее 11 баллов – работа не выполнена.

4 модуль. «Спортивное программирование»

Выпускная работа:

1. Теоретический тест 2 варианта (10 вопросов).

Темы вопросов: Виды алгоритмов, логические выражения, системы счисления, графы, динамика, исследование кода программы на выбранном языке программирования.

Оценивание по 5-бальной системе:

Дано верных ответов 10 – 5

Дано верных ответов 7-9 – 4

Дано верных ответов 5-6 – 3

Дано верных ответов 1-4 – 2.

2. 16 вариантов заданий по 2 задачи в каждом.

Темы задач: «Циклы», «Одномерные массивы», «Двумерные массивы», «Очереди», «Динамическое программирование», «Целочисленная арифметика», «Графы».

Оценивание каждой задачи:

Решено все верно – 5

Сдано решение с одной ошибкой или не проходит один тест – 4

Сдано решение с ошибками не более трех (в том числе и синтаксических) – 3

Задача не решена или решена с более чем 3 ошибками – 2.

3 оценки складываются в общий балл. Итоговая оценка:

14-15 баллов – 5 (1 место)

12-13 баллов – 4 (2 место)

9-11 баллов – 3 (3 место)

Менее 9 баллов – работа не выполнена.

Литература для учащихся

1. Златопольский Д.М. Сборник задач по программированию. 3-е изд., перераб. и доп. — СПб.: 2011. — 304с
2. Кристофидес Н. Теория графов. Алгоритмический подход. – М.: Мир, 1978.
3. Огнева М.В., Кудрина Е.В. ОЗ8 Turbo Pascal: первые шаги. Примеры и упражнения: Учеб. пособие. Изд.3-е, перераб. и доп. - Саратов: Изд-во “Научная книга”, 2008.
4. Окулов С.М., Программирование в алгоритмах. – БИНОМ, 2002.
5. Попов В.Б. Turbo Pascal для школьников. Финансы и статистика, 2006. – 528с.
6. Попов В. Б. Паскаль и Дельфи: самоучитель / Попов В. Б. — СПб.и др.: Питер, 2004. — 543с.
7. <http://ru.scribd.com/doc/76931800/ABC-Pascal> – А.С.Цветков. Язык программирования PASCAL. Система программирования ABC Pascal. Учебное пособие для 7-го класса. 2008-2009 г.

Литература для педагога

1. Беляев С.Н. Язык программирования С++; Красноярск, 2018. - 60 с.
2. Долинский М. С. Алгоритмизация и программирование на TurboPascal: от простых до олимпиадных задач, 2005
3. Милов А.В. Основы программирования в задачах и примерах. Издательство: Фолио, 2002. – 401с.
4. Порублев И.Н., Ставровский А.Б. Алгоритмы и программы. Решение олимпиадных задач. – М.: Вильямс, 2007
5. Фаронов В. В. Турбо Паскаль 7.0. Практика программирования: учебное пособие / Фаронов В. В. — М.: ОМД Групп, 2003. — 415с.
6. <http://el-prog.narod.ru/pascal.html> - Программирование для начинающих.
7. <http://mif.vspu.ru/books/pascal-tasks/> - Turbo Pascal 7.0. Задания для лабораторных занятий по программированию.
8. <http://ru.scribd.com/doc/76931800/ABC-Pascal> - А.С.Цветков. Язык программирования PASCAL. Система программирования ABC Pascal. Учебное пособие для 7-го класса. 2008-2009 г.
9. Беляев С.Н., Кормышов М.Д., Лалетин Н.В. Региональные олимпиады по информатике. 2010-2011 учебный год: учебно-методическое пособие / Научно-образовательный центр «Перспектива». - Железногорск, 2011. - 212 с.
10. Решение олимпиадных задач по программированию. Красноярск, 2017. - 100 с.