УДК 37:004

СТРАТЕГИЯ ПРЕЕМСТВЕННОГО ПЕРЕХОДА В ШКОЛЬНОЙ ИНФОРМАТИКЕ ОТ КОМПЬЮТЕРНОГО ИСПОЛНИТЕЛЯ *ЧЕРТЕЖНИК* К КОМПЬЮТЕРНОМУ ИСПОЛНИТЕЛЮ *ЧЕРЕПАХА*

С.И. Зенько

Учреждение образования «Белорусский государственный педагогический университет имени Максима Танка», Беларусь.

Аннотация. Выбор языка программирования, на основе которого продолжается формирование базовых понятий алгоритмизации у учащихся 6-х классов и начинается реализация процесса обучения их основам программирования, – актуальный вопрос и сегодня. В новом издании учебного пособия «Информатика» (6 класс) в качестве языка программирования рассматривается *Python*. В качестве средства обучения выступает компьютерный исполнитель *Черепаха*. В Республике Беларусь накоплен достаточный опыт для обучения учащихся на основе языка программирования *PascalABC.NET* (с компьютерным исполнителем *Чертежник*). В статье представлена стратегия преемственного перехода между компьютерными исполнителями. Обоснована и предложена оптимизированная подсистема команд компьютерного исполнителя *Черепаха* для изучения начал программирования. Приведены поурочные методические рекомендации для учителей информатики и примеры заданий, используя которые можно реализовать изучение темы "Алгоритмы и исполнители" с учащимися 6-го класса в соответствии с действующей учебной программой по информатике.

Ключевые слова. Дидактика информатики, школьная информатика, методика обучения началам программирования, компьютерный исполнитель *Черепаха*, компьютерный исполнитель *Чертежник*, язык программирования *Python*, деятельностно-семантический подход, когнитивный подход.

STRATEGY OF SUCCESSFUL TRANSITION IN SCHOOL INFORMATICS FROM COMPUTER PERFORMER *DRAWMAN* TO COMPUTER PERFORMER *TURTLE*

S. Zenko

Educational institution "Belarusian State Pedagogical University named after Maxim Tank", Belarus

Abstract. The choice of a programming language, on the basis of which the formation of basic concepts of algorithmization among 6th grade pupils continues and the implementation of the process of teaching them the basics of programming begins, is an urgent issue today. In the new edition of the textbook "Informatics" (6th grade), *Python* is considered as a programming language. The *Turtle* performer is used as a computer tool. The Republic of Belarus has accumulated sufficient experience in teaching schoolchildren based on the programming language *PascalABC.NET* (with a computer performer *Drawman*). The article presents a strategy for the succession transition between computer performers. An optimized subsystem of commands for learning the basics of programming with the help of a *Turtle* computer performer is substantiated and proposed. The article provides instructional guidelines for computer science teachers and examples of tasks that can be used to implement the study of the topic "Algorithms and performers" with 6th grade pupils in accordance with the current informatics curriculum.

Keywords. Didactics of computer science, school informatics, methodology for teaching basic programming, computer performer *Drawman*, computer performer *Turtle*, programming language *Python*, activity-semantic approach, cognitive approach.

Введение

В 2024 г. было выпущено второе издание учебного пособия «Информатика» для учащихся 6 класса [8]. В нем пересмотрено изложение учебного материала по отдельным темам. Ряд тем были дополнены. С методической точки зрения особое внимание к себе привлекает тема «Алгоритмы и исполнители». Это связано с тем, что в таких параграфах как «Среда программирования и компьютерный исполнитель», «Изучение и изменение готовых программ» и «Составление программ. Использование подпрограмм (вспомогательных алгоритмов)» вместо среды программирования *PascalABC.NET* [10, С. 121–122] предложено рассматривать интегрированную среду разработки и обучения *IDLE* (*Python* 3.11 64-bit) [8, С. 130–131] и, соответственно, учебный материал представлять для компьютерного исполнителя *Черепаха* [8, С. 133–161] вместо компьютерного исполнителя *Чертежник* [10, С. 122–140].

Переход от языка программирования *PascalABC.NET* к языку программирования *Python* в учебном процессе можно рассматривать как перспективную и оправданную тенденцию. Она естественным эволюционным образам при развитии технологий программирования и языков программирования наблюдается при модернизации и

совершенствовании представления учебного контента в разных странах ([3], [6], [12], [16]). Однако с этим становятся остро актуальными вопросы адаптации методики обучения учащихся основам алгоритмизации с учетом тех особенностей и условий, которые требуется соблюдать при переходе к новой среде разработке и при обучения с помощью нового компьютерного исполнителя.

Важно также отметить, что переосмысление подходов к изложению учебного материала и выстраивание обновленной стратегии в методике обучению учащихся в 6 классе основам алгоритмизации и программирования влекут за собой изменения как в деятельности учителей информатики, которые на протяжении достаточного времени уже преподают эти темы, так и при методической подготовке студентов – будущих учителей информатики в педагогических университетах Республики Беларусь. Вместе с этим потребуется пересмотр и межпредметных взаимосвязей при формировании общеучебных понятий [5] учебного предмета «Информатика» и таких учебных предметов как «Математика» и «Физика».

Методы и материалы исследования

При осуществлении исследования мы опирались на идеи деятельностной, семантической и когнитивной концепций философии, психологии и педагогики, а также на концепцию использования деятельностно-семантического подхода [4] в процессе конструирования и реализации методической системы подготовки учителей информатики.

В качестве материалов исследования выступают ***научно-методические*** (Н. М. Аусиловой [3], И. В. Барышевой [12], Л. Л. Босовой [1], Ю. А. Быкадорова [2], Д. Ю. Гришкова [3], А. С. Зуфаровой [6], О. А. Козлова [12], Е. В. Малкиной [12], В. А. Мишина [1], Н. Н. Самылкиной [1], Н. Э. Стоянова [16],Р. А. Суходуба [6], В. Н. Тарасовой [16], О. В. Шепелюка [2], Н. В. Шестаковой [12]) ***и учебно-дидактические*** (Е. Н. Войтехович [7–10], П. Л. Гращенко [14], В. М. Котова [8], А. И. Лапо [7–10, 13, 14], Н. П. Макаровой [8, 10, 13], А. Н. Мороз [9], Л. Г. Овчинниковой [11], С. Г. Пузиновской [11], А. Е. Пупцева [13, 14]) ***разработки***, позволяющие провести сравнительный анализ подходов к осуществлению процесса формирования у учащихся основных понятий алгоритмизации и начал программирования средствами *компьютерных исполнителей Чертежник* и *Черепаха*. Кроме этого, указанные разработки изучаются и в контексте выявления обоснованной последовательности представления учащимся подсистемы системы команд компьютерного исполнителя *Черепаха* с целью вырабатывания стратегии преемственного перехода в методике обучения учащихся началам

программирования средствами компьютерных исполнителей в рамках таких языков программирования как *PascalABC.NET* и *Python*.

Компьютерный исполнитель *ЧЕРЕПАХА* и методические вопросы выделения учебной подсистемы его команд

Общеизвестно, что система команд компьютерного исполнителя *Чертежник* включает пять команд. Это команды *поднять перо, опустить перо, сместиться в точку, сместиться на вектор* и *команда настройки размеров окна*, в котором осуществляется построения изображения компьютерным исполнителем. Анализируя представление учебного материала в отечественных учебных пособиях по информатике разных поколений ([10], [13], [14]), убеждаемся, что при небольшом количестве команд эффективность изучения темы «Алгоритмы и исполнителе» вместе с тем достигалась в процессе последовательного использования разных подсистем системы (табл. 1) команд компьютерного исполнителя *Чертежник*.

Таблица 1. – Последовательность расширения набора рассматриваемых с учащимися команд компьютерного исполнителя *Чертежник*

Начальный набор команд	Промежуточный набор команд	Окончательный набор команд
uses Drawman; PenUp; PenDown; ToPoint(*x,y*);	uses Drawman; **Field(*N,M*);**[1] PenUp; PenDown; ToPoint(*x,y*);	uses Drawman; **procedure *name;*** Field(*N,M*); PenUp; PenDown; ToPoint(x,y); **OnVector(a,b);**

Как показала практика обучения учащихся в 6 классах началам программирования средствами компьютерного исполнителя *Чертежник* может быть реализовано результативно за счет того, что небольшой набор команд достаточно быстро запоминается учащимися и далее сложности могли возникать только с пониманием сущности работы отдельных команд (в частности, команды сместиться на вектор), либо с ходом построения самого алгоритма.

Анализируя систему команд компьютерного исполнителя *Черепаха*, легко заметить, что она включает существенно больше команд. Конечно, с одной стороны, это позволяет расширить круг решаемых задач по созданию изображений с помощью

[1] Полужирным выделены те команды, за счет которых расширяется система команд компьютерного исполнителя на соответствующем этапе обучения учащихся основам алгоритмизации и началам программирования.

компьютерного исполнителя. Например, сделать изображения цветными. Однако, в свою очередь, наличие большого перечня команд затрудняет их представление и удержание в памяти учащихся (не добавляет легкости этому и имеющаяся справочная система, так как, во-первых, она представлена только на английском языке и требует настройки автоперевода при обращении к ней; во-вторых, переводной вариант также не в полной мере отвечает возрасту шестиклассников. Помимо этого, следует отметить, что не все представленные в учебном пособии команды (в частности, *towards(x,y)* и *distance(x,y)* [8, С. 140]), нашли свое развитие и реализацию в система учебных примеров и заданий.

Для реализации стратегии преемственного перехода в школьной информатике к новому компьютерному исполнителю считаем необходимым провести определенную ***оптимизацию рассматриваемой подсистемы команд компьютерного исполнителя Черепаха для учебных целей и представить последовательность их использования и расширения исходя из логики формирования основных понятий алгоритмизации и начал программирования средствами компьютерного исполнителя.*** В таблице (табл. 2) к каждому уроку представлены базовый и дополнительный наборы команд. Базовый набор команд нами рассматривается как обязательный для изучения и использования учащимися на указанном уроке. Дополнительный набор команд может быть рассмотрен учителем с теми учащимися, которые последовательно и непрерывно демонстрируют высокие результаты учебной деятельности. Название тем уроков приведены в соответствии с примерным календарно-тематическим планированием [7, С. 9–10].

Таблица 2. – Последовательность расширения набора рассматриваемых с учащимися команд компьютерного исполнителя *Черепаха*

Тема урока	Среда программирования
Базовый набор команд	*Дополнительный набор команд и знаков*
import turtle turtle.shape('*name form*') turtle.penup() turtle.pendown() turtle.setpos(*x*,*y*)	**turtle.setup(*w*,*h*)** – команда для установления размеров окна, отличных от тех, которые по умолчанию. **#** – знак, используемый для отображения в коде программы текста комментария.
Тема урока	Компьютерный исполнитель
Базовый набор команд	*Дополнительный набор команд и знаков*
import turtle turtle.shape('*name form*') **turtle.setup(*w*,*h*)** turtle.penup() turtle.pendown() turtle.setpos(*x*,*y*)	**#** – знак, используемый для отображения в коде программы текста комментария. **turtle.speed()** – команда для настройки скорости выполнения программы компьютерным исполнителем Черепаха.

	turtle.done() – команда, позволяющая оставлять открытым холст с результатом выполнения программы.

Тема урока	Изучение и изменение готовых программ
Базовый набор команд и знаков	*Дополнительный набор команд*
import turtle turtle.shape('*name form*') turtle.setup(*w*,*h*) turtle.penup() turtle.pendown() turtle.setpos(*x,y)* **pensize(*th*)** **turtle.color(*'name color'*)** **turtle.fillcolor(*'name color'*)** **begin_fill()** **end_fill()** **#**	**turtle.circle(r)** – команда для рисования окружности указанного радиуса **turtle.forward(*d*)** – команда для перемещения компьютерного исполнителя Черепаха на указанное расстояние в пикселях вперед **turtle.right(α)** – команда для поворота направо (на месте) компьютерного исполнителя Черепаха на указанное количество градусов **turtle.setheading(α)** – команда для установки направления движения компьютерного исполнителя Черепаха (0 – вправо, на восток; 90 – вверх, на север; 180 – влево, на запад; 270 – вниз, на юг). turtle.speed(*v*) turtle.done()

Тема урока	Использование вспомогательных алгоритмов
Базовый набор команд и знаков	*Дополнительный набор команд*
def *name_functions()* import turtle turtle.shape('*name form*') turtle.setup(*w*,*h*) turtle.penup() turtle.pendown() turtle.setpos(*x,y)* **turtle.forward(*d*)** **turtle.right(*α*)** **turtle.setheading(*α*)** **#**	**turtle.backforward(*d*)** – команда для перемещения компьютерного исполнителя Черепаха на указанное расстояние в пикселях назад **turtle.left(α)** – команда для поворота налево (на месте) компьютерного исполнителя Черепаха на указанное количество градусов turtle.speed(*v*) turtle.done()

Тема урока	Составление алгоритма для исполнителя
Базовый набор команд и знаков	*Дополнительный набор команд*
import turtle def *name_functions()* turtle.shape('*name form*') turtle.setup(*w*,*h*) turtle.penup() turtle.pendown() turtle.setpos(*x,y)* turtle.forward(*d*) turtle.right(*α*)	turtle. backforward(*d*) turtle.left(α) turtle.speed(*v*) turtle.done()

turtle.setheading (α) pensize(*th*) turtle.color(*'name color'*) turtle.fillcolor(*'name color'*) begin_fill() end_fill() # turtle.circle(*r*)	

Особенности построения дидактической системы учебных задания для реализации стратегии преемственного перехода при обучении учащихся началам программирования с помощью компьютерного исполнителя *ЧЕРЕПАХА*

При построении дидактической системы учебных заданий для реализации стратегии преемственного перехода при обучении учащихся в 6 классе началам программирования с помощью компьютерного исполнителя *Черепаха* нами предлагается придерживаться следующей последовательность структурирования учебного материала по урокам.

На уроке «Понятие алгоритма и исполнителя» для пропедевтических целей предлагать учащимся задания, которые потребуют выполнить алгоритмы, записанные в словесной форме. *Исполнителю Шестиклассник* необходимо по предложенным координатам точек создать изображение в декартовой системе координат на клеточной бумаге. *Исполнителю Следопыт* нужно по предложенной системе командам (двигайся вперед, поверни на север, поверни на восток, поверни на юг, поверни на запад) пройти указанный маршрут на карте.

При разработке заданий для *исполнителя Школьник* полезно обратится к разделу «Рисунки и фигуры по координатам» электронного ресурсу [15].

На уроке «Способы записи алгоритмов», продолжая предварительную подготовку учащихся к будущему написанию ими программ, полезно предлагать следующие задания. В условиях в явном виде представлен результат построения геометрических фигур по координатам (для *исполнителя Шестиклассник*) и нарисован маршрут на карте (для *исполнителя Следопыт*, который он прошел). Учащимся необходимо в словесной или в графической (в виде блок-схемы) форме предоставить запись алгоритма, который должен был выполнить каждый из исполнителей, чтобы получить представленный результат. На рисунке (рис. 1) изображены пример ломаной и ряда геометрических фигур, которые знакомы учащимся 6 класса из уроков учебного предмета «Математика» ([18]). Обучающимся можно предложить для двух различных

объектов записать соответствующие алгоритм для исполнителя Шестиклассник в словесной и (или) графической форме.

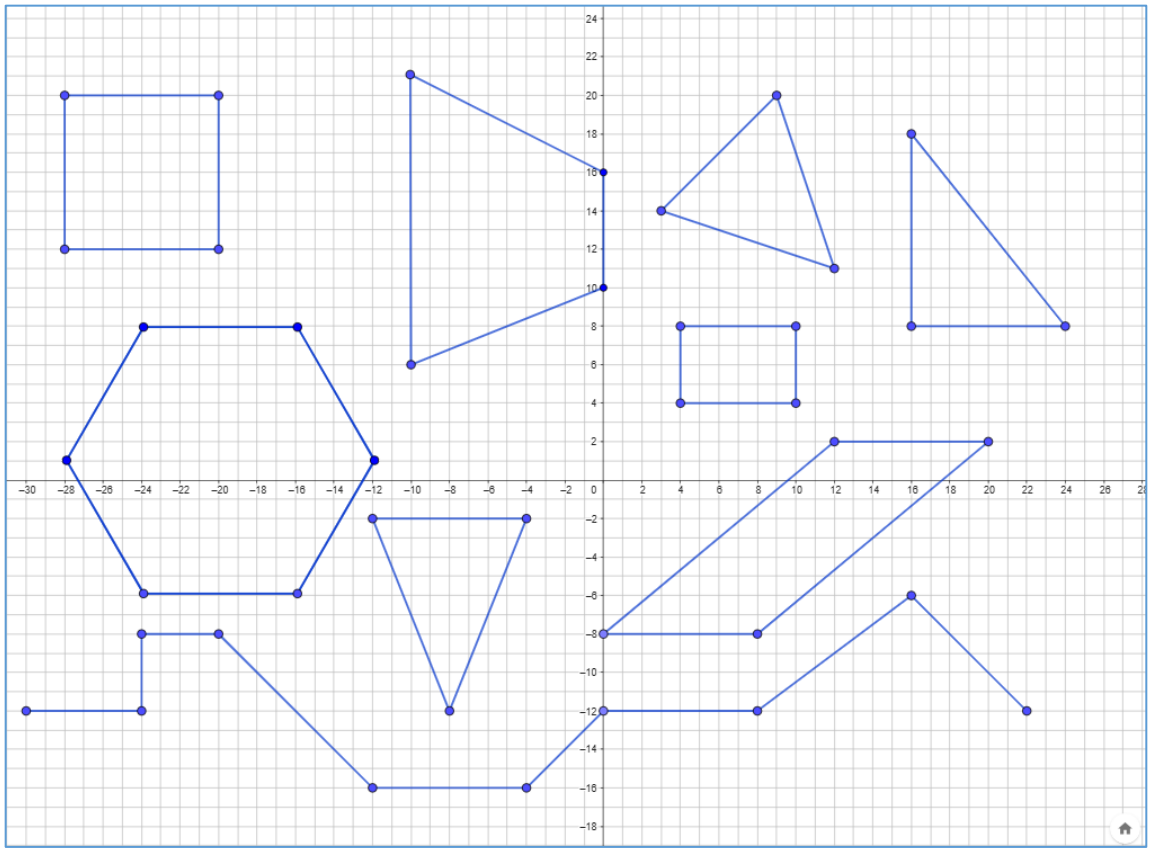


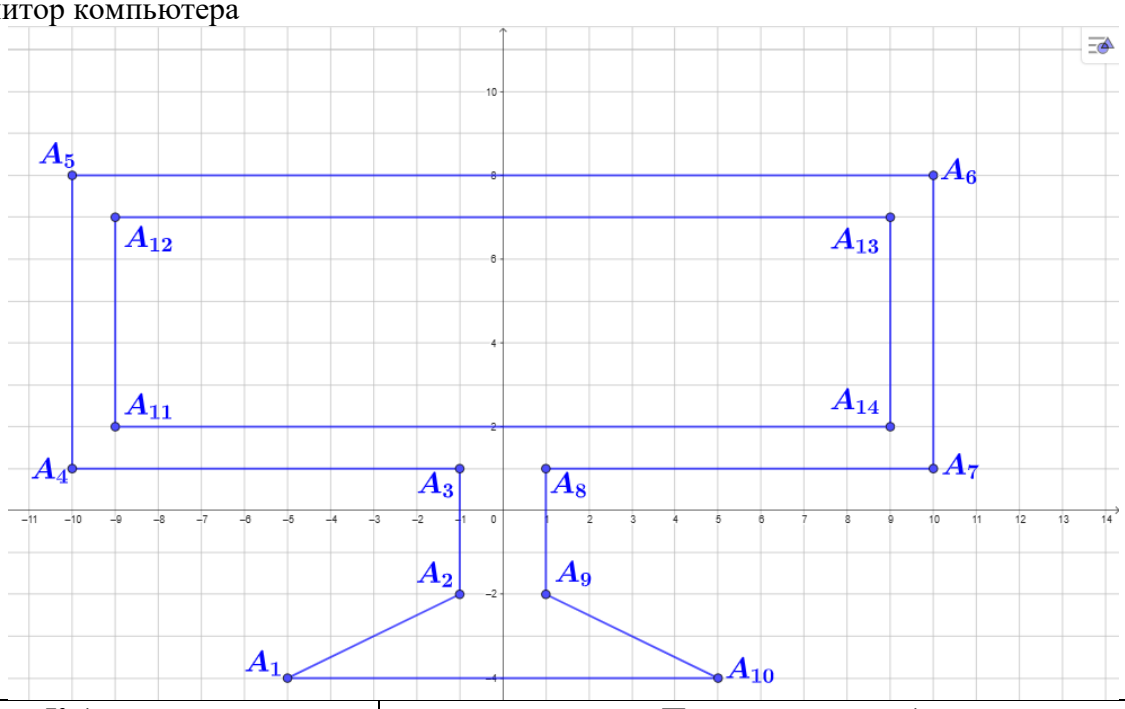
Рисунок 1. – Пример ломаной и ряда геометрических фигур на клеточной бумаге, которые могут быть использованы при разработке учащимися алгоритмов для *исполнителя Шестиклассник*

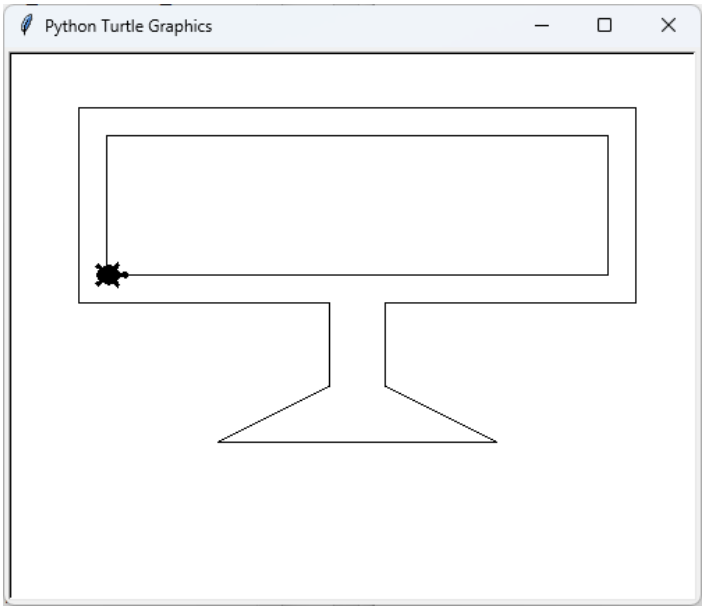
На уроке «Среда программирования» осуществляется работа с заданиями, в которых учащимся нужно непосредственно для *компьютерного исполнителя Черепаха* подготовить код программы, позволяющий нарисовать простейшие изображения по координатам.

При этом заранее обсуждается с учащимися то, что в качестве условных единиц измерения на компьютере выступаю пиксели. И в качестве соотношения (масштаба), можно рассматривать, например, вариант 1 : 40, т.е. 1 см соответствует 40 px (или длина стороны 1 клетки равна 20 px). Также полезно сразу привести иллюстрацию задания, пример кода программы, его словесное пояснение и результат выполнения (табл. 3).

Далее по изображению и словесному пояснению записать код программы в тетради. Затем по аналогичному коду программы для исполнителя *Черепах* попробовать в тетради нарисовать изображение (то есть побыть в роли компьютерного исполнителя). И, далее, на основе рисунка и блок-схемы непосредственно набрать код программы на компьютере и протестировать ее работу.

Таблица 3. – Пример вводного задания для учащихся по разработке кода программы для компьютерного *исполнителя Черепаха*

Визуальное представление условия			
Монитор компьютера 			
Код программы		***Пояснение команд***	
1	import turtle	Вызов *исполнителя Черепаха.*	
2	turtle.shape('turtle')	Выбор внешнего вида исполнителя	
3	turtle.setup(500,400)	Установить размеры окна 500 на 400 пикселей.	
4	turtle.penup()	Поднять перо, чтобы не оставлять след при перемещении.	
5	*Рисуем контур монитора*		
6	turtle.setpos(-100,-80)	Переместится в точку с координатами (–100; –80).	A_1
7	turtle.pendown()	Опустить перо, чтобы оставлять след при перемещении.	
8	turtle.setpos(-20,-40)	Переместится в точку с координатами (–20; –40).	A_2
9	turtle.setpos(-20,20)	Переместится в точку с координатами (–20; 20).	A_3
10	turtle.setpos(-200,20)	Переместится в точку с координатами (–200; 20).	A_4
11	turtle.setpos(-200,160)	Переместится в точку с координатами (–200; 160).	A_5
13	turtle.setpos(200,160)	Переместится в точку с координатами (200; 160).	A_6
13	turtle.setpos(200,20)	Переместится в точку с координатами (200; 20).	A_7
14	turtle.setpos(20,20)	Переместится в точку с координатами (20; 20).	A_8
15	turtle.setpos(20,-40)	Переместится в точку с координатами (20; –40).	A_9
16	turtle.setpos(100,-80)	Переместится в точку с координатами (20; –80).	A_{10}
17	turtle.setpos(-100,-80)	Переместится в точку с координатами (–100; –80).	A_1
18	turtle.penup()	Поднять перо, чтобы не оставлять след при перемещении.	
19	*Рисуем экран монитора*		
20	turtle.setpos(-180,40)	Переместится в точку с координатами (–180; 40).	A_{11}
21	turtle.pendown()	Опустить перо, чтобы оставлять след при перемещении.	
22	turtle.setpos(-180,140)	Переместится в точку с координатами (–180; 140).	A_{12}
23	turtle.setpos(180,140)	Переместится в точку с координатами (180; 140).	A_{13}
24	turtle.setpos(180,40)	Переместится в точку с координатами (180; 40).	A_{14}

25	turtle.setpos(-180,40)	Переместится в точку с координатами (–180; 40).	A_{11}
26	turtle.done()	Оставить холст с результатом выполнения открытым.	
Результат работы программы			
			

На уроке «Компьютерный исполнитель» осуществляется развитие начальных умений учащихся по написанию кода программы, полученных на предыдущем уроке.

Учащимся предлагаются задания, следующего вида:

1) имеется на клеточной бумаге изображение, которое нужно нарисовать, и фрагмент начала кода программы. Необходимо дописать код программы, используя возможность копирования предыдущих строк кода и изменения координат точек, в которые должен перемещаться компьютерный исполнитель *Черепаха*;

2) *имеются* на клеточной бумаге *изображение*, которое надо построить, *текстовый документ* с системой команд компьютерного исполнителя *Черепаха* и *файл* с указанием в виде комментариев этапов плана построения изображения. Учащимся надо с помощью копирования из текстового документа и вставки в файл программы необходимых команд написать программу для построения изображения;

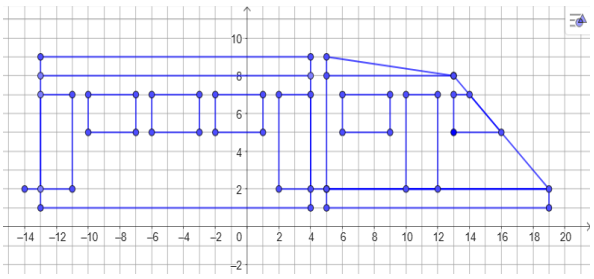
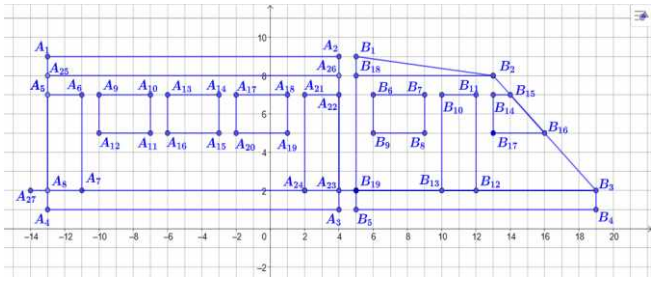
3) дана композиция из нескольких одинаковых объектов, расположенных в разных четвертях координатной плоскости. Учащимся на скорость нужно написать программу для ее рисования.

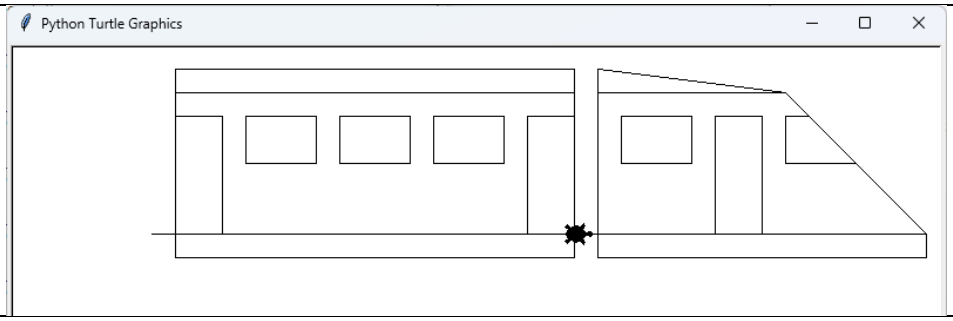
На двух последующих ***уроках по теме «Изучение и изменение готовых программ»*** осуществляется знакомство с дополнительными командами. Речь идет о командах, позволяющих создавать цветные изображения, а также о командах *переместиться вперед (назад)* и *повернуться на угол* по часовой (против часовой) стрелки. Исходя из этого, учащимся предоставляются две группы заданий.

Первая группа заданий: имеются цветное изображение и код программы, с помощью которого создано черно-белое изображение. Учащимся надо: 1) используя комментарии выделить этапы построения черно-белого изображения; 2) руководствуясь дополнительной инструкций с визуальным указанием на каком этапе построения каким образом настроить цвет пера, которым осуществляется рисование, а также на каком этапе можно закрасить отдельные части изображения; 3) предоставляется дополненный вариант цветного рисунка и учащимся самостоятельно нужно дописать код программы, разместив отдельные из имеющихся объектов в новых частях координатной плоскости.

Представляет интерес предложенный Л. Г. Овчинниковой, С. Г. Пузиновской в рабочей тетради на печатной основе вариант такой разработки по раскраске квадрата и домика [11, С. 100–101]. Указанные примеры могут быть взяты за основу обучения учащихся при работе с заданиями второго вида. В таблице (табл. 4) представлен пример разработки для учащихся, которые имеют высокую мотивацию к изучению программирования и готовы создавать сложные рисунки. Также важно здесь отметить, что составление фрагментов кода по рисованию окон и дверей в поезде отличаются только указанием координат. Поэтому учащимся можно предложить самостоятельно доработать код программы, используя идею примера 19.5 учебного пособия [8, С. 142–143].

Таблица 4. – Пример дополнительной инструкции для учащихся по рисованию кабины поезда и одного вагона в цветном виде средствами компьютерного исполнителя *Черепаха*

Текстовое и визуальное представление условия
Нужно было нарисовать кабину поезда (локомотив) и один вагон. На бумаге в клеточку представлены рисунки (один без указания шагов построения, второй с шагами построения в соответствии с номерами букв: А – построение вагона, В – построение кабины).
Вагон поезда  
Промежуточный результат работы программы



Код программы

Начало программы

```
import turtle
turtle.shape('turtle')
turtle.setup(800,400)
turtle.penup()
```

Кабина

#1.1 Внешний корпус кабины

```
turtle.setpos(100,180)
turtle.pendown()
turtle.setpos(260,160)
turtle.setpos(380,40)
turtle.setpos(380,20)
turtle.setpos(100,20)
turtle.setpos(100,180)
turtle.penup()
```

#1.2 Большое окно кабины

```
turtle.setpos(120,140)
turtle.pendown()
turtle.setpos(180,140)
turtle.setpos(180,100)
turtle.setpos(120,100)
turtle.setpos(120,140)
turtle.penup()
```

#1.3 Дверь кабины

```
turtle.setpos(200,140)
turtle.pendown()
turtle.setpos(240,140)
turtle.setpos(240,40)
turtle.setpos(200,40)
turtle.setpos(200,140)
turtle.penup()
```

#1.4 Маленькое окно кабины

```
turtle.setpos(260,140)
```

Вагон

#2.1 Внешний корпус вагона

```
turtle.setpos(-260,180)
turtle.pendown()
turtle.setpos(80,180)
turtle.setpos(80,20)
turtle.setpos(-260,20)
turtle.setpos(-260,180)
turtle.penup()
```

#2.2_1 Дверь в левой части вагона

```
turtle.setpos(-260,140)
turtle.pendown()
turtle.setpos(-220,140)
turtle.setpos(-220,40)
turtle.setpos(-260,40)
turtle.setpos(-260,140)
turtle.penup()
```

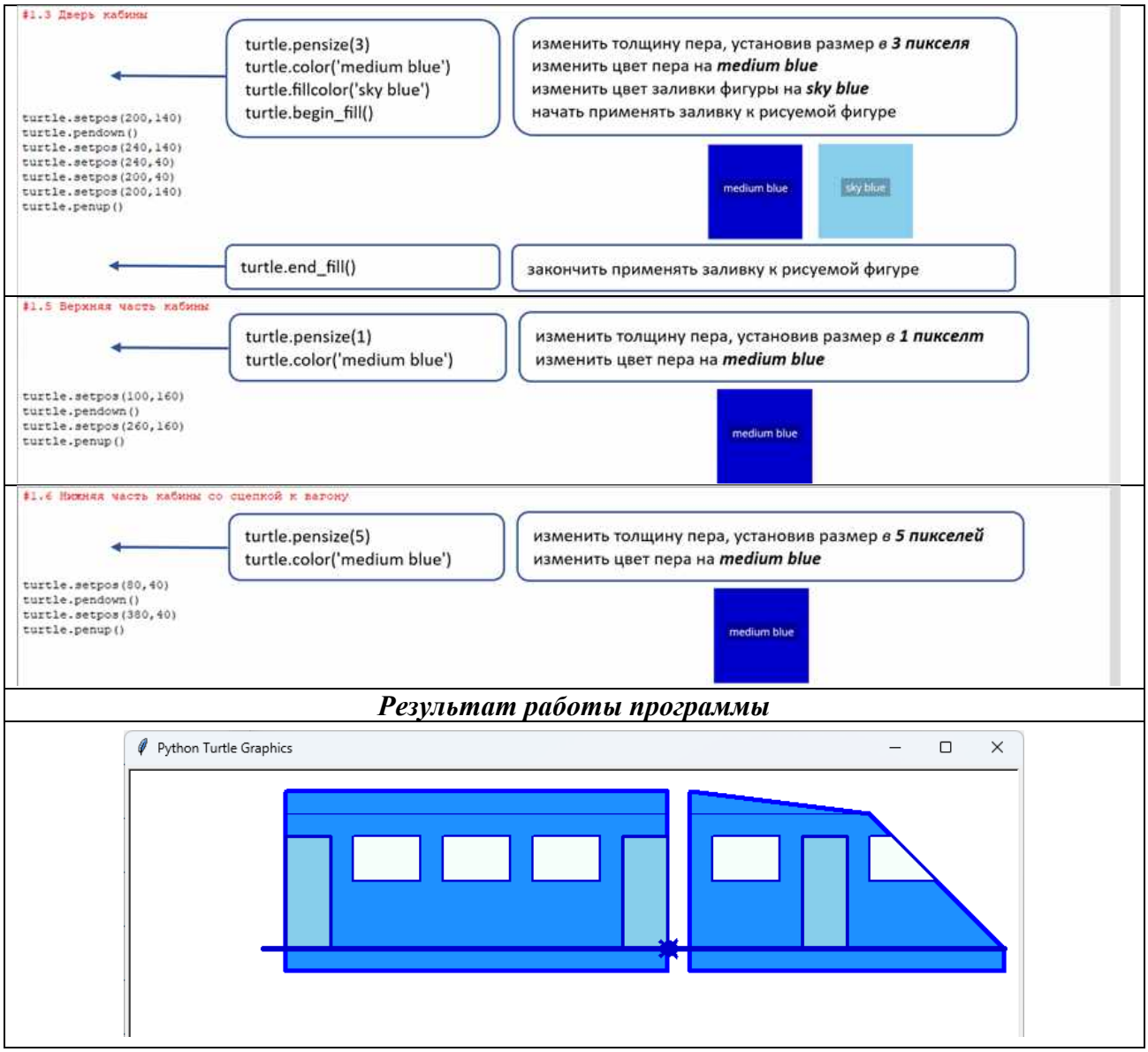
#2.3_1 Окно в левой части вагона

```
turtle.setpos(-200,140)
turtle.pendown()
turtle.setpos(-140,140)
turtle.setpos(-140,100)
turtle.setpos(-200,100)
turtle.setpos(-200,140)
turtle.penup()
```

#2.3_2 Окно в центральной части вагона

```
turtle.setpos(-120,140)
turtle.pendown()
turtle.setpos(-60,140)
turtle.setpos(-60,100)
turtle.setpos(-120,100)
turtle.setpos(-120,140)
turtle.penup()
```

turtle.pendown() turtle.setpos(280,140) turtle.setpos(320,100) turtle.setpos(260,100) turtle.setpos(260,140) turtle.penup()	**#2.3_3 Окно в правой части вагона** turtle.setpos(-40,140) turtle.pendown() turtle.setpos(20,140) turtle.setpos(20,100) turtle.setpos(-40,100) turtle.setpos(-40,140) turtle.penup()
	#2.2_2 Дверь в правой части вагона turtle.setpos(40,140)turtle.pendown() turtle.setpos(80,140) turtle.setpos(80,40) turtle.setpos(40,40) turtle.setpos(40,140) turtle.penup()
#1.5 Верхняя часть кабины turtle.setpos(100,160) turtle.pendown() turtle.setpos(260,160) turtle.penup()	**#2.5 Верхняя часть вагона** turtle.setpos(-260,160) turtle.pendown() turtle.setpos(80,160) turtle.penup()
#1.6 Нижняя часть кабины со сцепкой к вагону turtle.setpos(80,40) turtle.pendown() turtle.setpos(380,40) turtle.penup()	**#2.4 Нижняя часть вагона со сцепкой к следующему вагону** turtle.setpos(-280,40) turtle.pendown() turtle.setpos(80,40) turtle.penup()
Инструкция для цветного представления кабины	
	
	



Результат работы программы

Вторая группа заданий направлена на рассмотрение альтернативного способа написания кода программы, используя команды переместиться вперед и повернуть направо. Выполняя такие действия с фрагментами кода программы, описанной в таблице (табл. 4), учащиеся смогут убедиться, что рисование окон (#1.2 Большое окно кабины; #2.3_1 Окно в левой части вагона; #2.3_2 Окно в центральной части вагона; #2.3_3 Окно в правой части вагона) и дверей (#1.3 Дверь кабины; #2.2_1 Дверь в левой части вагона; #2.2_2 Дверь в правой части вагона) при таком подходе представляют собой один и тот же набор команд (табл. 5). Единственным отличием является указание того, в какую точку компьютерного исполнителя *Черепаху* изначально нужно разместить.

Таблица 5. – Примеры альтернативного способа написания фрагментов кода программы для компьютерного исполнителя *Черепаха*

Фрагменты кода программы для рисования окна и двери	
Окно	*Дверь*
turtle.forward(60) turtle.right(90) turtle.forward(40) turtle.right(90) turtle.forward(60) turtle.right(90) turtle.forward(40) turtle.right(90)	turtle.forward(40) turtle.right(90) turtle.forward(100) turtle.right(90) turtle.forward(40) turtle.right(90) turtle.forward(100) turtle.right(90)

Важно обратить внимание учащихся на то, что в конце рисования альтернативным способом окна (или двери) нужно разместить компьютерного исполнителя в том направлении, в котором он должен продолжить создание всего рисунка (именно для этого и указана последняя команда *turtle.right(90)*). Для вычисления угла поворота *Черепахи* используется формула: $\beta = 180° - \alpha$, где α – это известный внутренний угол фигуры, относительно которого требуется сделать поворот. В системе заданий целесообразно придерживаться работы изначально со случаями, в которых исполнителя надо поворачивать вправо на 90° , затем на 45° и 60 , а далее еще и на 120 и 135° .

Следует помнить, что на текущем этапе обучения у учащихся 6 класса знания и умения по вычислению длин сторон геометрических фигур аналитико-вычислительными методами еще достаточно ограничены [17, 18]. Поэтому для определения примерного расстояния, на которое компьютерный исполнитель *Черепаха* должен переместиться, можно использовать результаты, полученные при измерении длин сторон с помощью линейки и указания этой величины с учетом масштаба. А далее корректировать эти данные исходя из тестирования программы.

На уроке «Использование вспомогательных алгоритмов» основное внимание уделяется обоснованию эффективности разработки программ для построения изображений с учетом того, что ряд действий могут быть вынесены в отдельный вспомогательный алгоритм (подпрограмму). Можно показать на предыдущем примере как уменьшиться код программы, если использовать две подпрограммы ***def*** *window ()* и ***def*** *door ()*. В таблице (табл. 6) представлены соответствующее подпрограммы.

Таблица 6. – Подпрограммы для компьютерного исполнителя *Черепаха*

Подпрограмма для рисования окна	*Подпрограмма для рисования двери*

`def window():`
`  turtle.pendown()`
`  turtle.forward(60)`
`  turtle.right(90)`
`  turtle.forward(40)`
`  turtle.right(90)`
`  turtle.forward(60)`
`  turtle.right(90)`
`  turtle.forward(40)`
`  turtle.right(90)`
`  turtle.penup()`
	`def door():`
	`  turtle.pendown()`
	`  turtle.forward(40)`
	`  turtle.right(90)`
	`  turtle.forward(100)`
	`  turtle.right(90)`
	`  turtle.forward(40)`
	`  turtle.right(90)`
	`  turtle.forward(100)`
	`  turtle.right(90)`
	`  turtle.penup()`

На текущем уроке учащимся можно предложить задания, в которых для компьютерного исполнителя *Черепаха* подготовлены подпрограммы, рисующие отдельные буквы алфавита ([8, С. 150], [9, С. 108]). Учащимся нужно используя эти подпрограммы:

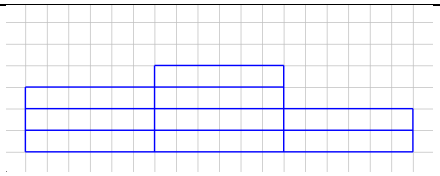
1) составить программы для написания указанных слов: а) цветом по умолчанию АЛГОРИТМ, ПРОГРАММА, ИНФОРМАТИКА, ЧЕРЕПАХА; б) определенным цветом: МОРЕ (синим – *blue*), ТРАВА (зеленым – *green*), РОЗА (красным – *red*);

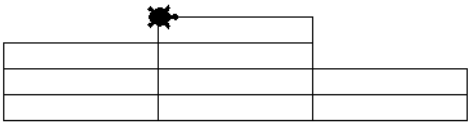
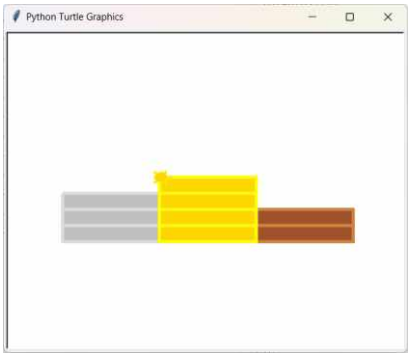
2) составить код для придуманного своего слова (если какой-то из букв не окажется среди фрагментов готовых подпрограмм, то разработать алгоритмы самостоятельно);

3) запустить программу с готовым рисунком и придумать подпись к нему.

На уроке «Составление алгоритма для исполнителя» можно организовать работу по выполнению заданий, в которых надо создать рисунок, используя одинаковые геометрические фигуры. Для рисования экземпляра геометрической фигуры обязательно использовать подпрограмму. Это могут быть: пьедестал для награждения, созданный из прямоугольников (табл. 7); горы из равносторонних треугольников ([11 , С. 109]), ракета, сконструированная из различных равнобедренных прямоугольных треугольников; кораблики, созданные из трапеций и др. Указанные объекты могут быть соответствующим образом раскрашены и подписаны.

Таблица 7. – Примеры задания на использование вспомогательных алгоритмов для рисования пьедестала из равных прямоугольников с помощью *Черепахи*

Визуальное представление условия	**Визуальное представление результата**
	

<table>
<tr><td></td><td>  </td></tr>
<tr><td colspan="2">Вариант реализации с использованием подпрограммы</td></tr>
<tr><td colspan="2">
<pre>
import turtle
#Прямоугольник
def rectangle():
    turtle.pendown()
    turtle.forward(120)
    turtle.right(90)
    turtle.forward(20)
    turtle.right(90)
    turtle.forward(120)
    turtle.right(90)
    turtle.forward(20)
    turtle.seth(0)
    turtle.penup()
</pre>
<pre>
#Основная программа
turtle.shape('turtle')
turtle.penup()
turtle.setup(500,400)
#Третье место
turtle.setpos(60, -40)
rectangle()
turtle.setpos(60, -20)
Prymoygolnik()
#Второе место
turtle.setpos(-180, -40)
rectangle()
turtle.setpos(-180, -20)
rectangle()
turtle.setpos(-180, 0)
rectangle()
</pre>
<pre>
#Первое место
turtle.setpos(-60, -40)
rectangle()
turtle.setpos(-60, -20)
rectangle()
turtle.setpos(-60, 0)
rectangle()
turtle.setpos(-60, 20)
rectangle()
</pre>
</td></tr>
<tr><td>Визуальное представление результата в цветном варианте</td><td>Команды для настройки толщины пера и цвета</td></tr>
<tr><td> 
</td><td>
<pre>
turtle.pensize(4)
#Третье место
turtle.color('peru')
turtle.fillcolor('sienna')
#Второе место
turtle.color('gainsboro')
turtle.fillcolor('silver')
#Первое место
turtle.color('yellow')
turtle.fillcolor('gold')
</pre>
</td></tr>
</table>

Заключение

Таким образом, в статье нами: предложена оптимизированная подсистема команд компьютерного исполнителя *Черепаха*; выделены базовый и дополнительные наборы команд; представлены рекомендации по последовательности расширения подсистемы рассматриваемых с учащимися 6 класса команд в соответствии с ходом изучения темы; раскрыты особенности построения дидактической системы учебных заданий при изучении начал программирования с помощью компьютерного исполнителя *Черепаха*; описана через методические рекомендации к урокам стратегия преемственного перехода

в использовании компьютерных исполнителей при обучении учащихся основам алгоритмизации и началам программирования.

Список библиографических ссылок (на языке оригинала)

1. Босова, Л. Л. О разноуровневом обучении программированию в курсе информатики основной школы в условиях дифференциации содержания обучения / Л. Л. Босова, Н. Н. Самылкина, В. А. Мишин // Преподаватель XXI век. – 2024. – № 1. – Часть 1. – С. 253–273.

2. Быкадоров, Ю. А. Преподавание информатики в начальной школе: зарубежный опыт / Ю. А. Быкадоров, О. В. Шепелюк // Подготовка учителя начальных классов: проблемы и перспективы : материалы V Междунар. науч.-практ. конф., Минск, 5 дек. 2018 г. / Белорус. гос. пед. ун-т ; редкол.: Н.В. Жданович [и др.]. – Минск, 2019. – С. 8–12.

3. Гришков, Д. Ю. Язык высокого уровня программирования Python / Д. Ю. Гришков, Н. М. Аусилова // Наука и реальность. – 2022. – № 1(9). – С. 114–117.

4. Зенько, С. И. Деятельностно-семантический подход к профессиональной направленности формирования понятийной компетенции учителя информатики в педагогическом университете / С. И. Зенько // Весці Бел. дзярж. пед. ун-та. Сер. 3, Фізіка. Матэматыка. Інфарматыка. Біялогія. Геаграфія. – 2018. – № 4. – С. 61–71.

5. Зенько, С. И. Особенности изучения общеучебных понятий школьной информатики и математики / С. И. Зенько // Математика. – 2019. – № 5. – С. 3–15.

6. Зуфарова, А. С. Методика обучения программированию учащихся: проблемы и решения / А. С. Зуфарова, Р. А. Суходуб // Управление образованием: теория и практика. – 2022. – Том 12. – № 4. – С. 166–174.

7. Информатика. 6–11-е кл.: примерное календарно-тематическое планирование: пособие для учителей учреждений образования, реализующих образоват. прогр. общ. сред. образования с рус. яз. обучения и воспитания / А. И. Лапо, Е. Н. Войтехович. – Минск : НИО : Аверсэв, 2023. – 63 с.

8. Котов, В. М. Информатика : учеб. пособие для 6 кл. учреждений общ. сред. образования с рус. яз. обучения / В. М. Котов, Н. П. Макарова, А. И. Лапо, Е. Н. Войтехович. – 2-е изд. пересм. и дополненное. – Минск : Нар. асвета, 2024. – 183 с.

9. Лапо, А. И. Информатика, 6 класс : практикум / А. И. Лапо, Е. Н. Вайтехович, А. Н. Мороз. – Минск : Аверсев, 2024. – 128 с.

10. Макарова, Н. П. Информатика : учеб. пособие для 6 кл. учреждений общ. сред. образования с рус. яз. обучения / Н. П. Макарова, А. И. Лапо, Е. Н. Войтехович. – Минск : Нар. асвета, 2018. – 168 с.

11. Овчинникова, Л. Г. Информатика : рабочая тетрадь для 6 класса / Л. Г. Овчинникова, С. Г. Пузиновская. – Минск : Аверсэв, 2024. – 126 с.

12. Проблемы программирования в рамках школьного образования / И. В. Барышева, О. А. Козлов, Е. В. Малкина, Н. В. Шестакова // Вестник Нижегородского университета им. Н. И. Лобачевского. – Серия: Социальные науки. – 2023. – № 3 (70). – С. 174–179.

13. Пупцев, А. Е. Информатика : учеб. пособие для 6-го кл. общеобразоват. учреждений с белорус. и рус. яз. обучения 11-летним сроком обучения / А. Е. Пупцев, Н. П. Макарова, А. И. Лапо. – Минск : Нар. асвета, 2008. – 126 с.

14. Пупцев, А. Е. Информатика : учеб. пособие для 6-го кл. учреждений, обеспечивающих получение общ. сред. Образования, с рус. яз. обучения с 12-летним сроком обучения / А. Е. Пупцев, П. Л. Гращенко, А. И. Лапо. – Минск : Нар. асвета, 2003. – 158 с.

15. Рисунки и фигуры по координатам / Е. Е. Алексеева. – [Электронный ресурс]. URL: https://alekseevaee.ru/metodobespechenie/programmy/risuem-po-koordinatam#Risunki-10-00 (дата обращения: 01.10.2024).

16. Стоянов, Н. Э. Развитие средств обучения основам программирования во второй половине XX – начале XXI века / Н. Э. Стоянов, В. Н. Тарасова // История и педагогика естествознания. –2022. – № 2-3. – С. 74–79.

17. Учебная программа по учебному предмету «Математика» для V класса учреждений образования, реализующих образовательные программы общего среднего образования с русским языком обучения и воспитания. – [Электронный ресурс]. URL: https://adu.by/images/2023/08/matem/up_mat_5_rus_1.docx (дата обращения: 01.10.2024).

18. Учебная программа по учебному предмету «Математика» для VI класса учреждений образования, реализующих образовательные программы общего среднего образования с русским языком обучения и воспитания – [Электронный ресурс]. URL: https://adu.by/images/2023/08/matem/up_mat_6_rus_1.docx).

References (на английском языке)

1. Bosova, L. L. O raznourovnevom obuchenii programmirovaniyu v kurse informatiki osnovnoy shkoly v usloviyakh differentsiatsii soderzhaniya obucheniya / L. L. Bosova, N. N. Samylkina, V. A. Mishin // Prepodavatel' XXI vek. – 2024. – № 1. – Chast' 1. – P. 253–273. (In Russian)

2. Bykadorov, YU. A. Prepodavaniye informatiki v nachal'noy shkole: zarubezhnyy opyt / YU. A. Bykadorov, O. V. Shepelyuk // Podgotovka uchitelya nachal'nykh klassov: problemy i perspektivy : materialy V Mezhdunar. nauch.-prakt. konf., Minsk, 5 dek. 2018 g. / Belorus. gos. ped. un-t ; redkol.: N.V. Zhdanovich [i dr.]. – Minsk, 2019. – P. 8–12. (In Russian)

3. Grishkov, D. YU. YAzyk vysokogo urovnya programmirovaniya Python / D. YU. Grishkov, N. M. Ausilova // Nauka i real'nost'. – 2022. – № 1(9). – P. 114–117. (In Russian)

4. Zen'ko, S. I. Deyatel'nostno-semanticheskiy podkhod k professional'noy napravlennosti formirovaniya ponyatiynoy kompetentsii uchitelya informatiki v pedagogicheskom universitete / S. I. Zen'ko // Vestsí Bel. dzyarzh. ped. un-ta. Ser. 3, Fízíka. Matematyka. Ínfarmatyka. Bíyalogíya. Geagrafíya. – 2018. – № 4. – P. 61–71. (In Russian)

5. Zen'ko, S. I. Osobennosti izucheniya obshcheuchebnykh ponyatiy shkol'noy informatiki i matematiki / S. I. Zen'ko // Matematika. – 2019. – № 5. – P. 3–15. (In Russian)

6. Zufarova, A. S. Metodika obucheniya programmirovaniyu uchashchikhsya: problemy i resheniya / A. S. Zufarova, R. A. Sukhodub // Upravleniye obrazovaniyem: teoriya i praktika. – 2022. – Tom 12. – № 4. – P. 166–174. (In Russian)

7. Informatika. 6–11-ye kl.: primernoye kalendarno-tematicheskoye planirovaniye: posobiye dlya uchiteley uchrezhdeniy obrazovaniya, realizuyushchikh obrazovat. progr. obshch. sred. obrazovaniya s rus. yaz. obucheniya i vospitaniya / A. I. Lapo, Ye. N. Voytekhovich. – Minsk : NIO : Aversev, 2023. – 63 p. (In Russian)

8. Kotov, V. M. Informatika : ucheb. posobiye dlya 6 kl. uchrezhdeniy obshch. sred. obrazovaniya s rus. yaz. obucheniya / V. M. Kotov, N. P. Makarova, A. I. Lapo, Ye. N. Voytekhovich. – 2-ye izd. peresm. i dopolnennoye. – Minsk : Nar. asveta, 2024. – 183 p. (In Russian)

9. Lapo, A. I. Informatika, 6 klass : praktikum / A. I. Lapo, Ye. N. Vaytekhovich, A. N. Moroz. – Minsk : Aversev, 2024. – 128 p. (In Russian)

10. Makarova, N. P. Informatika : ucheb. posobiye dlya 6 kl. uchrezhdeniy obshch. sred. obrazovaniya s rus. yaz. obucheniya / N. P. Makarova, A. I. Lapo, Ye. N. Voytekhovich. – Minsk : Nar. asveta, 2018. – 168 p. (In Russian)

11. Ovchinnikova, L. G. Informatika : rabochaya tetrad' dlya 6 klassa / L. G. Ovchinnikova, S. G. Puzinovskaya. – Minsk : Aversev, 2024. – 126 p. (In Russian)

12. Problemy programmirovaniya v ramkakh shkol'nogo obrazovaniya / I. V. Barysheva, O. A. Kozlov, Ye. V. Malkina, N. V. Shestakova // Vestnik Nizhegorodskogo universiteta im. N. I. Lobachevskogo. – Seriya: Sotsial'nyye nauki. – 2023. – № 3 (70). – P. 174–179. (In Russian)

13. Puptsev, A. Ye. Informatika : ucheb. posobiye dlya 6-go kl. obshcheobrazovat. uchrezhdeniy s belorus. i rus. yaz. obucheniya 11-letnim srokom obucheniya / A. Ye. Puptsev, N. P. Makarova, A. I. Lapo. – Minsk : Nar. asveta, 2008. – 126 p. (In Russian)

14. Puptsev, A. Ye. Informatika : ucheb. posobiye dlya 6-go kl. uchrezhdeniy, obespechivayushchikh polucheniye obshch. sred. Obrazovaniya, s rus. yaz. obucheniya s 12-letnim srokom obucheniya / A. Ye. Puptsev, P. L. Grashchenko, A. I. Lapo. – Minsk : Nar. asveta, 2003. – 158 p. (In Russian)

15. Risunki i figury po koordinatam / Ye. Ye. Alekseyeva. – [Electronic resource]. Available at: https://alekseevaee.ru/metodobespechenie/programmy/risuem-po-koordinatam#Risunki-10-00 (In Russian)

16. Stoyanov, N. E. Razvitiye sredstv obucheniya osnovam programmirovaniya vo vtoroy polovine XX – nachale XXI veka / N. E. Stoyanov, V. N. Tarasova // Istoriya i pedagogika yestestvoznaniya. –2022. – № 2-3. – P. 74–79. (In Russian)

17. Uchebnaya programma po uchebnomu predmetu «Matematika» dlya V klassa uchrezhdeniy obrazovaniya, realizuyushchikh obrazovatel'nyye programmy obshchego srednego obrazovaniya s russkim yazykom obucheniya i vospitaniya. – [Electronic resource]. Available at: https://adu.by/images/2023/08/matem/up_mat_5_rus_1.docx (In Russian)

18. Uchebnaya programma po uchebnomu predmetu «Matematika» dlya VI klassa uchrezhdeniy obrazovaniya, realizuyushchikh obrazovatel'nyye programmy obshchego srednego obrazovaniya s russkim yazykom obucheniya i vospitaniya – [Electronic resource]. Available at: https://adu.by/images/2023/08/matem/up_mat_6_rus_1.docx) (In Russian)