

**АДМИНИСТРАЦИЯ АНГАРСКОГО МУНИЦИПАЛЬНОГО
ОБРАЗОВАНИЯ
УПРАВЛЕНИЕ ОБРАЗОВАНИЯ**

**МБОУ «СОШ №10 с углубленным изучением отдельных предметов»
МАУ «Ангарский лицей №2»**

ПРОГРАММА

**факультатива «Как научить компьютер понимать
человека (программирование и алгоритмизация)»**

(рассчитана на 68 часов для уча-
щихся 8-9 классов инновационного общеоб-
разовательного учреждения)

автор:

Трифорова О.Ю.-
учитель информатики высшей категории

2023 г.

Актуальность

Факультатив по данной программе проходил ежегодно в как минимум в двух образовательных учреждениях. Вывод от применения программы – программирование актуально не только при очном изучении олимпиадных алгоритмов, но и при дистанционном обучении. Результат работы по программе – ежегодные призеры и победители по программированию не только на муниципальном, но и региональном и федеральном уровнях.

В связи с введением стандартов 2 поколения, перехода к новым условиям обучения, а также с введением Нового закона об образовании необходимо адаптировать учащихся к быстро изменяющимся внешним условиям.

В связи с актуализацией и активизацией олимпиадного движения все острее встает проблема подготовки учащихся к участию в олимпиадах. Чем раньше ребенок начинает серьезно заниматься какой-то учебной дисциплиной, тем больше шансов получить отличный результат.

Возникает вопрос: для чего же нужно изучать программирование? Есть два ответа на этот вопрос. Во-первых, принцип программного управления работой компьютера является одним из фундаментальных принципов информатики. А во-вторых, с точки зрения профориентационной функции предмета программирование является профессиональной областью деятельности, весьма важной, современной и престижной. Профессия программиста – одна из самых дефицитных в мире. Программированию учатся всю жизнь. Подготовка программиста начинается с изучения основных понятий и овладения простейшими приемами. В третьих, языки программирования ПУТОН и СИ в настоящее время широко используется в технических ВУЗах для изучения не только основ программирования. В школьном курсе информатики на программирование выделяется недостаточное количество часов и начинают изучать программирование только в 9-м классе.

Программирование как тема курса информатики, с одной стороны, и как профессиональная деятельность, с другой стороны, в информационном обществе приобретает все большее значение.

Небольшой объем часов в курсе школьной информатики, выделяемый на изучение темы «Алгоритмизация и программирование» в 8 и 9 классе, и, одновременно с этим, возрастающие потребности общества, а также проводимые олимпиады всех уровней: от школьного до международного, с узкой направленностью на программирование требуют выявления учащихся, способных мыслить алгоритмически и в последствии писать программы на языках программирования высокого уровня, на более ранних ступенях обучения.

При обычном обучении информатики, темы «алгоритмы» и «программирование» изучаются очень мало и поздно, это замедляет формирование алгоритмического мышления, не способствует развитию интереса учащихся в области программирования, учащиеся, как правило, не готовы успешно выступать на олимпиадах по информатике, теряют интерес к предмету. Обучение по данной программе создает благоприятные условия для более раннего «погружения» учащихся в мир логики, математического моделирования, для интеллектуального и духовного воспитания личности ребенка, социально - культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

Есть категория школьников, с математическим складом ума. Программирование, по большому счету, не требует зубрежки алгоритмов и математических понятий. Важнее выработать особый тип мышления. Поэтому в возрасте 13-14 лет можно начинать серьезно изучать программирование. Ни одна образовательная программа -- будь то Угринович, Макарова, Семакин, Босова, не ставят это своей целью. И изучение сред таких как Лого-Миры, Робот, Чертежник, Паркетчик - малоэффективны и не востребованы, хотя красочны и интересны.

Актуальность программы осуществляется тем, что учащиеся должны понимать значение алгоритмизации и программирования в жизни общества, уметь выделять систему поня-

тий, характерную для конкретной образовательной области, пользоваться ей, расширять её объём, уметь переносить полученные знания, способы и приёмы деятельности из одной образовательной области в другую.

Курс является ориентационным в системе предпрофильной подготовки:

- ✓ способствует расширению кругозора и включает оригинальный материал, выходящий за рамки школьной программы,
- ✓ расширяет кругозор учащихся,
- ✓ осуществляет учебно-практическое знакомство с проблемой методов и принципов программирования.

Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

Настоящая программа актуальна для инновационных учебных заведений, так как разработана в соответствии с концепцией углубленного и профильного обучения. Программа рассчитана на учащихся инновационных общеобразовательных.

Новизна программы:

Программирование является «стержнем информатики». В нем синтезировано все, что десятилетиями нарабатывалось. Это результаты работы специалистов, работающих на «стыке» математики и информатики, достижения в вычислительной технике, огромный опыт формализации и решения сложнейших проблем в программировании.

Программа содержит дополнительный изучаемый материал (рекурсии, графы, олимпиадные задачи), значительно расширяет возможности формирования универсальных учебных и предметных навыков. В данном курсе на конкретных примерах рассматривается программирование на языках высокого уровня Pascal и СИ. Показаны основные методы составления программ и примеры использования их при решении некоторых физических, математических, экономических и других задач.

Программа основывается на доступности материала и построена по принципу «от простого к сложному». Тематика занятий разнообразна, что способствует творческому развитию ребенка, фантазии, самореализации. Обучение строится таким образом, чтобы учащиеся хорошо усвоили приемы работы в среде программирования, научились «читать и понимать» простейшие алгоритмы и программы, а затем и создавать свои для решения практических и олимпиадных задач. Постепенно образуется система специальных навыков и умений, формируется интерес к творчеству, пробуждается желание творить самостоятельно. Творческое начало и безграничная фантазия заложены в каждом ребенке.

При обучении программированию в курсе информатики наиболее сложным для учащихся оказывается самопроверка (отладка и тестирование собственных программ). Дальнейшая же проверка правильности выполнения учебных заданий преподавателем, в свою очередь, является наиболее трудоемкой частью процесса обучения и предполагает проработку методики будущего оценивания ожидаемых решений. Если при проверке решения теоретических задач основное внимание уделяется анализу хода решения и правильности полученного ответа, то при решении практических задач ситуация иная. Несмотря на важность работ в области доказательства правильности программ, предлагаемые в них методы можно и нужно использовать в основном во время разработки алгоритма решения задачи. Завершением же этапа реализации (собственно программирования) является тестирование программы, то есть выявление особенностей представленного решения на наборе специально подобранных тестов. Признавая очевидные преимущества проведения тестирования программ с целью оценки их работоспособности и эффективности, многие преподаватели искренне полагают, что результаты тестирования никогда не могут дать достоверной информации о качестве программы и уж тем более оценить степень обученности учащихся. Тем не менее, тестирование уже много лет используется в мире как средство оценки решений различных задач, и это ведет к совершенствованию систем тестов, появлению в данной области

деятельности новых компьютерных технологий, в большей степени отвечающих потребностям методологии современного программирования. Целью, в отличие от традиционного обучения, является усвоение не только результатов научного познания и системы знаний, но и поиск самого пути. В результате формируется познавательная самостоятельность ученика, и развиваются его творческие способности. Таким образом, проблемное обучение - это оптимальное сочетание репродуктивной и творческой деятельности по передаче и усвоению системы научных понятий и приемов, способов логического мышления. Это дидактический подход, учитывающий психологические закономерности мыслительной деятельности субъектов. С внедрением компьютера в процесс обучения впервые появилась возможность, в несколько раз повысив активность учащихся, обеспечить цикличность функционирования традиционного контура обратной связи "преподаватель-ученик" в реальном масштабе времени. В результате намного проще стало реализовывать ведущие принципы развивающего обучения: индивидуализацию и дифференциацию. Кроме того, компьютер, позволяя ошибаться, разрешая ошибаться, создавая возможность ошибаться, дает возможность познавать через противоречия. А, как показано Ж.Пиаже, ошибочные теории учеников являются существенной частью процесса формирования мышления.

Методологическое обоснование программы:

Программа рассчитана на 68 часов для учащихся 8-9 классов (либо по одному часу в 8 и 9 классе, либо 2 часа в 9). Программа ориентирована на обучение по машинному варианту в сетевом классе на базе современных ЭВМ.

Основными формами организации занятий в курсе должны быть лекционно-практические занятия. Изложение ведется на основе проблемно-поискового подхода. Учащиеся работают над конкретными содержательными задачами и выполняют проектные работы, как групповые, так и индивидуальные. Выполнение каждого проекта учащимися завершается составлением отчета, что позволяет систематизировать знания по изучаемому разделу и обобщить полученные результаты.

Занятия по информатике в корне отличаются от традиционных занятий по любому другому предмету. Во – первых на занятиях по программированию должна поощряться ошибка, так как только через ошибку можно прийти к результату. Во-вторых, постоянная обратная связь с обучаемым через компьютер, объективная и лишенная эмоций, - это инструментальный индивидуальный и развивающего обучения. В - третьих, стиль мышления у программистов свой, отличающийся от стиля мышления как математика, так и любого другого специалиста. Любая сложная программа – это миллион составляющих, движущихся и взаимодействующих. В результате этого взаимодействия должен получиться определенный результат.

Данная программа содержит в себе фундаментальную часть изучения языка программирования PASCAL или C++.

Конкретные требования к уровню знаний, умений и навыков по каждому разделу программы приведены в таблице «Содержание курса» данной рабочей программы.

Цель спецкурса: развитие логического и алгоритмического мышления школьников, формирование творческих навыков, побуждение интереса к изучению программирования.

Задачи спецкурса:

- ✓ развитие интереса учащихся к изучению программирования;
- ✓ формирование творческого подхода к решению задач на программирование;
- ✓ формирование навыков алгоритмического мышления;
- ✓ приобретение навыков работы в системах программирования;
- ✓ формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами, умения соблюдать нормы информационной этики и права;

Результат:

Личностные результаты:

- широкие познавательные интересы, инициатива и любознательность, мотивы познания и творчества; готовность и способность учащихся к саморазвитию и реализации творческого потенциала в духовной и предметно-продуктивной деятельности за счет развития их образного, алгоритмического и логического мышления;
- готовность к повышению своего образовательного уровня и продолжению обучения программированию;
- готовность к самостоятельным поступкам и действиям, принятию ответственности за их результаты; готовность к осуществлению
- индивидуальной и коллективной информационной деятельности;
- способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и
- технических условий безопасной эксплуатации средств ИКТ;
- формирование коммуникативной компетентности в процессе образовательной, учебно-исследовательской, творческой и других
- видов деятельности.

Метапредметные результаты:

- владение основными общеучебными умениями информационно-логического характера: анализ объектов и ситуаций;
- синтез как составление целого из частей и самостоятельное достраивание недостающих компонентов; выбор оснований и критериев для сравнения и классификации объектов; обобщение и сравнение данных; подведение под понятие, выведение следствий;
- установление причинно-следственных связей;
- построение логических цепочек рассуждений;
- владение умениями организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить; планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;
- прогнозирование – предвосхищение результата; контроль – интерпретация полученного результата, его соотнесение с имеющимися данными с целью установления соответствия или несоответствия (обнаружения ошибки);
- коррекция – внесение необходимых дополнений и корректив в план действий в случае обнаружения ошибки; оценка – осознание учащимся того, насколько качественно им решена учебно-познавательная задача;
- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы; поиск и выделение необходимой информации, применение методов информационного поиска; структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий;
- самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера;
- владение информационным моделированием как основным информационным моделированием как основным методом приобретения знаний;
- умение создавать вербальные и графические модели, «читать» чертежи и схемы, самостоятельно переводить алгоритм на язык программы;
- опыт принятия решений и управления объектами (исполнителями) с помощью составленных для них алгоритмов (программ);
- владение основами взаимодействия и сотрудничества со сверстниками и взрослыми;

- умение правильно, четко и однозначно сформулировать мысль в понятной собеседнику форме;
- умение осуществлять в коллективе совместную информационную деятельность; владение устной и письменной речью;
- развитие компетентности в области использования информационно-коммуникационных технологий в учебной деятельности и повседневной жизни.

Предметные результаты:

- умение использовать термины «объект», «среда», «исполнитель», «команда», «алгоритм», «программа», «процедура», «угол», «вектор»;
- понимание различий между употреблением этих терминов в обыденной речи и в алгоритмике;
- умение различать системы команд исполнителей;
- умение задавать углы поворота и векторы перемещения исполнителей;
- умение определять координаты исполнителей;
- умение выбирать необходимую алгоритмическую структуру;
- умение составлять алгоритмы управления исполнителями и записывать их на языке программирования;
- умение формально выполнять алгоритмы;
- умение выделять в программе процедуры;
- умение отлаживать и выполнять программу по шагам;
- знание требований к организации компьютерного рабочего места, соблюдение требований безопасности и гигиены в работе с компьютером.

По завершению курса учащийся научится программировать на хорошем уровне, что будет способствовать его профессиональному самоопределению и успешной сдаче экзамена по информатике в 9 классе.

Для тех ребят, которые свою будущую профессию связывают с информатикой, программированием, представление об идеях и методах программирования должно складываться еще в школе. Обычно школьники плохо представляют, где и как они смогут воспользоваться своими знаниями. Те ученики, для которых красота математических идей и рассуждений представляет ценность сама по себе, не нуждаются в дополнительных обоснованиях целесообразности изучения таких разделов математики, как теория графов или комбинаторика, но для более прагматически настроенных учеников имеет смысл показать перспективу использования данных тем в их дальнейшем обучении.

Критерии отслеживания ре-зультатов

- ✓ *текущий*: отчеты по факту выполнения контрольного блока, предусмотренного в конце каждой темы.
- ✓ *итоговый*: выполнение итогового проекта.
- ✓ участие в городских и областных конкурсах и выставках.

Краткая структура спецкурса

Наименование разделов и поурочных тем	<i>Количество часов</i>
1. Массивы. Работа с массивами	16 часов
2. Сортировки. Методы сортировки	4 часа
3. Геометрия. Задачи с отрезками	4 часа
4. Длинная арифметика	6 часов
5. Понятие процедуры и функции	8 часов
6. Рекурсия. Использование рекурсивных процедур и функции	6 часов
7. Понятие графа	2 часа
8. Применение графов для решения задач. Алгоритм Дейкстры	8 часов
9. Комбинаторика. Понятие комбинаторного алгоритма	2 часа
10. Решение задач на комбинаторные алгоритмы	6 часов
11. Резерв	6 часов

Итого: 68 часов

Описание разделов программы учебно-тематического плана

№ занятия	Основное содержание по темам	Содержание учебного материала	Характеристика основных видов деятельности ученика (на уровне учебных действий)
1	2	3	
1	Массивы. Работа с массивами.	Понятие массива. Описание типа. Одномерный (линейный) массив	<p>Аналитическая деятельность:</p> <ul style="list-style-type: none"> определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p>Практическая деятельность:</p> <ul style="list-style-type: none"> решать задачи на составление алгоритмов и программ; разрабатывать и отлаживать программы в выбранной среде программирования; составлять документации программ по образцам
		Основные приемы присвоения значений элементам массива (определение их константы, организация расчета их по формулам, ввод с клавиатуры, использование генератора случайных чисел)	
		Алгоритмы нахождения суммы, произведения и количества элементов массива. Алгоритм поиска	
		Перестановки элементов в одномерном массиве. Поиск по условию. Работа с индексами элементов массива	
		Поиск минимального и максимального элемента в массиве	
		Отбор нужных элементов	
		Работа с несколькими массивами	
		Двумерный массив (матрица элементов). Описание двумерного массива. Заполнение и вывод массива в виде прямоугольной матрицы	
		Действия над элементами двумерного массива	
		Свойства диагоналей двумерного массива	
		Поиск в двумерном массиве	
		Бинарный поиск	
		Работа со строками и столбцами в двумерном массиве	
		Алгоритмы обработки матриц	
		Крестики-нолики. Обработка элементов матрицы	
Слияние массивов			
2	Алгоритмы сортировки массивов и последовательности.	Понятие сортировки. Сортировка методом пузырька	<p>Аналитическая деятельность:</p> <ul style="list-style-type: none"> определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива.
		Сортировка методом вставки	
		Подсчет числа инверсий. Сортировка подсчетом	
		Решение задач с применением сортировки	

			<p>Практическая деятельность:</p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы в выбранной среде программирования; • составлять документации программ по образцам
3	Геометрия. Задачи с отрезками	Пересечение отрезков	<p>Аналитическая деятельность:</p> <ul style="list-style-type: none"> • определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p>Практическая деятельность:</p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы в выбранной среде программирования; • составлять документации программ по образцам
		Проверка принадлежности точки прямой и плоскости. Метод половинного деления	
		Определение площадей фигур. Метод Монте – Карло. Метод прямоугольников. Метод Симпсона.	
		Решение геометрических задач	
4.	Длинная арифметика	Представление чисел. Реализация процедуры считывания длинного числа.	<p>Аналитическая деятельность:</p> <ul style="list-style-type: none"> • определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p>Практическая деятельность:</p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы в выбранной среде программирования; • составлять документации программ по образцам
		Реализация процедуры вывода длинного числа. Сложение длинных чисел.	
		Решение задач на действия с много-разрядными числами.	
		Умножение длинного числа на короткое.	
		Вычитание длинных чисел.	
		Деление длинного числа на короткое	
		Решение задач на действия с много-разрядными числами.	
		Сравнение длинных чисел.	

			тации программ по образцам	
5	Понятие процедуры и функции	Понятие процедуры. Основные определения. Виды процедур. Параметры процедур	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> решать задачи на составление алгоритмов и программ; разрабатывать и отлаживать программы в выбранной среде программирования; составлять документации программ по образцам 	
		Способы передачи параметров		
		Способы применения процедур		
		Процедура-функция. Отличие процедуры-функции от процедуры		
		Особенности применения процедур и процедур-функций		
		Использование процедур для решения задач. Модульный тип построения программ.		
Решение задач на применение процедур и функций				
		Возвращаемые значения в процедурах и функциях		
6	Рекурсия. Использование рекурсивных процедур и функции	Понятие рекурсивной процедуры-функции. Особенности применения рекурсии		<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> решать задачи на составление алгоритмов и программ; разрабатывать и отлаживать программы в выбранной среде программирования; составлять документации программ по образцам
		Рекурсивная обработка деревьев		
		Топологическая сортировка		
		Как обойтись без рекурсии. Таблица значений		
		Стек отложенных заданий		
		Точка выхода из рекурсии (как избежать ошибки)		
7	Понятие графа	Основные понятия. Виды графов. Поиск на графе и его обход	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> определять зависимость времени работы программы (количества шагов выполнения) от размера ис- 	
		Проверка связности графа с ненаправленными ребрами. Выделение связной компоненты графа		

			<p>ходных данных, например длины массива.</p> <p>Практическая деятельность:</p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы в выбранной среде программирования; • составлять документации программ по образцам
8	Применение графов для решения задач. Алгоритм Дейкстры	Паросочетания и реберные покрытия. Метод увеличивающих путей.	<p>Аналитическая деятельность:</p> <ul style="list-style-type: none"> • определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p>Практическая деятельность:</p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы в выбранной среде программирования; • составлять документации программ по образцам
		Паросочетания в двудольных графах. Паросочетания в произвольных графах (алгоритм Эдмондса)	
		Задача об оптимальном каркасе. Алгоритм Прима. Алгоритм Крускала	
		Жадные алгоритмы. Взвешенные паросочетания	
		Связные графы с неотрицательными весами ребер	
		Кратчайшие пути в графе	
		Алгоритм Дейкстры	
		Решение задач на применение алгоритма Дейкстры	
9	Комбинаторика. Понятие комбинаторного алгоритма	Классические операции комбинаторики. Генерация комбинаторных объектов. Перестановки. Размещения. Сочетания.	<p>Аналитическая деятельность:</p> <ul style="list-style-type: none"> • определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p>Практическая деятельность:</p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы
		Разбиение числа на слагаемые. Подмножества.	
		Формирование комбинаторных групп из N по K	

			<p>в выбранной среде программирования;</p> <ul style="list-style-type: none"> • составлять документации программ по образцам
10	Решение задач на комбинаторные алгоритмы	Решение задач на размещения с повторениями	<p><i>Аналитическая деятельность:</i></p> <ul style="list-style-type: none"> • определять зависимость времени работы программы (количества шагов выполнения) от размера исходных данных, например длины массива. <p><i>Практическая деятельность:</i></p> <ul style="list-style-type: none"> • решать задачи на составление алгоритмов и программ; • разрабатывать и отлаживать программы в выбранной среде программирования; • составлять документации программ по образцам
		Решение задач с перестановками	
		Использование разбиений для решения задач	
		Коды Грея	
		Подсчет количества	
11	Резерв	Резерв используется на усмотрение учителя, для отработки полученных знаний при решении задач	

Средства обучения

1. Компьютерные классы, оснащенные персональными компьютерами.
2. В состав программных средств должны входить:
 - операционная система WINDOWS;
 - MS Office 2007 и выше;
 - демонстрационные базы данных;
 - программы для демонстрации учебных фильмов;
 - среды разработки: Паскаль, Си, Питон и тд
3. Раздаточный материал для индивидуальной работы учащихся по всем разделам программы.

ЛИТЕРАТУРА

ЛИТЕРАТУРА, НЕОБХОДИМАЯ УЧИТЕЛЮ

1. И.К. Сафронов Задачник – практикум по информатике.Издательство «БХВ - Петербург» 2012 г..
2. Г. Семакина, Е. Хеннер Задачник – практикум в 2 т. Издательство «Лаборатория базовых знаний» 2012г.

ЛИТЕРАТУРА, ИСПОЛЬЗОВАННАЯ ПРИ СОСТАВЛЕНИИ ПРОГРАММЫ

1. Н. Угринович, В. Морозов, В.Нечаев, преподавание курса «Информатика и информационные технологии»: Методическое пособие для учителей. Издательство «БИНОМ. Лаборатория знаний». 2012.

ЛИТЕРАТУРА, ИСПОЛЬЗУЕМАЯ УЧАЩИМИСЯ

1. Г. Семакина, Е. Хеннер Задачник – практикум в 2 т. Издательство «Лаборатория базовых знаний» 2012г.
2. Школа программиста, сайт <http://acmp.ru/>

ПЕРЕЧЕНЬ ДИДАКТИЧЕСКИХ МАТЕРИАЛОВ, РАЗРАБОТАННЫХ УЧИТЕЛЯМИ ИНФОРМАТИКИ

1. Разноуровневые дидактические материалы.
2. Проекты с заданиями.