**Развитие навыков программирования учащихся посредством использования разноуровневых заданий на уроке информатики**

Авторы: Сергиенко Дарья Михайловна, Ахмутинова Унзиля Мухамедкалиевна, Нурмуханова Асель Сериковна

НИШ ХБН г.Петропавловск

**Абстракт.**

Информатика является важной областью образования, где накопление новых знаний и методов решения задач является важным процессом, требующим вычислительного подхода мышления во всех сферах человеческой деятельности. В данной статье рассматривается вопрос развития у учащихся 9 класса навыков программирования. На актуальность темы указывают такие факторы, как: учащиеся по окончании основной школы сдают внешний экзамен по информатике, в старшей школе введён новый предмет – программирование. Исследование обусловлено необходимостью улучшения навыков в программировании через различные формы заданий, где учащиеся смогут решать задачи используя определенный язык программирования. Авторами разработан сборник разноуровневых заданий, где каждый учащиеся может самостоятельно выполнять задания по уровням сложности. Это будет способствовать самоорганизации и осознанной подготовке к СОР, СОЧ. Сборник содержит задания связанные с разделом программирования в 9 классе на следующие темы как, вложенные циклы, блок-схемы, таблицы трассировок на циклические алгоритмы c вложенными циклами, практические задания на различные алгоритмы обработки двумерных массивов.

**Ключевые слова**: дифференциация, разноуровневые задания, программирование, самостоятельное обучение, сборник задач.

**Введение.** Развитие навыков программирования становится все более важным в современном мире. Каждый год появляются новые технологии и требования к профессиональным навыкам, и умение программировать является одним из ключевых в этой области. Развитие навыков программирования является одной из наиболее важных задач в современном образовании. Однако, для того чтобы ученики смогли эффективно развивать свои навыки, необходимо использовать разноуровневые задания на уроках информатики. Это позволит ученикам работать на своем уровне и постепенно улучшать свои навыки. В условиях современного общества, где информационные технологии и программирование играют ключевую роль, важно сформировать у учащихся базовые навыки программирования. При этом учитель должен учитывать индивидуальные особенности каждого ученика, уровень их знаний и умений. Разноуровневые задания на уроке информатики могут стать эффективным инструментом для развития навыков программирования учащихся, учитывая разный уровень их подготовки.

Дифференциация в образовании - это процесс адаптации обучения к различным потребностям, уровням знаний, интересам и способностям учеников в классе (Tomlinson, 2014). Дифференциация включает в себя использование различных методов обучения, материалов и оценки, а также различные уровни сложности заданий, чтобы учитывать индивидуальные различия учащихся (Subban, 2006). В целом, дифференциация направлена на улучшение результатов обучения и повышение мотивации учащихся через персонализированный подход к обучению (McTighe & O'Connor, 2005).

*Цель исследования:* Исследовать влияние применения разноуровневых заданий по программированию на уроках информатики в контексте развития навыков учащихся.

Исходя из цели исследования были поставлены следующие *задачи:*

* Изучить теоретические аспекты применения разноуровневых заданий на уроке информатики.
* Определить критерии для разработки разноуровневых заданий по программированию.
* Разработать примеры разноуровневых заданий для учащихся.
* Провести экспериментальное исследование на выборке учащихся, используя разработанные задания.
* Проанализировать результаты эксперимента и сделать выводы о влиянии разноуровневых заданий на развитие навыков программирования учащихся.

*Гипотеза:* Применение разноуровневых заданий на уроках информатики способствует эффективному развитию навыков программирования учащихся, учитывая их индивидуальные особенности и уровень подготовки.

*Объект исследования:* Учащиеся 9 класса, изучающие программирование на уроках информатики.

*Предмет исследования:* Влияние применения разноуровневых по программированию на уроках информатики в контексте развития навыков учащихся.

Инструменты исследования: Анкетирование, наблюдение, тестирование.

**Методология.**

*Теоретическая часть:* В данной части исследования проведён обзор литературы, связанной с разноуровневыми заданиями на уроках информатики и развитием навыков программирования учащихся. Например, Fisher, Frey и Pumpian (2012) описывают, как использование разноуровневых заданий может помочь учителям адаптировать свой подход к обучению под потребности и уровни знаний различных учеников, и как это может улучшить результаты обучения. А Бондаренко (2018) рассматривает использование разноуровневых заданий и программирования в дифференцированном обучении. Автор описывает, как учителя могут создавать индивидуальные задания, которые соответствуют уровню знаний и потребностям каждого ученика, и как это может помочь им достигнуть большего успеха в обучении. Семенова и Колесникова (2018) обсуждают роль дифференциации и разноуровневых заданий в стимулировании самостоятельного обучения. Авторы описывают, как использование этих методов может помочь учащимся развивать свои навыки и повышать мотивацию к обучению.

Рассмотрены различные подходы к формированию навыков программирования, а также преимущества и недостатки использования разноуровневых заданий. Определены критерии для разработки разноуровневых заданий по программированию, учитывающие возраст, уровень знаний и способности учащихся.

*Экспериментальная часть:* в ходе эксперимента были разработаны и использованы сборники разноуровневых заданий для учащихся. Эксперимент проводился в НИШ ХБН г. Петропавловска с участием учеников 8 и 9 классов. Были сравнены результаты по разделам программирования на предыдущих годах обучения, то есть в 7 и 8 классах.

По результатам эксперимента были собраны данные о качестве выполнения заданий учащимися, их прогрессе в навыках программирования и мотивации. Данные были проанализированы, что позволило сделать выводы о влиянии разноуровневых заданий на развитие навыков программирования.

Разноуровневые задания по программированию на уроках информатики могут быть организованы на трех уровнях: теоретическом, аналитическом и эмпирическом.

На теоретическом уровне ученики должны знать основные понятия и принципы программирования, такие как переменные, операторы, условные конструкции, циклы и функции. Также они должны иметь понимание структур данных и алгоритмов. Теоретические задания могут включать в себя написание кода по конкретной задаче, объяснение того, как он работает, или рассуждения о его эффективности.

На аналитическом уровне ученики должны уметь анализировать и исправлять ошибки в коде, создавать более сложные алгоритмы и программы, и понимать, как разные части кода взаимодействуют друг с другом. Аналитические задания могут включать в себя задачи на определение проблем в коде, усовершенствование алгоритмов, анализ результата работы программы и сравнение разных вариантов решения задач.

На эмпирическом уровне ученики должны иметь возможность применять свои знания и навыки на практике. Это может включать в себя написание кода для решения реальных задач, например, разработки игры, приложения или веб-сайта. Эмпирические задания могут также включать в себя тестирование программы, чтобы убедиться в ее корректности и работоспособности.

**Результаты.**

Результаты исследований показывают, что разноуровневые задания по программированию на уроках информатики имеют множество преимуществ. Они позволяют учителю адаптировать задания к уровню каждого ученика, а также развивают у учеников навыки решения проблем, критического мышления и творческого подхода к решению задач. Однако, такие задания требуют большего количества времени на подготовку и проведение урока, и могут быть сложными для некоторых учеников, которые могут испытывать трудности на более высоких уровнях.

Существует множество методов, которые помогают повысить навыки программирования у учащихся на уроках информатики:

1. Практика. Навыки программирования можно повысить только путем практики. Учителя могут давать учащимся множество задач и примеров кода, чтобы они могли попрактиковаться в их написании, анализе и исправлении ошибок.
2. Постановка целей. Учащиеся будут мотивированы заниматься программированием, если будут поставлены четкие цели. Цели должны быть конкретными, измеримыми, достижимыми, релевантными и ограниченными по времени.
3. Коллективное программирование. Работа в группах может помочь учащимся совместно решать задачи, делиться знаниями и опытом, а также повышать свои навыки программирования.
4. Обратная связь. Учителя должны давать учащимся обратную связь по их работе. Это поможет им понять свои ошибки и улучшить свои навыки.
5. Использование онлайн-ресурсов. Существует множество онлайн-ресурсов, которые могут помочь учащимся повысить свои навыки программирования. Это могут быть курсы, учебники, видеоуроки, вебинары, форумы и т.д.
6. Создание проектов. Работа над собственными проектами может помочь учащимся улучшить свои навыки программирования, а также научиться решать реальные проблемы.

При разработке эффективных разноуровневых заданий мы применяли все эти методы, чтобы ученики могли постепенно развивать и улучшать свои навыки программирования. Это позволило нам составить набор заданий на каждом уровне, которые соответствовали уровню знаний учащихся, чтобы ученики могли постепенно развивать навык программирования.

Для учеников со слабой академической способностью А, со средней академической способностью В и с сильной академической способностью С использовали простые задания на знание синтаксиса языка программирования С++ для написания кода с работой двумерного массива. Далее учащиеся выполняли задания по уровню сложности - задания на создание простых алгоритмов, задания на разработку более сложных алгоритмов.

Задания для учащихся уровня А, включали в себя задачи на отработку базовых навыков, т.е. создать простую программу, которая выводит двумерный массив, пользователь вводит двумерный массив, программа выводит таблицу умножения на экран. Эти задания помогли ученикам понять, как работать с массивами и как выводить данные на экран.

Задания для учащихся уровня В, которые уже знакомы с понятием "двумерные массивы" и могут работать с ними, включали в себя написание программы, которая находит сумму элементов двумерного массива, произведение четных элементов в массиве, подсчитать количество чисел кратных 2 или 5. Эти задания помогли ученикам узнать, как работать с массивами, как выполнять математические операции с элементами, как использовать составные условия и циклы в массиве.

Задания для учащихся уровня С, которые уже владеют навыками работы с двумерными массивами и могут решать сложные задачи, были предложены на написание программ, которая находит среднее арифметическое элементов двумерного массива, сортирует двумерный массив по возрастанию или убыванию. Эти задания позволили учащимся развить свои навыки работы с массивами, а также сортировки и поиска алгоритмов. Учащиеся уровня С помогали учащимся уровня В и А в поисках ошибок улучшить свои навыки программирования.

Каждое задание, представленное в сборнике было ясным и конкретным, с определенной целью отработки навыков программирования. Выполняя задания, ученики понимают, что именно они должны сделать, и как это сделать. После выполнения разноуровневых заданий по программированию во время урока либо консультаций важно оценивать результаты учеников и обеспечивать обратную связь. Учитель должен оценивать задания в соответствии с уровнем сложности, поощрять учеников, которые справляются со сложными заданиями, и помогать тем, кто испытывает трудности.

В рамках исследования был проведен опрос среди учащихся «На сколько улучшились ваши навыки в программировании?». Результаты представлены на рисунке 1 ниже.

Рисунок 1. Результаты опроса



Один из преимуществ использования разноуровневых заданий заключается в том, что каждый ученик может выбрать для себя подходящий уровень сложности. Это помогает сохранить интерес и мотивацию к изучению программирования. Кроме того, разноуровневые задания помогают учителям определить уровень усвоения школьной программы каждым учеником и подстроить урок под его потребности.

**Заключение/обсуждение.**

Таким образом, разноуровневые задания способствуют индивидуализации обучения программированию, учитывая разные уровни знаний и способности учащихся. После внедрения сборников с разноуровневыми заданиями учащиеся продемонстрировали более высокий уровень мотивации и интереса к программированию, что свидетельствует о положительном влиянии методики на их вовлечение в учебный процесс.

Анализ результатов показал, что учащиеся в 8 классе имели 75% качества знаний, а в 9 классе имеют 73,5% качества. Однако, на данную отрицательную динамику могли повлиять и другие факторы, а именно: наличие одного суммативного оценивания за раздел и усложненная тема раздела. Однако, мотивация учащихся к изучению программирования в 4 четверти уже значительно возросла.

Таким образом, гипотеза исследования подтвердилась: применение разноуровневых заданий на уроках информатики способствует эффективному развитию навыков программирования учащихся, учитывая их индивидуальные особенности и уровень подготовки.

Рекомендации для педагогов:

Использовать разноуровневые задания для индивидуализации обучения программированию, адаптируя материал к потребностям и уровню учащихся.

Регулярно проводить анализ и корректировку разноуровневых заданий, опираясь на результаты тестирования и обратную связь от учащихся.

Обеспечить доступность и прозрачность критериев оценивания для учащихся, чтобы они могли адекватно воспринимать свой прогресс и понимать, как улучшить свои навыки программирования.

Мотивировать учащихся к самостоятельному изучению и освоению программирования за пределами учебного процесса, предоставляя им информацию о дополнительных ресурсах, конкурсах и проектах.

Подобный подход позволяет учитывать индивидуальные особенности и уровень подготовки учеников, стимулируя их интерес и мотивацию к изучению программирования. Педагогам рекомендуется активно использовать разноуровневые задания в своей практике, опираясь на результаты данного исследования и собственный опыт.

**Список литературы:**

1. Tomlinson, C. A. (2014). Leading and Managing a Differentiated Classroom (2nd ed.). Alexandria, VA: ASCD.
2. Subban, P. (2006). Differentiated Instruction: A Research Basis. International Education Journal, 7(7), 935-947.
3. McTighe, J., & O'Connor, K. (2005). Seven Practices for Effective Learning. Educational Leadership, 63(3), 10-17.
4. Fisher, D., Frey, N., & Pumpian, I. (2012). The differentiated classroom: Responding to the needs of all learners. Alexandria, VA: ASCD.
5. Бондаренко, О. (2018). Программирование и разноуровневые задания в процессе дифференцированного обучения. Москва: Просвещение.
6. Семенова, Е., & Колесникова, Л. (2018). Самостоятельное обучение и дифференциация в образовании. Москва: Просвещение.