

можно найти в разнообразных интернет-источниках (приложение).

Использование искусственного интеллекта на уроках иностранного языка поможет учащимся совершенствовать свои навыки и быстрее достичь поставленных целей в изучении иностранного языка.

Однако, несмотря на все возможности и преимущества использования современных образовательных технологий в процессе обучения иностранным языкам, мы не должны забывать о важности роли самих преподавателей. Именно учитель является ключевой фигурой в образовательном процессе. Он может вдохновить и мотивировать своих учащихся, обеспечить доверие и поддержку, индивидуальный подход и предопределить успешность обучения. Технологии, включая искусственный интеллект, должны служить средством для усиления компетенций и профессионализма преподавателя, а не заменять его.

Список литературы

1. Кондрахина, О.А. Использование возможностей искусственного интеллекта для преподавания иностранных языков: новая реальность [Электронный ресурс] / О.А. Кондрахина, О.Н. Петрова. – Режим доступа: <https://cyberleninka.ru/article/n/ispolzovanie-vozmozhnostey-iskusstvennogo-intellekta-dlya-prepodavaniya-inostrannyh-yazykov-novaya-realnost>. – Дата доступа: 14.06.2024.

2. Пырнова, О.А. Технологии искусственного интеллекта в образовании [Электронный ресурс] / О.А. Пырнова, Р.С. Зарипова. – Режим доступа: <https://cyberleninka.ru/article/n/tehnologii-iskusstvennogo-intellekta-v-obrazovanii>. – Дата доступа: 02.06.2024.

3. Шобонов, Н.А. Искусственный интеллект в образовании [Электронный ресурс] / Н.А. Шобонов,

М.Н. Булаева, С.А. Зиновьева. – Режим доступа: <https://cyberleninka.ru/article/n/iskusstvennyy-intellekt-v-obrazovanii-1>. – Дата доступа: 02.06.2024.

Приложение

Ссылки и QR-коды доступа к интернет-источникам информации по использованию искусственного интеллекта на уроках иностранного языка

<https://clck.ru/37LWfZ>



<https://clck.ru/37LWk9>



<https://clck.ru/37LWoi>



<https://clck.ru/37LX7s>



<https://clck.ru/37LXBP>



Дата поступления в редакцию: 24.06.2024

ИСПОЛЬЗОВАНИЕ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON КАК АЛЬТЕРНАТИВНОГО ИНСТРУМЕНТА ПРИ ИЗУЧЕНИИ ТЕМЫ «АЛГОРИТМИЗАЦИЯ И ПРОГРАММИРОВАНИЕ»

Цыманович Владимир Вячеславович

Аннотация. Использование языка программирования Python является альтернативным инструментом при изучении темы «Алгоритмизация и программирование» в средней школе. Python, с его простым и интуитивно понятным синтаксисом, является отличным выбором для введения школьников в мир программирования. В статье рассматриваются преимущества использования Python в образовательном процессе: доступность, многофункциональность и широкие возможности для решения задач и создания различных проектов. Автор статьи предлагает методику внедрения Python в учебный план по алгоритмизации и программированию, включая разработку уроков, задач и проектов, адаптированных к возрастным и когнитивным особенностям учащихся. Приводятся примеры задач, которые могут быть реализованы с использованием Python.

Раздел «Алгоритмизация и программирование» является стержнем курса информатики, однако помимо интереса он всегда вызывал у учащихся большие трудности при его изучении. Как результат, развитие

таких черт личности, как невнимательность, неусидчивость, отсутствие мотивации изучать программирование. Учащийся рассматривает себя неуспешным при изучении данной темы, не прослеживает связи

со своей будущей профессиональной деятельностью. Возникает противоречие: с одной стороны – высокий социальный заказ, а с другой стороны – невысокие результаты, которые демонстрируют учащиеся при изучении данной темы, особенно в сравнении с другими разделами курса информатики.

Одним из вариантов разрешения данного противоречия является использование альтернативного языка программирования. Динамичное развитие информационных технологий в современном мире приводит к появлению все новых языков. Традиционно в школьном курсе информатики отводится место на рассмотрение базовых алгоритмических конструкций и основ программирования на одном из языков высокого уровня, который должен быть учебным и универсальным, иметь достаточно простой синтаксис и низкие аппаратно-системные требования. В традициях отечественного образования прочно закрепился язык программирования Pascal, однако в последнее время получили распространение и другие языки.

Основные преимущества программирования Python – это его универсальность и простота, а также свободная лицензия и открытые исходные коды. Он подходит для выполнения вычислений и обработки данных, для разработки программного обеспечения, создания веб-сайтов, разработки компьютерных игр, а также для решения многих других прикладных задач.

Выбор языка программирования, на котором учащимся будут написаны первые программы, безусловно, играет важную роль.

Тема «Алгоритмизация и программирование» реализуется в деятельности учителя на уроке, во внеклассной работе (факультатив, стимулирующие, олимпиады), в индивидуальной работе с высокомотивированными учащимися (научно-исследовательская деятельность, самообразование).

Рассмотрим задачу нахождения суммы двух чисел на разных языках программирования в рамках учебной деятельности на уроке (таблица 1):

PascalABC	Python	C++
<pre>var i,j:longint; begin read(i,j); writeln(i+j); end.</pre>	<pre>a,b=input().split() rint(int(a)+int(b))</pre>	<pre>#include <iostream> int main() { int x,y; std::cin>>x; std::cin>>y; std::cout<<x+y; return 0; }</pre>

Таблица 1

Уже такой несложный пример демонстрирует, что код на языке Python гораздо лаконичнее аналогичных решений на других языках. Это достигается во многом за счет динамической типизации, так как раздел описания переменных в таком случае отсутствует. Python стремится к более простому, менее громоздкому синтаксису. Одной из интересных особенностей

языка является выделение структурных частей кода с помощью отступов (пробелов или табуляций), поэтому в Python отсутствуют операторные скобки begin/end, как в языке Паскаль, или фигурные скобки, как в C++. Это позволяет сократить количество строк и символов в программе и приучает к «хорошему» стилю программирования.

Использование мобильных телефонов и планшетов как устройств для работы с информацией привело к низкому уровню владения клавиатурой, что отмечают все учителя информатики. Зачастую ученик не может решить задачу именно потому, что просто не успевает набрать код программы. Сократить время, необходимое для набора кода, и позволяет использование Python. Гораздо эффективнее рассмотреть и решить 5–6 задач вместо 2–3, не тратя время урока на работу с клавиатурой.

В качестве примера, демонстрирующего эффективность использования Python на уроке, рассмотрим несколько заданий из учебника X класса по теме «Массивы». Условие задачи: «Написать программу, которая введет элементы массива с клавиатуры и выведет сумму третьего и пятого элементов» (таблица 2).

Решение задачи на языке	
PascalABC	Python
<pre>var a:array [1..10] of integer; n,s:integer; begin writeln('введите количество элементов в массиве >=5'); readln(n); writeln('введите ',n,' чисел через пробел'); for var x:=1 to n do read(a[x]); s:=a[3]+a[5]; writeln('сумма элементов = ',s); end.</pre>	<pre>print('введите элементы массива') d = list(map(int,input().split())) s=d[2]+d[4] print('Сумма элементов =',s)</pre>

Таблица 2

Разница в длине кода впечатляет. Она достигается за счет отсутствия блока описания переменных, нет необходимости использования циклов для ввода данных, нет операторных скобок begin/end. Учащийся, который пишет программу на Python, успеет решить 2 и более задачи за то время, которое он бы потратил на работу с кодом в Pascal для одной задачи.

В некоторых задачах эта разница будет еще серьезнее. Условие: «Задан одномерный массив из целых чисел. Найти сумму элементов массива» (таблица 3).

При решении приведенной задачи можно использовать функцию sum, которая находит сумму элементов массива. Эта полезная функция есть в Python, но отсутствует в Pascal. Применение встроенных возможностей языка делает решение многих задач школьно-

Решение задачи на языке	
PascalABC	Python
<pre>var a:array [1..10] of integer; n,s:integer; begin writeln('введите количество элементов в массиве >=5'); readln(n); writeln('введите 'n,' чисел через пробел'); for var x:=1 to n do read(a[x]); s:=0; for var x:=1 to n do s:=s+a[x]; writeln('сумма элементов = 's); end.</pre>	<pre>print('введите элементы массива') d = list(map(int,input().split())) print('Сумма элементов =',sum(d))</pre>

Таблица 3

го курса проще и понятнее для учащихся.

Объяснение нового материала на уроке также можно организовать эффективнее. В качестве примера приведем фрагмент конспекта урока информатики в VIII классе по теме «Ветвление». (Приложение 1).

Обязательным элементом творческой деятельности является *свобода выбора пути решения задачи*. Даже если учитель уверен в существовании более правильного, рационального решения, он не в праве настаивать на нем. Ученик имеет право на ошибку: умный не тот, кто не делает ошибок, а тот, кто их не повторяет.

Для учащихся с высоким и средним уровнем учебной мотивации предлагается дополнить конспекты собственным вариантом решения задачи либо интересным алгоритмом. В результате получается набор задач, который позволяет увидеть совершенно разные подходы к данной теме. Например, конспект по теме «Нахождение НОД и НОК» (Приложение 2).

Внеклассная работа (факультатив, стимулирующие, олимпиады) дает возможность инициировать обучение программированию с начальной школы. Для учащихся III–IV классов организованы стимулирующие занятия. В доступной форме ребята знакомятся с синтаксисом языка, средой разработки, главными понятиями программирования. Для объяснения работы основных алгоритмических конструкций применяются блок-схемы. Используем опорные конспекты, где ребята видят соответствие команд в программе и блоков. (Приложение 3). Если уровень мотивации учащихся высок, предлагается задание на заполнение пропущенных мест в конспекте. (Приложение 4). Это способствует развитию навыков самостоятельной формулировки задач и выбора оптимального пути их решения, анализа полученных результатов.

Работа по обучению программированию продолжается на II ступени общего среднего образования.

Именно в подростковом возрасте проявляется осознанный интерес к самостоятельной интеллектуальной деятельности. Однако эмоциональная неустойчивость, колебания настроения, характерные для этого возраста, существенно влияют на результаты творческой деятельности. Чтобы создать ситуацию успеха для каждого ребенка, необходимо разрабатывать дифференцированные задания. (Приложение 5). Очень важно продемонстрировать, что задача имеет не одно и даже не два решения. Разнообразие возможностей Python позволяет придумывать решения, которые зависят от уровня знаний, и одну задачу можно решать разными способами множество раз.

Одаренный и высокомотивированный ученик должен развиваться не только на уроке. Эффективной формой дифференцированного обучения и воспитания является факультатив. Его основное предназначение – углубление и расширение знаний, развитие способностей и интересов учащихся. Язык программирования Python является отличным инструментом для подготовки к участию в олимпиадах и конкурсах различного уровня. Олимпиады по информатике являются по сути своей олимпиадами по программированию.

Рассмотрим задачу городского тура олимпиады по информатике для учащихся VII–VIII класса. Условие: *«Вася и Петя играют в следующую игру. Петя говорит Васе два натуральных числа N и M ($N \leq M$), а Вася переставляет в числе N цифры так, чтобы получилось наибольшее число, не превосходящее M . Число N может являться результатом перестановки своих цифр. Записи чисел N и M , а также ответа не начинаются с цифры 0.*

Помогите Васе по заданным числам N и M найти наибольшее число, не превосходящее M , которое при этом состоит из всех цифр числа N (кроме ведущих нулей), записанных в некотором порядке.

Решение данной задачи выглядит следующим образом:

```
import itertools
n,m = input().split()
a=list(itertools.permutations(n))
ans=[]
m=int(m)
for x in a:
    k=int(''.join(x))
    if k<=m:
        ans.append(k)
print(max(ans))
```

Решение мы получаем, используя модуль *itertools* и функцию *permutations*, которая возвращает итератор с последовательными перестановками из элементов входной последовательности. Это позволяет получить очень лаконичное решение.

Для успешного выступления на разных этапах олимпиад требуется целенаправленная подготовка, которая должна обеспечить отличное владение необходимыми знаниями, работа по развитию таких качеств личности, как сила воли, стремлении к победе, трудолюбие. При подготовке основной целью считаем

не натаскивание на решение олимпиадных заданий, а достижение такой образовательной базы, в которой не будет пробелов.

Подготовка к олимпиаде требует отбора детей с определенным уровнем мотивации и устойчивым познавательным интересом. Каждый учитель начинает учебный год с поиска одаренных учащихся, которым предлагается участие в предметных олимпиадах: олимпиада – это рост уверенности учеников в своих силах и знаниях.

Индивидуальная работа (научно-исследовательская деятельность, самообразование) является еще одним методом работы с учащимися, обычно в формате консультаций, хотя иногда это совместный поиск решения какой-либо задачи. Как правило, это работа со старшеклассниками. В отличие от предыдущих форм работы они проводятся, как правило, эпизодически, поскольку организуются по мере возникновения запроса.

Особенность мыслительной деятельности старшеклассников – активизация теоретического мышления, стремление дойти до истины. В этом возрасте возникает потребность и способность к экспериментированию. В старших классах для развития гибкости мышления используются различные приемы: решить задачу несколькими способами, решить задачи-ловушки (с некорректными, недостающими или избыточными данными или не имеющие решения), в готовом алгоритме найти и исправить ошибки. Особое внимание уделяется оригинальности и оптимизации решения.

Интерес к изучению программирования у учеников можно вызвать различными творческими заданиями и проектами, например, участием в научно-исследовательской работе как способе углубить знания по определенной теме, увидеть связи между темами или практическое применение изучаемой темы.

Таким образом, программирование – одна из исторически первых компьютерных технологий обработки информации, актуальность которой с течением времени только возрастает. Владение языками программирования входит в набор компетенций, необходимых молодому человеку для социализации в современной цифровой среде, выполнения учебных и будущих профессиональных обязанностей, развития качеств цифровой личности. Одним из таких языков

является высокоуровневый язык программирования Python, адаптированный к решению разнообразных задач.

Python можно и нужно использовать в качестве базового языка обучения программированию в школе. Введение этого языка позволяет повысить интерес к изучению основ программирования. Даже за то ограниченное количество часов, которое выделяется на эту тему, гораздо большее число обучающихся осваивают тему «Алгоритмизация и программирование» и показывают хорошие результаты обучения по сравнению с обучением языку Pascal. И это важно, ведь в конечном итоге речь идет о формировании цифровых компетенций школьников, столь необходимых для успешной профессиональной самореализации в цифровую эпоху.

Список литературы

1. Зиннатуллин З.И. Использование языка программирования Python при решении криптоарифметических задач в средней школе / З.И. Зиннатуллин, О.М. Корчажкина // Информатика в школе. – 2023. – №4. – С. 67–69.
2. Кочеткова О.А. Обучение учащихся программированию на языке Python в рамках элективного курса по информатике / О.А. Кочеткова, Ю.Н. Пудовкина // Современные проблемы науки и образования. – 2019. – №2. – С. 59.
3. Каракозов, С.Д. Python как базовый язык обучения программированию в школе / С.Д. Каракозов, В.Г. Маняхина // Информатика в школе. – 2020. – №1. – С. 26–29.
4. Кириенко, Д. Почему Python должен быть первым языком программирования в школе? // Блог Д. Кириенко [Электронный ресурс]. – Режим доступа: <https://dkirienko.livejournal.com/167787.html>. – Дата доступа: 09.06.2024.
5. Ларсон, К. Какой язык программирования стоит выучить первым? [Электронный ресурс]. – Режим доступа: <https://m.habr.com/ru/company/ruvds/blog/315572>. – Дата доступа: 09.09.2024.
6. Маркелов, В.К. Язык программирования Python как альтернативный инструмент для решения заданий ЕГЭ по информатике / В.К. Маркелов, О.А. Завьялова // Информатика в школе. – 2023. – №2. – С. 63–67.

Приложение 1

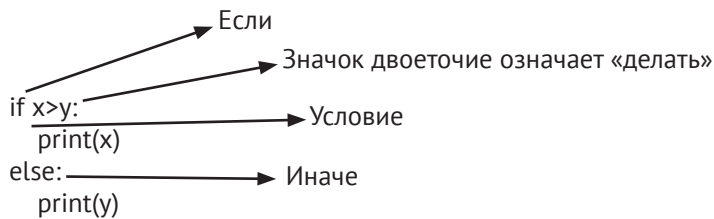
Команда ветвления

Рассмотрим задачу: «Даны два значения, x и y . Вывести большее из этих значений». Для ее решения нужно выполнять либо команду `print(x)` либо команду `print(y)`, то есть выбирать одно из нескольких действий.

Такую задачу поможет нам решить **команда ветвления IF**

```
x = int(input()) // вводим числа
```

```
y = int(input())
```



Словами эту структуру можно описать так: если значение переменной x больше значения переменной y , то делать команду **print(x)**, а если нет, то делать команду **print(y)**.

Рассмотрим примеры:

<p>Задача 1. У Маши – x яблок, а у Оли – y яблок. Вывести большее число яблок.</p>	<p>Задача 2. У Маши – x яблок, а у Оли – y яблок. Вывести имя девочки, у которой большее число яблок.</p>																
<p>Пример:</p> <table border="1"> <thead> <tr> <th>Ввод</th> <th>Вывод</th> </tr> </thead> <tbody> <tr> <td>4 5</td> <td>5</td> </tr> <tr> <td>6 2</td> <td>6</td> </tr> <tr> <td>7 7</td> <td>7</td> </tr> </tbody> </table>	Ввод	Вывод	4 5	5	6 2	6	7 7	7	<p>Пример:</p> <table border="1"> <thead> <tr> <th>Ввод</th> <th>Вывод</th> </tr> </thead> <tbody> <tr> <td>4 5</td> <td>Оля</td> </tr> <tr> <td>6 2</td> <td>Маша</td> </tr> <tr> <td>7 7</td> <td>Равно</td> </tr> </tbody> </table>	Ввод	Вывод	4 5	Оля	6 2	Маша	7 7	Равно
Ввод	Вывод																
4 5	5																
6 2	6																
7 7	7																
Ввод	Вывод																
4 5	Оля																
6 2	Маша																
7 7	Равно																
<p>Это случай, когда может быть только 2 варианта ответа. Используем IF (если) и ELSE (иначе)</p>	<p>Это случай, когда может быть несколько вариантов ответа. Используем IF (если), ELIF (иначе если) и ELSE (иначе)</p>																

Приложение 2

Нахождение НОД и НОК

Способ 1. Перебор всех возможных делителей.

1.1. Если используем обычные переменные

```
a,b = map(int,input().split())
k = min(a,b)
for x in range(1,k+1):
    if a%x==0 and b%x==0:
        ot=x
print(ot)
```

1.2. Если используем список

```
a,b = map(int,input().split())
k = min(a,b)
ot = []
for x in range(1,k+1):
    if a%x==0 and b%x==0:
        ot.append(x)
print(max(ot))
```

2. Алгоритм Евклида.

```
a = int(input())
b = int(input())
while a!=b:
    if a>b:
        a=a-b
    else:
        b=b-a
print(x)
```

Но! Плохо будет работать, если числа 1000000000 и 1.

3. Поэтому улучшаем:

```
a = int(input())
b = int(input())
while a!=0 and b!=0:
    if a>b:
        a=a%b
    else:
        b=b%a
print(a+b)
```

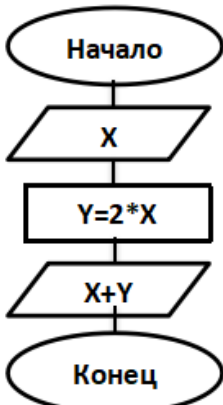
НОК Находим по формуле:

$$\text{НОК}(a,b) = \frac{a * b}{\text{НОД}(a,b)}$$

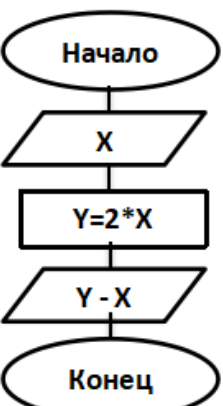
Приложение 3

Решение задач в системе программирования Python (4 класс)

Рассмотрим задачу: У Коли – X конфет, а у Пети – в 2 раза больше. Сколько всего конфет у ребят?
Для ее решения:

Составим блок- схему	Запишем блок-схему в виде программы	Пояснение
	<pre>X=int(input()) Y=2*X print(X+Y)</pre>	<p>Количество конфет у Коли – это то число, которое мы будем вводить (переменная X). Количество конфет у Пети будем вычислять (переменная Y). Ответ – это сумма переменных X и Y.</p>

У Коли – X конфет, а у Пети – в 2 раза больше. На сколько конфет у Пети больше?

Составим блок- схему	Запишем блок-схему в виде программы	Пояснение
	<pre>X=int(input()) Y=2*X print(Y- X)</pre>	<p>Эта программа отличается от предыдущей тем, что мы находим по смыслу другое значение</p>

Задача 1. Коля носит обувь 36 размера. Будет ли он носить ботинки размера X? (Вывести сообщение «Да» или «Нет»).

Составим блок-схему	Запишем блок-схему в виде программы
	<pre>X=int(input())</pre>

Задача 2. Термометр показал, что на улице N градусов. Программа должна вывести сообщение, какая температура на улице: положительная или отрицательная (меньше 0).

Составим блок-схему	Запишем блок-схему в виде программы	Пояснение
	<pre>N=int(input()) if N>0: print(' ') else: print(' ')</pre>	<p>Сначала вводим число N. При помощи команды if (если). Проверяем, значение N больше 0 или нет. Если значение N больше, то мы выводим, что температура 'Положительная', а если нет, то мы выводим сообщение 'Отрицательная'</p>

Задачи с шахматной доской

1. Даны 2 числа: x1,y1 – это координаты шахматной ладьи на поле. Затем еще 2 числа: x2,y2 – это координаты фигуры. Ответ на вопрос: поьет ли ладья фигуру на 1 ход?

Например:

Ввод	Вывод
2 2 4 2	Yes
2 2 5 7	No

Решение:

```
x1, y1, x2, y2=map(int,input().split())
if x1==x2 or y1==y2:
    print('YES')
else:
    print('NO')
```

2. Даны 2 числа: x_1, y_1 – это координаты шахматного слона на поле. Затем еще 2 числа: x_2, y_2 – это координаты фигуры. Ответ на вопрос: побьет ли ладья фигуру на 1 ход?

Например:

Ввод	Вывод
2 2 5 5	Yes
2 2 5 7	No

Решение:

```
x1, y1, x2, y2=map(int,input().split())
if max(x1, x2)-min(x1, x2)==max(y1, y2)-min(y1, y2):
    print('YES')
else:
    print('NO')
```

3. Даны координаты клетки на шахматном поле. Вывести, какая это клетка: черная или белая?

```
n=input()
x=n[0]
y=n[1]
y=int(n[1])
if x=='A':
    x=1
elif x=='B':
    x=2
elif x=='C':
    x=3
elif x=='D':
    x=4
elif x=='E':
    x=5
elif x=='F':
    x=6
elif x=='G':
    x=7
elif x=='H':
    x=8
if (x+y)%2==0:
    print('Чёрная')
else:
    print('Белая')
```

4. Даны координаты шахматного короля на поле. Сколько клеток он бьет в таком расположении?

Варианты решения.

Вариант 1.

```
k=input()
sp=['A','B','C','D','E','F','G','H']
x=sp.index(k[0])+1
y=int(k[1])
ot=0
if x+1>=1 and x+1<=8 and y+1>=1 and y+1<=8:
    ot=ot+1
if x+1>=1 and x+1<=8:
    ot=ot+1
if x+1>=1 and x+1<=8 and y-1>=1 and y-1<=8:
    ot=ot+1
if y-1>=1 and y-1<=8:
    ot=ot+1
if x-1>=1 and x-1<=8 and y-1>=1 and y-1<=8:
```

```
ot=ot+1
if x-1>=1 and x-1<=8:
    ot=ot+1
if x-1>=1 and x-1<=8 and y+1>=1 and y+1<=8:
    ot=ot+1
if y+1>=1 and y+1<=8:
    ot=ot+1
print(ot)
```

Вариант 2.

```
k=input()
sp=['A','B','C','D','E','F','G','H']
x=sp.index(k[0])+1
y=int(k[1])
ot=0
```



```
smx=[1,1,1,0,-1,-1,-1,0]
smy=[1,0,-1,-1,-1,0,1,1]
for z in range(8):
    if 1<=x+smx[z]<=8 and 1<=y+smy[z]<=8:
        ot=ot+1
print(ot)
```

Вариант 3.

```
k = input() #начальные координаты короля
if k[0]=='A':
    x=1
elif k[0]=='B':
    x=2
elif k[0]=='C':
    x=3
elif k[0]=='D':
    x=4
elif k[0]=='E':
    x=5
elif k[0]=='F':
    x=6
elif k[0]=='G':
    x=7
```

```
else:
    x=8
y=int(k[1])
ot=0
# мы проверяем все возможные варианты хода
короля
if x+1>=1 and x+1<=8 and y+1>=1 and y+1<=8:
    ot=ot+1
if x+1>=1 and x+1<=8:
    ot=ot+1
if x+1>=1 and x+1<=8 and y-1>=1 and y-1<=8:
    ot=ot+1
if y-1>=1 and y-1<=8:
    ot=ot+1
if x-1>=1 and x-1<=8 and y-1>=1 and y-1<=8:
    ot=ot+1
if x-1>=1 and x-1<=8:
    ot=ot+1
if x-1>=1 and x-1<=8 and y+1>=1 and y+1<=8:
    ot=ot+1
if y+1>=1 and y+1<=8:
    ot=ot+1
print(ot)
```

Дата поступления в редакцию: 25.06.2024

МЕТОД МОДЕЛИРОВАНИЯ НА УРОКАХ РУССКОГО ЯЗЫКА КАК ЭФФЕКТИВНОЕ СРЕДСТВО РАБОТЫ С ОДАРЕННЫМИ УЧАЩИМИСЯ

Литвинова Людмила Владимировна

Аннотация. Публикация посвящена описанию метода моделирования, который целесообразно применять на уроках языка на I ступени общего среднего образования при работе с одаренными учащимися для интенсификации развития их интеллектуальных способностей, формирования универсальных учебных действий.

Организация работы с одаренными учащимися в учебном процессе имеет ряд особенностей. Наряду с увеличением объема и темпа обучения большое значение имеет характер подачи учебного материала. Развитию креативности, интеллектуальных способностей, творческого мышления способствует использование метода учебного моделирования на уроках в начальной школе.

Теория учебного моделирования берет свое начало в работах Д.Б. Давыдова и А.Ч. Варданяна, З.Д. Гольдина. Научное обоснование этой теории дано в трудах П.Я. Гальперина. В концепциях названных ученых обучение и развитие предстают как система интенсивного всестороннего развития личности. Основной задачей является не передача знаний, а организация собственной деятельности учащихся по овладению способами анализа и обобщения учебного материала с помощью моделирования.

В научно-педагогических исследованиях под учебным моделированием понимают метод обучения, направленный на овладение учащимися учебным материалом с помощью учебных моделей понятий

и способов действий, а под учебной моделью – вид модели, которая применяется в обучении с целью выделения и исследования существенных сторон изучаемого объекта [2]. Преимуществом данного метода является то, что одаренный учащийся включается в активную мыслительную деятельность. Наиболее эффективной является такая организация учебной деятельности, которая приводит к самостоятельному построению учебной модели.

Уже с I класса учащиеся работают с фонетическими моделями слов, одна из которых представлена на схеме 1.

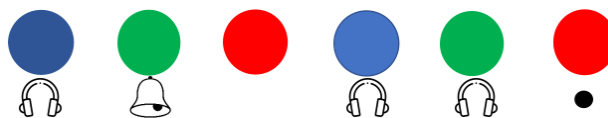


Схема 1. – Фонетическая модель слова

Схемы-модели используются для обозначения предложений. На схеме 2 представлена такая модель.

На первом этапе у учащихся формируется умение анализировать модели, наблюдать за разными спосо-