УДК 371:372.8
Doi: 10.26456/vtpsyped/2025.2.199

Реализация компетентностного подхода при решении заданий на рекурсию

В.С. Попов[1,2,3], Л.А. Абросимова-Романова[1]

[1] ФГБОУ ВО «Тверской государственный университет», г. Тверь
[2] ФГАОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)», г. Москва
[3] ГАОУ ВО «Московский городской педагогический университет», г. Москва

В статье представлены различные способы решения заданий при изучении рекурсии в рамках учебного предмета «Информатика» с позиций компетентностного подхода. Данные задания присутствуют в едином государственном экзамене по информатике под номером 16. Показано, что эффективности и результативности деятельности обучающихся способствует рассмотрение и применение различных методов, методик и инструментов решения заданий, формирование компетенций в их совокупности, в том числе личностных, универсальных, цифровых, математических, языковых компетенций, а также декомпозиция компетенций по уровням сложности учебного материала и компонентам.
Ключевые слова: *педагогический подход, компетентностный подход, системно-деятельностный подход, компетенция, компетентность, рекурсия, рекурсивная функция, циклический алгоритм, программирование, проверка.*

Введение

Современный мир предъявляет новые, повышенные требования к выпускникам образовательных организаций. Процессы глобализации и регионализации, разделения труда и специализации, существенных технологических изменений, а также вызовы современного мира и глобальные проблемы человечества являются причинами повышенных требований к компетентности обучающихся и работников.

Федеральные государственные образовательные стандарты (ФГОС), в том числе на уровнях основного общего (ООО) и среднего общего образования (СОО), в числе прочих также содержат требования к результатам образования. Внедрение ФГОС в педагогический процесс привело к актуализации педагогического понятия «подход» как «сложного многоуровневого инструментария, призванного помочь разрешить какую-либо глобальную образовательную проблему» [9, с. 45]. В качестве ведущего подхода для обучения в соответствии с ФГОС разработчиками стандарта был выбран системно-деятельностный подход [11]. В частности, в ФГОС указано, что системно-деятельностный подход

© Попов В.С., Абросимова-Романова Л.А., 2025

является его методологической основой, обеспечивает формирование готовности к образовательной деятельности и непосредственно персонализированную деятельность обучающихся в рамках педагогического процесса, а также создание развивающей среды образовательной организации [11]. Реализация федеральных рабочих программ (ФРП) как составной части федеральных основных образовательных программ (ФООП) в школах с 1 сентября 2023 года является одним из процессов, реактуализирующих компетентностный подход в педагогике. В частности, федеральные рабочие программы по информатике декларируют компетентностный подход в качестве основного образовательного подхода при обучении информатике [10]. Инструмент анализа больших текстовых данных Google Books Ngram Viewer подтверждает значительное увеличение частоты упоминания N-грамм «деятельностный подход», «компетентностный подход» и особенно «системно-деятельностный подход» и «метапредметный подход» (рис. 1) в период активного внедрения ФГОС в российских школах (2010–2015 гг.). Анализ частоты упоминания N-грамм показывает, что две последние N-граммы – «системно-деятельностный подход» и «метапредметный подход» – появились в образовательном дискурсе с первой половины 2010-х годов.

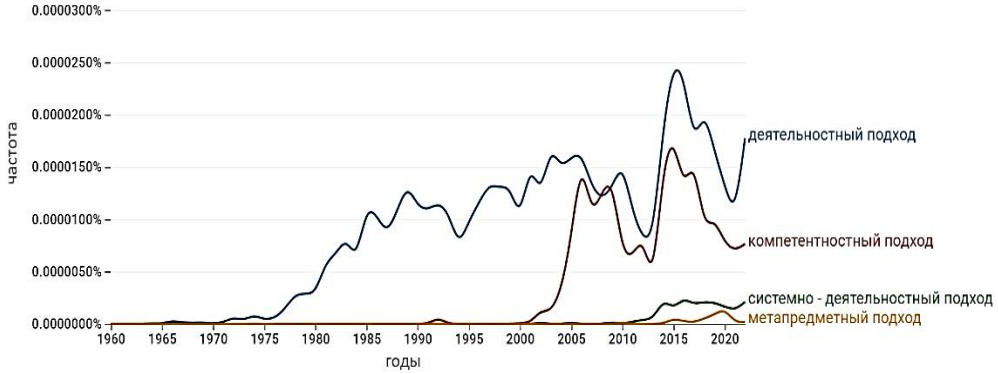

Рис. 1. Упоминания некоторых образовательных подходов в русскоязычной литературе за период 1960-2022 гг. По оси ординат указана частота N-граммы в корпусе литературы на русском языке за заданный год

Исследования ряда авторов подтверждают общие характеристики системно-деятельностного и компетентностного подходов. Компетентностный подход есть развитый в истории отечественной психолого-педагогической науки системно-деятельностный подход в образовании [6, с. 184]. Для системно-деятельностного подхода справедлива формула: компетенция → деятельность → компетентность [6, с. 185]: чтобы конкретная компетенция была присвоена личностью, она должна быть задействована в деятельности данного субъекта. Подчёркивается взаимодействие, взаимосвязанность и взаимозависимость многих современных подходов в образовании [6, с. 188].

И.А. Зимняя выделяла компетенции обучающихся в качестве «новой парадигмы результата образования» [7, с. 7–8]. Действительно, одним из главных принципов в компетентностном подходе является развитие и формирование полноценной компетентности учеников как субъектов педагогического процесса. Другим важным принципом компетентностного подхода является формирование активного, деятельностного человека и гражданина, для чего в фокусе внимания оказываются сложные вопросы мотивации и готовности субъектов деятельности. И.А. Зимняя понимала компетенции и компетентность как «всё, что способствует результативности деятельности» [8].

Одним из основных выводов международного доклада «Универсальные компетентности и новая грамотность: от лозунгов к реальности» является тезис о том, что «ключевым вопросом развития универсальных компетентностей является вопрос соотношения с предметными (дисциплинарными) знаниями и навыками» [12, с. 20]. Многочисленные примеры тесного взаимодействия и взаимозависимости компетенций и предметных знаний приведены в книге М.С. Добряковой «Ученик выигрывает: знания, грамотности и компетентности в школе» [4].

В данном исследовании показан способ организации содержания образования и формирования компетенций обучающихся при решении заданий на рекурсию учебного предмета «Информатика» на основе компетентностного подхода. На примере различных методов решения рекурсивного задания формата ЕГЭ по информатике показано, что результативности и эффективности учебной деятельности способствуют изучение различных и разнообразных методов, методик и инструментов для решения заданий; развитие личностных качеств и универсальных компетенций обучающегося; классификация содержания образования по отношению к уровням сложности (базовое, углублённое), компонентам компетенций (знания, навыки, ценностные отношения); взаимосвязанное развитие компетенций различных групп (ключевых компетенций) в их совокупности.

Основная часть

Далее в статье рассмотрены аналитические решения рекурсивных заданий путём подстановки значений аргументов в уравнения функций, рекурсивные и циклические решения на языке программирования, решения в электронных таблицах.

Пример № 1. Задание 16 демонстрационных версий ЕГЭ по информатике 2021 и 2022 гг. [1, с. 14, 2, с. 14].

Формулировка задания: «Алгоритм вычисления значения функции F(n), где n – натуральное число, задан следующими соотношениями:

$F(n) = 1$ при $n = 1$;

$F(n) = n + F(n - 1)$, если n чётно,

$F(n) = 2 \times F(n - 2)$, если $n > 1$ и при этом n нечётно.

Чему равно значение функции F(26)?»

Аналитический метод решения задания состоит в том, что обучающемуся требуется записать формулы для прямого и обратного хода рекурсии (табл. 1). Ответом на предложенное задание является значение функции F(26), равное 4122.

Таблица 1

Прямой и обратный ход аналитического решения

Прямой ход рекурсии	Обратный ход рекурсии
F(26) = 26 + F(25)	F(26) = 26 + F(25) = 26 + 4096 = 4122
F(25) = 2 × F(23)	F(25) = 2 × F(23) = 2 × 2048 = 4096
F(23) = 2 × F(21)	F(23) = 2 × F(21) = 2 × 1024 = 2048
F(21) = 2 × F(19)	F(21) = 2 × F(19) = 2 × 512 = 1024
…	…
F(7) = 2 × F(5)	F(7) = 2 × F(5) = 2 × 4 = 8
F(5) = 2 × F(3)	F(5) = 2 × F(3) = 2 × 2 = 4
F(3) = 2 × F(1)	F(3) = 2 × F(1) = 2 × 1 = 2
F(1) = 1	F(1) = 1

Рекурсивный способ решения задания на языке программирования Python представлен на рис. 2.

```
1 def F(n):
2     if n == 1:
3         return 1
4     elif n % 2 == 0:
5         return n + F(n - 1)
6     elif n > 1 and n % 2 == 1:
7         return 2 * F(n - 2)
8
9 print(F(26))
```

Рис. 2. Рекурсивный способ решения задания на языке программирования Python

В вышеприведённом (рис. 2) программном коде определена функция F(n), каждое условие заданной системы уравнений записывается после ключевых слов if/elif, после ключевого слова return записываются правые части рекуррентных формул или константы в случае записи базы рекурсии (рис. 3). На рис. 3 подчёркнуты условия, дополнительно выделены правые части рекуррентной формулы. Таким образом, обучающиеся преобразуют заданную математическую нотацию в программный код на языке программирования, формируя не только цифровые и математические компетенции, но также – языковые компетенции. Следует подчеркнуть, что кодированием считается операция преобразования знаков или их групп одной знаковой системы в знаки или группы знаков другой знаковой системы, что относится как к записи программного кода по заданной рекуррентной формуле, так и к переводу текста с одного языка на другой, что подчёркивает необходимость формирования компетенций в их совокупности, в данном примере – цифровых, математических и языковых компетенций.

Приведённый на рис. 2 рекурсивный алгоритм может быть заменён на циклический (итеративный), соответствующая программа представлена на рис. 4.

$$F(n) = \begin{cases} 1 & if\ n = 1 \\ n + F(n-1) & if\ n\ is\ even \\ 2 \times F(n-2) & if\ n > 1\ and\ n\ is\ odd \end{cases}$$

```
def F(n):
    if n == 1:
        return 1
    elif n % 2 == 0:
        return n + F(n - 1)
    elif n > 1 and n % 2 == 1:
        return 2 * F(n - 2)

print(F(26))
```

Рис. 3. Преобразование математической нотации рекуррентной формулы в функцию на языке программирования Python

Сохранение вычисленных значений функции F(n) выполняется в списке lst, так что F(n) = lst[n]. В соответствии с заданной математической нотацией, lst[1] = 1, lst[n] = n + lst[n – 1] для чётных n, lst[n] = 2 * lst[n – 2] для нечётных n, больших 1. Для удобства работы со списком lst принято, что lst[0] = 0. Как будет показано ниже в примерах 2 и 3, способности перевода рекурсивных вычислений в циклический вид значительно облегчают решение заданий в случае необходимости вычисления результата для больших аргументов рекурсивной функции или в случае параллельной рекурсии, когда имеется несколько рекурсивных вызовов в рамках одной рекурсивной ветви.

```
lst = [0, 1]
n = 2
while n <= 26:
    if n % 2 == 0:
        lst.append(n + lst[n - 1])
    else:
        lst.append(2 * lst[n - 2])
    n += 1

print(lst[26])
```

Рис. 4. Циклический способ решения задания на языке программирования Python

Пример № 2. Задание 16 демонстрационной версии ЕГЭ по информатике 2023 г. [3, с. 14].

Формулировка задания: «Алгоритм вычисления значения функции F(n), где n – натуральное число, задан следующими соотношениями:

F(n) = 1 при n = 1;

F(n) = n × F(n − 1), если n > 1.

Чему равно значение выражения F(2023) / F(2020)?»

Ниже представлен аналитический метод решения данного задания:

F(2023) = 2023 × F(2022)

F(2022) = 2022 × F(2021)

F(2021) = 2021 × F(2020)

Подставим полученные правые части рекуррентных уравнений в вышестоящие уравнения:

F(2022) = 2022 × F(2021) = 2022 × 2021 × F(2020)

F(2023) = 2023 × F(2022) = 2023 × 2022 × 2021 × F(2020)

F(2023) / F(2020) = (2023 × 2022 × 2021 × F(2020)) / F(2020) = 2023 × 2022 × 2021 = 8266912626

Ответом на задание является число 8266912626.

Традиционное решение с помощью записи рекурсивной функции на языке программирования претерпевает изменения: для решения данного задания ввиду больших аргументов рекурсивных функций по сравнению с базой рекурсии обучающимся требуется увеличить глубину стека вызовов, используя функцию sys.setrecursionlimit(limit) (рис. 5). Следует отметить, что далеко не все обучающиеся знакомы с понятием стека вызовов и методами увеличения глубины стека.

```
1  import sys
2  sys.setrecursionlimit(3000)
3
4  def F(n):
5      if n == 1:
6          return 1
7      else:
8          return n * F(n - 1)
9
10 print(F(2023) / F(2020))
```

Рис. 5. Рекурсивный способ решения задания на языке программирования Python и увеличение глубины стека

Однако обучающиеся, способные к представлению рекурсивных вычислений в циклическом виде, могут не использовать увеличение глубины стека вызовов и решить данное задание, используя аналогичный циклический программный код (рис. 6).

```
1 lst = [0, 1]
2 n = 2
3 while n <= 2023:
4     lst.append(n * lst[n - 1])
5     n += 1
6
7 print(lst[2023] / lst[2020])
```

Рис. 6. Циклический способ решения задания на языке программирования Python

Пример № 3. Вычисление n-го числа пентаначчи.

Пример задания: «Последовательность чисел пентаначчи F(n), где n – натуральное число, задана линейным рекуррентным соотношением:

F(n) = 0 при $0 \leq n \leq 3$;

F(n) = 1 при n = 4;

F(n) = F(n − 1) + F(n − 2) + F(n − 3) + F(n − 4) + F(n − 5), если n > 4.

Чему равно значение функции F(37)?»

Традиционное рекурсивное решение задания на языке программирования не способно вычислить ответ за адекватное время ввиду отсутствия мемоизации [5] и наличия нескольких рекурсивных вызовов при вычислении всех значений F(n) при n > 4 (рис. 7).

```
def F(n):
    if 0 <= n <= 3:
        return 0
    if n == 4:
        return 1
    if n > 4:
        return F(n-1)+F(n-2)+F(n-3)+F(n-4)+F(n-5)

print(F(37))
```

Рис. 7. Рекурсивная функция вычисления n-го числа ряда пентаначчи

В решении, приведённом на рис. 7, значение для каждого F(n) при $5 \leq n \leq 36$ будет вычисляться многократно. Как было замечено выше, главной причиной крайне медленного вычисления ответа программой является отсутствие мемоизации – сохранения уже вычисленных значений функции. Решение на языке программирования Python с использованием мемоизации приведено на рис. 8. Хотя приёмы мемоизации в языке программирования Python не являются слишком сложными или громоздкими для их использования обучающимися, данное предметное содержание относится к углублённому уровню изучения учебного предмета «Информатика».

Программа выведет ответ на задание – число 2621810068.

Циклический способ решения задачи нахождения 37-го числа пентаначчи приведён на рис. 9.

```
from functools import lru_cache
@lru_cache(None)

def F(n):
    if 0 <= n <= 3:
        return 0
    if n == 4:
        return 1
    if n > 4:
        return F(n-1)+F(n-2)+F(n-3)+F(n-4)+F(n-5)

print(F(37))
```

Рис. 8. Использование мемоизации при вычислении рекурсивных функций

```
lst = [0, 0, 0, 0, 1]
n = 5
while n <= 37:
    lst.append(lst[n-1]+lst[n-2]+lst[n-3]+lst[n-4]+lst[n-5])
    n += 1

print(lst[37])
```

Рис. 9. Циклический способ решения задания на языке программирования Python

Другим эффективным способом решения задания нахождения 37-го числа в последовательности пентаначчи является использование электронных таблиц. На рис. 10 показаны n, F(n) и соответствующие формулы электронной таблицы.

n	F(n)	n	F(n)	n	F(n)	n	F(n)
0	0	19	13624	0	0	19	=СУММ(B16:B20)
1	0	20	26784	1	0	20	=СУММ(B17:B21)
2	0	21	52656	2	0	21	=СУММ(B18:B22)
3	0	22	103519	3	0	22	=СУММ(B19:B23)
4	1	23	203513	4	1	23	=СУММ(B20:B24)
5	1	24	400096	5	=СУММ(B2:B6)	24	=СУММ(B21:B25)
6	2	25	786568	6	=СУММ(B3:B7)	25	=СУММ(B22:B26)
7	4	26	1546352	7	=СУММ(B4:B8)	26	=СУММ(B23:B27)
8	8	27	3040048	8	=СУММ(B5:B9)	27	=СУММ(B24:B28)
9	16	28	5976577	9	=СУММ(B6:B10)	28	=СУММ(B25:B29)
10	31	29	11749641	10	=СУММ(B7:B11)	29	=СУММ(B26:B30)
11	61	30	23099186	11	=СУММ(B8:B12)	30	=СУММ(B27:B31)
12	120	31	45411804	12	=СУММ(B9:B13)	31	=СУММ(B28:B32)
13	236	32	89277256	13	=СУММ(B10:B14)	32	=СУММ(B29:B33)
14	464	33	175514464	14	=СУММ(B11:B15)	33	=СУММ(B30:B34)
15	912	34	345052351	15	=СУММ(B12:B16)	34	=СУММ(B31:B35)
16	1793	35	678355061	16	=СУММ(B13:B17)	35	=СУММ(B32:B36)
17	3525	36	1333610936	17	=СУММ(B14:B18)	36	=СУММ(B33:B37)
18	6930	37	2621810068	18	=СУММ(B15:B19)	37	=СУММ(B34:B38)

Рис. 10. Итеративный способ решения задания в электронных таблицах

Приведённые решения примеров 2 и 3 демонстрируют, что многообразие рассмотренных методов, методик и инструментов для решения предлагаемых заданий на рекурсию позволяет обучающимся выполнять задания высокого уровня сложности, выбирая простые и эффективные методы решения. При знакомстве с рекурсией на углублённом уровне изучения учебного предмета «Информатика» обучающиеся могут получить представления и сформировать навыки изменения глубины стека вызовов и использования мемоизации, что позволит решать большое количество заданий при помощи обычной рекурсивной функции, записанной на языке программирования. Также к учебному содержанию в рамках обучения рекурсии и решения рекурсивных задач на углублённом уровне изучения информатики можно отнести содержание, не рассмотренное в данной статье, например, рекурсию высокого порядка, реализацию и виды мемоизации, рекурсивные структуры и типы.

Компонентная структура компетенций включает знания, навыки, ценностные отношения [12, с. 42]. Педагогам и обучающимся следует

уметь анализировать содержание образования в рамках изучаемой темы и соответствующих элементов содержания. Ниже приведены отдельные компоненты компетенций при изучении элемента содержания «Рекурсия» учебного предмета «Информатика».

Знания: понятие рекурсии, простая рекурсия, сложная (косвенная) рекурсия, рекурсивная функция, прямой и обратный ход рекурсии, база рекурсии, рекуррентная формула, терминальная и рекурсивная ветви, мемоизация, стек вызовов, отдельные виды рекурсивных функций (факториал, Фибоначчи, Ханойская башня и др.) и алгоритмов (обход дерева, алгоритмы динамического программирования [13]), структуры, синтаксис и семантика языка программирования для определения рекурсивных функций.

Навыки: написание рекурсивной функции на языке программирования, запись рекуррентной формулы, подстановка аргумента в рекуррентную формулу, преобразование рекурсивного алгоритма в циклический, использование мемоизации, увеличение глубины стека вызовов, использование функциональности электронных таблиц, в том числе формул.

Ценностные отношения: способность к проверке созданного решения и полученного результата с помощью различных методов и инструментов, понимание принципа информационной избыточности и концепции эффективности вычислений, способность выбора наиболее эффективных методов и инструментов решения задач, творческое применение знаний рекурсии в задачах реальной жизни и других предметных областях.

Вместе с изучением рассмотренного содержания происходит развитие личностных качеств: внимательности, аккуратности, гибкости, настойчивости, креативности, ориентации на результат; формирование универсальных и личностных компетенций обучающихся: мышления, способности взаимодействия с компьютером.

Заключение

Рассмотрены разнообразные решения трёх приведённых примеров заданий различного уровня сложности при изучении элемента содержания «Рекурсия» учебного предмета «Информатика». В данной работе представлено, каким образом изучение предметного содержания на примере рекурсии ведёт к формированию компетенций обучающихся, и как педагоги могут применять компетентностный подход для организации содержания образования в рамках изучения элементов содержания. Показано, что эффективности и результативности деятельности обучающихся способствует знакомство с широким набором методов, методик, инструментов решения учебных заданий. Формирование компетенций в их совокупности, развитие личностных качеств, декомпозиция содержания образования по компонентам

компетенций и уровням сложности способствуют более полному усвоению содержания образования и улучшению образовательных результатов обучающихся.

Список литературы

1. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2022 года по информатике. М.: Федеральный институт педагогических измерений, 2021. 22 с.
2. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2023 года по информатике. М.: Федеральный институт педагогических измерений, 2022. 22 с.
3. Демонстрационный вариант контрольных измерительных материалов единого государственного экзамена 2024 года по информатике. М.: Федеральный институт педагогических измерений, 2023. 24 с.
4. Добрякова М.С. Ученик выигрывает: знания, грамотности и компетентности в школе: Методическое пособие по разработке содержания образования с учетом формирования универсальных компетентностей и новой грамотности – «мягких навыков». Москва: Национальный исследовательский университет «Высшая школа экономики», 2023. 240 с. ISBN 978-5-7598-2954-6. DOI: 10.17323/978-5-7598-2954-6. EDN: NKYJPJ.
5. Долинский М.С. Ускорение рекурсивных решений при помощи мемоизации // Информатика в школе. 2022. № 6 (179). С. 55-67. DOI: 10.32517/2221-1993-2022-21-6-55-67. EDN: PSWCPK.
6. Жданко Т.А. Системно-деятельностный подход: сущностная характеристика и принципы реализации // Научно-педагогический журнал Восточной Сибири Magister Dixit. 2012. № 4. С. 183–189. EDN: PJRCNX.
7. Зимняя И.А. Ключевые компетенции – новая парадигма результата образования // Эксперимент и инновации в школе. 2009. № 2. С. 7–14.
8. Зимняя И.А. Компетентностный подход в образовании [Электронный ресурс]. // Психологический институт РАО. URL: https://www.youtube.com/watch?v=aomPknCyiKg (дата обращения: 15.01.2025)
9. Кудаков О.Р. «Подход» и «Подходность» в современных педагогических исследованиях // Казанский педагогический журнал. 2013. № 2(97). С. 32–49.
10. Попов В.С., Алефиренко Е.А., Черницына Л.Ю. Компетентностный подход в федеральных рабочих программах по информатике на уровне среднего общего образования // Текущие вызовы в подготовке кадров. Обучение специалистов по современным направлениям информационных технологий, кибербезопасности и ИКТ-электроники, актуальным для экономики данных: сборник научных трудов. Тверь: Издательство Тверского государственного университета, 2024. С. 604–607.
11. Приказ Минобрнауки России от 17 мая 2012 г. № 413 «Об утверждении федерального государственного образовательного стандарта среднего общего образования». Министерство образования и науки Российской Федерации. 2012. Зарегистрировано в Минюсте России 7 июня 2012 г. № 24480. URL: https://clck.ru/3GfDbW

12. Универсальные компетентности и новая грамотность: от лозунгов к реальности: монография / под ред. М.С. Добряковой, И.Д. Фрумина; при участии К.А. Баранникова, Н. Зиила, Дж. Мосс, И.М. Реморенко, Я. Хаутамяки. Москва: Национальный исследовательский университет «Высшая школа экономики», 2020. 472 с. DOI: 10.17323/978-5-7598-2177-9. EDN: ADLXUC.
13. Popov V. S. Dynamic Programming in Tasks of the Russian Unified State Exam in Computer Science // 2024 6th International Youth Conference on Radio Electronics, Electrical and Power Engineering (REEPE), Moscow, Russian Federation, 2024. P. 1-6. DOI: 10.1109/REEPE60449.2024.10479768.

Об авторах:

ПОПОВ Владислав Сергеевич – аспирант ФГБОУ ВО «Тверской государственный университет» (170100, г. Тверь, ул. Желябова, д. 33); старший преподаватель кафедры «Информационные системы и телекоммуникации» ФГАОУ ВО «Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)» (105005, г. Москва, ул. 2-я Бауманская, д. 5, стр. 1); эксперт Института развития профильного обучения ГАОУ ВО «Московский городской педагогический университет» (129226, г. Москва, 2-й Сельскохозяйственный проезд, д. 4, корп. 1); e-mail: popov_vlad@mail.ru

АБРОСИМОВА-РОМАНОВА Лариса Алексеевна – доктор филологических наук, профессор кафедры «Социальная работа и педагогика» ФГБОУ ВО «Тверской государственный университет» (170100, г. Тверь, ул. Желябова, д. 33); e-mail: larrar@yandex.ru

Implementation of a competence-based approach in solving recursion tasks

V.S. Popov[1, 2, 3], L.A. Abrosimova-Romanova[1]

[1]Tver State University, Tver, Russia
[2]Bauman Moscow State Technical University, Moscow, Russia
[3]Moscow City University, Moscow, Russia

The article discusses the study of various methods for solving recursion tasks of the subject «Informatics» from the standpoint of the competence-based approach. These tasks are present in the unified state exam in informatics (number 16). It is shown that the effectiveness of students' activities is facilitated by the consideration and application of various methods, techniques and tools for solving tasks, the formation of a set of competencies, including personal, universal, digital, mathematical, language competencies, as well as the decomposition of competencies by levels of complexity of educational content and its components.
Keywords: *pedagogical approach, competence-based approach, system-activity approach, competence, competency, recursion, recursion function, cyclic algorithm, programming, verification.*

Принято в редакцию: 29.08.2024 г.
Подписано в печать: 05.11.2024 г.