

П. 2.10

РЕСПУБЛИКА ДАГЕСТАН
МУНИЦИПАЛЬНОЕ КАЗЁННОЕ УЧРЕЖДЕНИЕ
«УПРАВЛЕНИЕ ОБРАЗОВАНИЯ»
МО «НОВОЛАКСКИЙ РАЙОН»

368040 тел 8 (242) 21-4-81 21-4-83 факс 8 (242) 21-4-81
ОГРН - 1020500910726

с. Новолакское
ИНН 0524004098

ПРИКАЗ

«23» октября 2023г.

№ 800 - ОД

**Об утверждении плана проведения семинаров по изучению языков
программирования на уроках информатики**

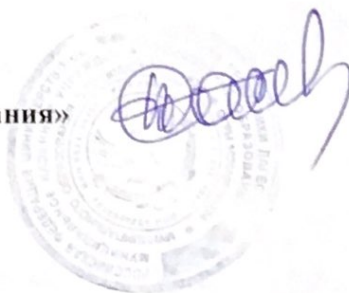
В соответствии с планом мероприятий по повышению качества образования в МО «Новолакский район» на период до 2026 года, в целях повышения эффективности образовательной деятельности посредством индивидуализации образования, применения современных образовательных технологий, объектно-ориентированных языков программирования, создания цифровой образовательной среды, непрерывного совершенствования профессионального уровня и педагогического мастерства педагога информатики общеобразовательных организаций МО «Новолакский район» в 2022 -2024 учебном году

ПРИКАЗЫВАЮ:

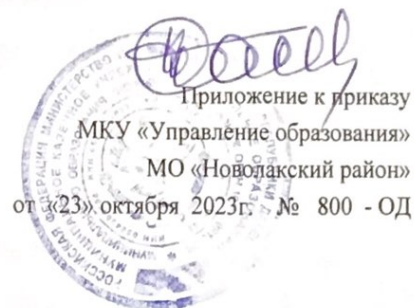
1. Утвердить план проведения семинаров по изучению языков программирования на уроках информатики в общеобразовательных организациях МО «Новолакский район» (далее -План)(Приложение 1)
2. Заведующему методическим кабинетом Давлетмурзаевой К.И. проконтролировать реализацию Плана.
3. Методисту МКУ «Управление образования» Шахмардановой Ш.Н. подготовить аналитическую справку по итогам реализации Плана до 01.05.2024г.
4. Руководителям образовательных организаций провести организационные мероприятия согласно Плану.
5. Контроль за исполнением данного приказа возложить на заместителя начальника УО Атлангериева Р.С..

приложение на 7л. в 1 экз.

Начальник
МКУ «Управление образования»



Кудиева С.Р.



Приложение к приказу
МКУ «Управление образования»
МО «Новолакский район»
от «23» октября, 2023г. № 800 - ОД

План
проведения семинаров по изучению языков программирования на уроках
информатики на 2023-2024 учебный год

Методическая тема семинаров по информатике в ОО МО «Новолакский район» на 2023—2024 учебный год:

Повышение эффективности образовательной деятельности посредством индивидуализации образования, применения современных образовательных технологий, объектно-ориентированных языков программирования, создания цифровой образовательной среды, непрерывного совершенствования профессионального уровня и педагогического мастерства педагога.

Цели работы семинаров:

1. Повышение качества обучения информатике;
2. Понимание педагогами ведущих современных образовательных концепций;
3. Приобретение педагогами практического опыта педагогического проектирования, объектно-ориентированных языков программирования;
4. Освоение педагогами современных образовательных технологий деятельностного типа;
5. Развитие творческого потенциала педагогов;
6. Формирование позитивного образа семинаров как эффективно действующего органа, способствующего профессиональному росту педагогов;
7. Организационно-методическое и информационное сопровождение мероприятий, связанных с развитием детской одаренности, популяризацией предмета;
8. Осознание педагогами необходимости в непрерывном повышении собственного профессионального и интеллектуального уровня.

Ожидаемые результаты:

1. Повышение уровня профессиональной компетенции педагогов.
2. Повышение уровня успеваемости, качества знаний учащихся.
3. Внедрение современных технологий деятельностного типа в образовательную практику.
4. Диссеминация положительного педагогического опыта.
5. Успешное участие школьников в предметных олимпиадах, конкурсах, научно-исследовательской и проектной деятельности.
6. Повышение интереса учащихся к предмету.

7. Формирование положительного отношения к современным образовательным концепциям у педагогов.

№	Содержание работы	Сроки исполнения	Ответственные
1.	Выбор языка программирования. Методика проведения уроков при изучении раздела «Программирование»	Октябрь 2023 г МКОУ «Новолакская гиназия»	Методист УО курирующий информатику. Руководитель РМО
2.	«Ввод и вывод информации»	Ноябрь 2023г. МКОУ «Ямансуйская СОШ»	Методист УО курирующий информатику. Руководитель РМО
3.	«Вычисления в программе»	Декабрь 2023г. МКОУ «Ахарская СОШ»	Методист УО курирующий информатику. Руководитель РМО
4.	«Ветвление в программе. Условный оператор»	Январь 2024г МКОУ «Новомехельтинская СОШ»	Методист УО курирующий информатику. Руководитель РМО
5.	Изучение темы «Циклы»	Февраль 2024г. МКОУ «Шушинская СОШ»	Методист УО курирующий информатику. Руководитель РМО
6.	Изучение темы «Массивы»	Март 2024г. МКОУ «Чапаевская СОШ №1»	Методист УО курирующий информатику. Руководитель РМО
7.	Изучение темы «Строки»	Апрель 2024 г. МКОУ «Чапаевская СОШ №2»	Методист УО курирующий информатику. Руководитель РМО

Пояснение к разделу «Программирование»

(Информатика)

Выбор языка программирования.

Вопрос выбора языка программирования, на мой взгляд, не решается однозначно в пользу объектно-ориентированных языков.

К преимуществам преподавания в школе объектно-ориентированных языков могу отнести

современный вид оболочки и выполняемой программы. Кроме того, у некоторых детей папа, мама или старшие братья и сестры на работе пишут программы на таких языках. И наконец, "его" можно купить в магазине, взять у приятеля, а не переписать со старых дискет из запасов учителя.

Это всё является для учащихся стимулирующими факторами.

К недостаткам отношу следующее.

Во-первых, среда объектно-ориентированных языков программирования достаточно сложна для 85% учащихся 8-9 классов общеобразовательных школ (я не рассматриваю физмат лицей и лицей информационных технологий). Закрытое нужное окно или панель часто ведет к бессмысленному сидению у компьютера растерянного ученика до подхода преподавателя. Да и не понятно многим ученикам, куда что писать. Объектов и окошек много. С обучением программированию, сводящемуся к "нажмите эту клавишу" я не согласна в принципе. В каждом классе есть дети, которые могут самостоятельно запрограммировать простейшие задачи. Программирование развивает мышление и логику ребенка.

Во-вторых, время, необходимое для получения программы, при использовании объектно-ориентированных языков больше. При планировании решения задачи за несколько учебных дней возникают проблемы с теми учащимися, кто не был в первой или второй день. Кроме того, дети забывают, что делали. Преподавание информатики по два спаренных урока в 8-9 классах не предусмотрено санитарными нормами. За 20-25 минут, которые ребенок может провести за компьютером, большинство учеников может написать и отладить (или списать, для некоторых и это хорошо) небольшую учебную программу на языках Pascal или Qbasic. Но только треть учащихся может довести работу до конца при использовании языков Visual Basic или Delphi.

В-третьих, на ЕГЭ и на вступительном экзамене возникают проблемы. На Pascale или Qbasicе все равно нужно ребенку поработать.

Программированию на объектно-ориентированных языках должно предшествовать программирование на Pascale или Qbasicе. Причем моя практика показывает, что в соблюдении приемственности языков Pascal - Delphi и Qbasic - Visual Basic нет необходимости. Если в 10 (9) классе был Pascal, то в 11 (10) переход к новому языку Visual Basic осуществляется за один урок (с проведением параллелей между языками). За каникулы дети все равно многое забыли. В случае знакомства с двумя языками в дальнейшем в институте у учащихся будет меньше проблем.

При переводе базового курса в 8-9 классы вопрос об языке программирования, на мой взгляд, решается так: базовый уровень - Pascal или Qbasic, профильный уровень - Pascal или Qbasic, затем Delphi или Visual Basic.

Методика проведения уроков при изучении раздела "Программирование"

Традиционный "оргмомент" при изучении этой темы часто сводится к индивидуальным приветствиям при входе в кабинет и вопросам о разрешении сесть за компьютер. Дети, сделавшие домашнее задание на компьютере, обычно стремятся зайти в класс и подготовиться к показу своей работы еще во время перемены. Те же, у кого что-либо не получилось, тоже стремятся завладеть вниманием учителя перед уроком. В конце урока при программировании на компьютере тоже часто возникают ситуации, когда дети не спешат покинуть кабинет информатики даже при постоянном отказе учителя принимать задания после звонка с урока. Поэтому я стараюсь составить поурочное планирование так, чтобы программирование в одной параллели совпадало по времени с теоретическими темами в других параллелях. Это позволяет, кроме того, уменьшить накапливаемое статическое напряжение в кабинете информатики, которое ощутимо при работе 10 компьютеров 1997 г. выпуска в условиях отсутствия кондиционера и прочих благ.

Домашние задания выполняются на компьютере (в нашей школе сейчас около 90% учащихся

имеют дома компьютеры) или в тетради по возможности и желанию учащегося. При выполнении этих заданий лучше предоставить детям полную свободу модификации задания. Проверка домашнего задания осуществляется в начале урока. Интересные варианты показываются всей группе. Откладывание проверки домашнего задания на время после уроков и выполнение этого учителем без ученика лишат учителя возможности высказать замечания по программе. Не писать же рецензию на треть страницы. Кроме того, через несколько дней такая проверка потеряет актуальность. И еще, ученик сам откроет и запустит программу, а учитель только посмотрит. Это быстрее. Иногда выполнение этой процедуры выдает ученика, скопировавшего чужую работу. Текст программы нужно смотреть обязательно. Только там часто видны ошибки и входные данные, на которых эти ошибки проявляются.

Ученики, не выполнившие домашнее задание (но не те, у кого не получилось что-либо и есть что показать) делают иногда на уроке задачу на доске или за партой под присмотром учителя, в то время как остальные работают за компьютерами. Однако следует учитывать, что при этом дети, трудившиеся дома, получают гораздо меньше внимания учителя. Поэтому я использую такой прием только, когда необходимо позаботиться об оценках. Не знаю, как в других регионах, а московских школах система четвертных и тем более годовых оценок трехбалльная, 3-4-5. Ребенок, не выполняющий домашние задания и не работающий самостоятельно в классе, не сможет написать нормально контрольную, поэтому приходится проявлять особую заботу о таких детях.

Количество заданий и сами задания по каждой теме зависят от состава учащихся. По каждой теме несколько программ целесообразно записывать на доске. При этом объясняются типичные ошибки и показываются стандартные приемы. В слабых группах количество задач, написанных на доске, должно быть больше, чем в сильных группах. Такие задачи на компьютере затем не набираются.

При составлении программ на доске при решении задач нового для учеников типа после объяснения операторов эффективно приглашение к доске учеников, желающих попробовать написать программу. Желающие всегда найдутся, если дети будут знать, что в этом случае учитель, во-первых, поможет и, во вторых, не поставит плохую отметку. Лучше просто посадить ученика, если он явно не справляется с программой. Только выбирайте добровольца или предлагайте выступить в этом качестве нескольким ученикам, которые, на Ваш взгляд, смогут выполнить задание. Но этот прием можно использовать только при решении задач на новую тему. Иначе дети быстро приспосабливаются и Вам будут обеспечены стройные ряды добровольцев, не открывающих дома учебник и тетрадь.

К подбору задач нужно относиться внимательно. Решение математических задач на получение рядов, конечных и бесконечных сумм, задач с использованием векторной алгебры в условиях обычной школы вынуждает учителя информатики затратить не менее 10-15 минут на объяснение математической сути каждого типа задач и каждой задачи, если речь о векторной алгебре. У учеников со слабой математической подготовкой возникает установка на непонимание дальнейшего материала, то есть собственно информатики. Кроме того, часто затруднена самостоятельная проверка ребенком результата работы программ.

Последние годы я вообще стала избегать задач, требующих знаний математики выше нахождения периметра, площади, степени числа и вычисления по заданной формуле даже при работе со старшими детьми в общеобразовательной школе. Хотя считаю, что такие задачи очень полезны для актуализации математических знаний и с удовольствием бы вела раздел или курс "Математические основы программирования". Но информатики только 1 или 2 урока в неделю и программирование только один из многих разделов программы курса "Информатика".

Не стоит пренебрегать использованием тестов при изучении темы "Программирование".

В тесты включаю теоретические вопросы, вопросы на определение результата выполнения операторов и вопросы синтаксического характера. Наличие синтаксических вопросов

положительно сказывается на ускорении отладки программ за компьютером. В приложении приведены вопросы двух тестов: теста “Переменные” для языка Pascal и теста “Условный оператор” для языка Visual Basic.

Лучше проводить компьютерное тестирование с использованием тестирующих программ со случайным выбором вопроса. В этом случае ученики вынуждены будут работать самостоятельно. Такой тест можно, при необходимости, сделать повторно. Можно поставить обе отметки. Вопросы ведь почти все разные. При условии выдачи информации о правильности каждого ответа компьютерное тестирование выполняет не только контролирующую функцию, но и обучающую. Ответ проверяется сразу же, а не через несколько дней, когда ребенок и вопрос забыл. Листок с проверенным письменным тестом ребенком часто рассматривается только с точки зрения носителя полученной им отметки. Кроме того, отметка, поставленная компьютером, для ребенка обезличена. Необходимо позаботиться и о сохранении результатов теста в формате, исключающем ручную правку ребенком. Причем дети должны знать о том, что их работа протоколируется компьютером.

Контрольные работы делаются либо на компьютере, либо на бумаге. В первом случае я даю только одну задачу на урок (у меня нет спаренных уроков) и последние 10 минут объясняю на правильных детских работах решение задачи каждого варианта. Оценка ставится по степени близости к завершению задачи. При бумажном варианте в работе три задачи разного уровня сложности. Использую и экспресс-работы: одна несложная задача на время – 15 минут и без ошибок – “5”, 20 минут и без ошибок – “4”. При наличии ошибок оценка снижается, работа для исправления ученику не дается.

Изучение темы “Ввод и вывод информации”

Эта тема не обязательно должно быть первой темой при изучении программирования. Можно начать с решения графических задач, а при решении задач, требующих ввод необходимых данных объяснить технологию ввода информации. Можно дать простые графические программы в теме “Представление информации” как пример векторной графики. Так или иначе, перед переходом к вычислительным задачам дети должны уметь вводить и выводить значение переменных. Работа с файлами может быть дана только в сильных группах или на факультативных занятиях.

Изучение темы “Вычисления в программе”

Запись арифметических операторов не вызывает трудности у детей, особенно если ранее решались вычислительные задачи с помощью электронных таблиц. Простые вычислительные задачи характерны соблазном для детей ввести результат с клавиатуры. Поэтому следует обращать внимание детей на то, какие переменные являются входными, а какие – выходными данными. При выполнении вычислений значений функций таких проблем не возникает.

Изучение темы “Ветвление в программе. Условный оператор”

Особое внимание нужно обратить на запись условий. Условия попадания на отрезок, в заданную область, выполнения определенных соотношений между переменными, условия типа “все положительно”, “только одно положительно” должны быть отработаны отдельно. Понятны детям задачи, в которых в качестве операндов используются строковые переменные, значение которых вводится в процессе решения задачи.

Изучение темы “Циклы”

В 7-9 классах кажется целесообразным рассмотреть циклы на графических задачах. Графика эмоциональна и легко проверяема самими детьми. При выборе задач по теме “Графика” следует учесть, что в курсе математики тема “Вектора” изучается в 9 классе. Понятие о радиане тоже дается только в конце 9 класса. Поэтому дуги и сектора в 7-9 классах не рисуем. В приложении дан краткий конспект модуля “Графика. Циклы” для 7-9 классов.

Изучение темы “Массивы”

Темы “Циклы” является одной из самых сложных тем для понимания ребенком. Учащиеся охотно решают приближенные к реальным задачи на обработку статистической информации (задачи об экзаменах, депутатах, вузах и т.д.) с использованием пользовательских типов переменных (Qbasic, Visual Basic) или записей (Pascal, Delphi) и массивов. Смысл таких задач понятен учащимся. Объясняю тему “Массивы” (возможно, с опросом), затем сразу “Пользовательские типы (записи)” без решения формальных задач по массивам. Затем решаем задачи с пользовательскими типами. А уж потом можно без проблем решать задачи на формальную обработку одномерных и двумерных массивов.

Возникающие при такой последовательности проблемы в случае использования объектно-ориентированных языков уже относятся к информатике и являются углублением знаний учащихся. При решении задач с пользовательскими типами времени на каждую задачу нужно больше, так как в этом случае я требую смысловое название переменных. У учащихся уходит больше времени на набор текста программы. Быстрее набрать `A(i)`, чем `redion(i).deputat`.

Для избежания повторного ввода можно дать работу с файлами. Но тогда возникает необходимость режимов: ввода, чтения из файла, дозаписи и т.д. Появляется меню и из задачи вырисовывается небольшая справочная система или база данных. Но это уже для профильного уровня. На базовом уровне лучше, для избежания больших затрат времени на ввод данных, вводить условные данные, например, фамилии депутатов -Ф1, Ф2, Ф3.

Изучение темы “Строки”

Перед этой темой нужно коротко рассказать о подпрограммах, формальных и фактических параметрах. Подробно тему “Подпрограммы” можно только в сильной группе или на факультативе. Работа со строками дает прекрасную возможность закрепить и систематизировать знания по темам “Циклы”, “Массивы”. Смысл задач понятен детям, а результаты работы программ легко проверяются. Но, как и в других предметах, ребенок легко решает задачу, когда она стоит в разделе на какую-либо тему, но испытывает трудности при выборе способа решения, если та же задача находится в разделе обобщения и нужно самому предложить алгоритм решения. В некоторых случаях стоит ограничиться простейшими задачами с использованием строк.