



Действия по предписанию при обучении программированию на языке *Python*

Корчажкина Ольга Максимовна, к.т.н., с.н.с.

ФИЦ «Информатика и управление» РАН, г. Москва

16-17 мая 2024 года, г. Тверь



Что такое учебное предписание?

- *Учебное предписание* – это **особый тип** учебных инструкций, следуя которым учащиеся осваивают различные способы универсальных учебных действий (УУД).
- Порядок этих учебных инструкций воспроизводит **алгоритм** решения задачи.
- На этапе первичной модели учитель составляет **перечень** учебных предписаний, представляющих собой побудительные (мотивационные) предложения выполнить те или иные операции, вкуче составляющие **УУД**.
- По мере освоения отдельной операции на базе предъявляемого перечня записывается соответствующий **алгоритм**, используемый учащимися для выполнения **однотипных заданий**.
- Следуя расширяющимся и усложняющимся спискам предписаний, учащиеся осваивают и **другие виды операций**, которые формируют **разнообразные УУД**.



Виды учебных предписаний

- *Дидактические предписания* (предписания-модели) ориентируют учащихся на **целостное восприятие задачной ситуации**, состоящей из набора последовательных действий, приводящих к решению конкретной задачи.
- *Алгоритмические предписания* предусматривают **разделение учебного материала** на серию учебных заданий, последовательное выполнение которых приводит к освоению определённых УУД. При этом только овладение предыдущими сериями учебных заданий даёт возможность переходить к следующей серии.



Работа с «модельными» задачами



- выделение **типов задач**, для которых составляется предписание;
- выбор **задач-моделей** для решения;
- демонстрация **решения задачи** по предписанию с постепенным усложнением;
- организация **самостоятельного решения** учащимися однотипных задач;
- **обобщение результатов** решения задач с анализом последовательности действий по предписанию;
- **составление алгоритма** решения задачи на основе предписаний;
- **расширение типов задач**, в которых могут быть использованы сходные предписания и алгоритмы.



Работа с «модельными» задачами

Задача «Угадай-ка» с вызовом модуля *random*:

«Компьютер генерирует целое число из предложенного диапазона, а пользователь пытается угадать его, предлагая свои варианты, за определённое количество попыток».

```
Угадай-ка.py - C:\Users\olgak\OneDrive\Рабочий стол\Угадай-ка...
File Edit Format Run Options Window Help
import random
a = random.randint (1,10)
p = 0
while p < 5:
    b = int(input('Введите число: '))
    if a == b:
        print('Поздравляем! Вы угадали')
        break
    elif a > b:
        print('Загаданное число больше введённого')
    else:
        print('Загаданное число меньше введённого')
    p = p + 1
if p == 5:
    print('Вы не угадали. Компьютер загадал число:', a)

Ln: 17 Col: 0
```



Работа с «модельными» задачами



Задача «Угадай-ка»: составление предписания для игровой задачи на отгадывание числа, которое сгенерировано программой в пределах заданного диапазона (способ предписаний «по хронологии»).

Этап 1: начальные предписания:

- 1) подключи модуль генератора чисел *random*;
- 2) присвой переменной *a* псевдослучайное число от 0 до 10;
- 3) присвой переменной *b* целое число, введённое с клавиатуры;
- 4) запиши основную конструкцию условия, согласно которому выносится оценка «угадал – не угадал».

Этап 2: вводится ограничение на число попыток:

- 1) введи переменную *p*, отвечающую за число попыток, и присвой ей начальное значение 0;
- 2) объяви цикл с условием “*p* строго меньше 5”;
- 3) сдвинь вправо все условия, предназначенные для игры (if, elif, else);
- 4) увеличь на единицу число предпринятых попыток.

Этап 3: вводится инструкция о досрочном завершении игры с помощью оператора *break* в случае точного «попадания в цель».

Этап 4: если число попыток превышает заданную величину, а пользователь не угадал задуманное компьютером число, то на экран выводится сообщение о поражении, и игра прекращается.

После отработки программы для задачи-модели учащиеся самостоятельно модифицируют программу, изменяя интервал случайных чисел, число попыток или тексты для печати, а затем записывают полный алгоритм решения с помощью кратких предписаний.



```
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024
, 14:05:25) [MSC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license(
)" for more information.
>>>
= RESTART: C:\Users\olgak\OneDrive\Рабочий стол\
Угадай-ка.py
Введите число: 5
Загаданное число меньше введённого
Введите число: 3
Загаданное число больше введённого
Введите число: 4
Поздравляем! Вы угадали
>>>
===== RESTART: C:\Users\olgak\OneDrive\Рабо
чий стол\Угадай-ка.py =====
Введите число: 9
Загаданное число меньше введённого
Введите число: 1
Загаданное число больше введённого
Введите число: 2
Загаданное число больше введённого
Введите число: 3
Загаданное число больше введённого
Введите число: 4
Загаданное число больше введённого
Вы не угадали. Компьютер загадал число: 6
>>>
```

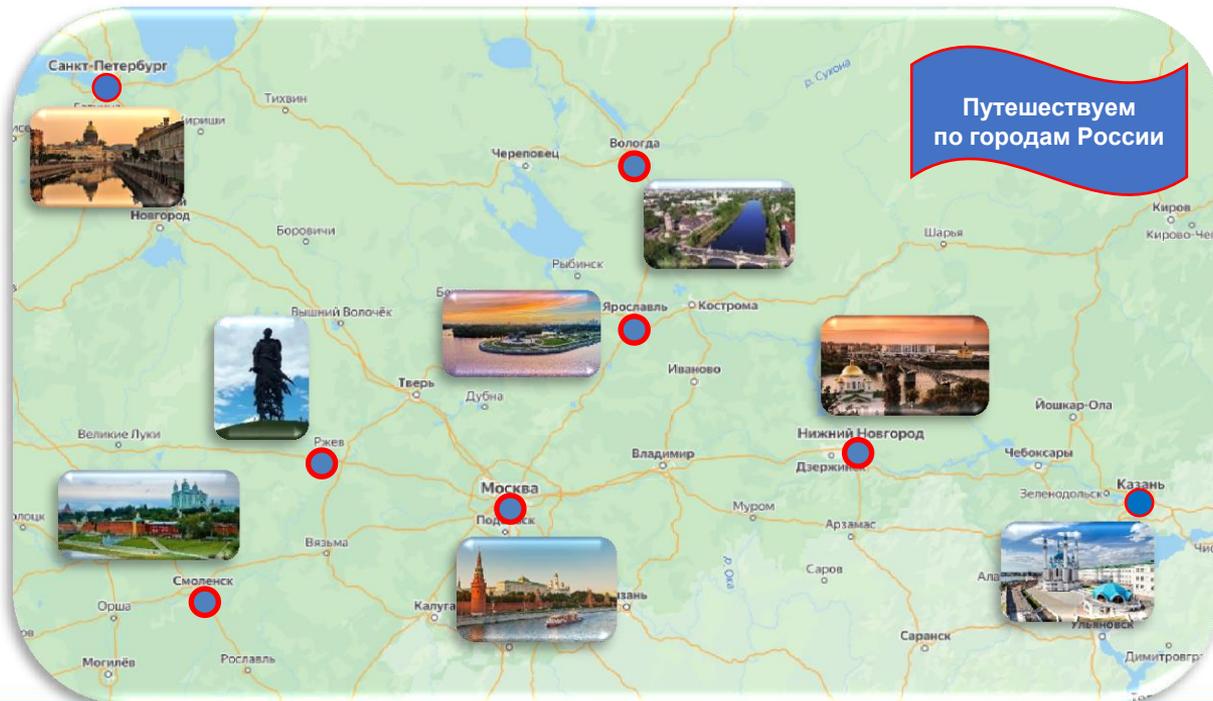


Действие по алгоритмическим предписаниям



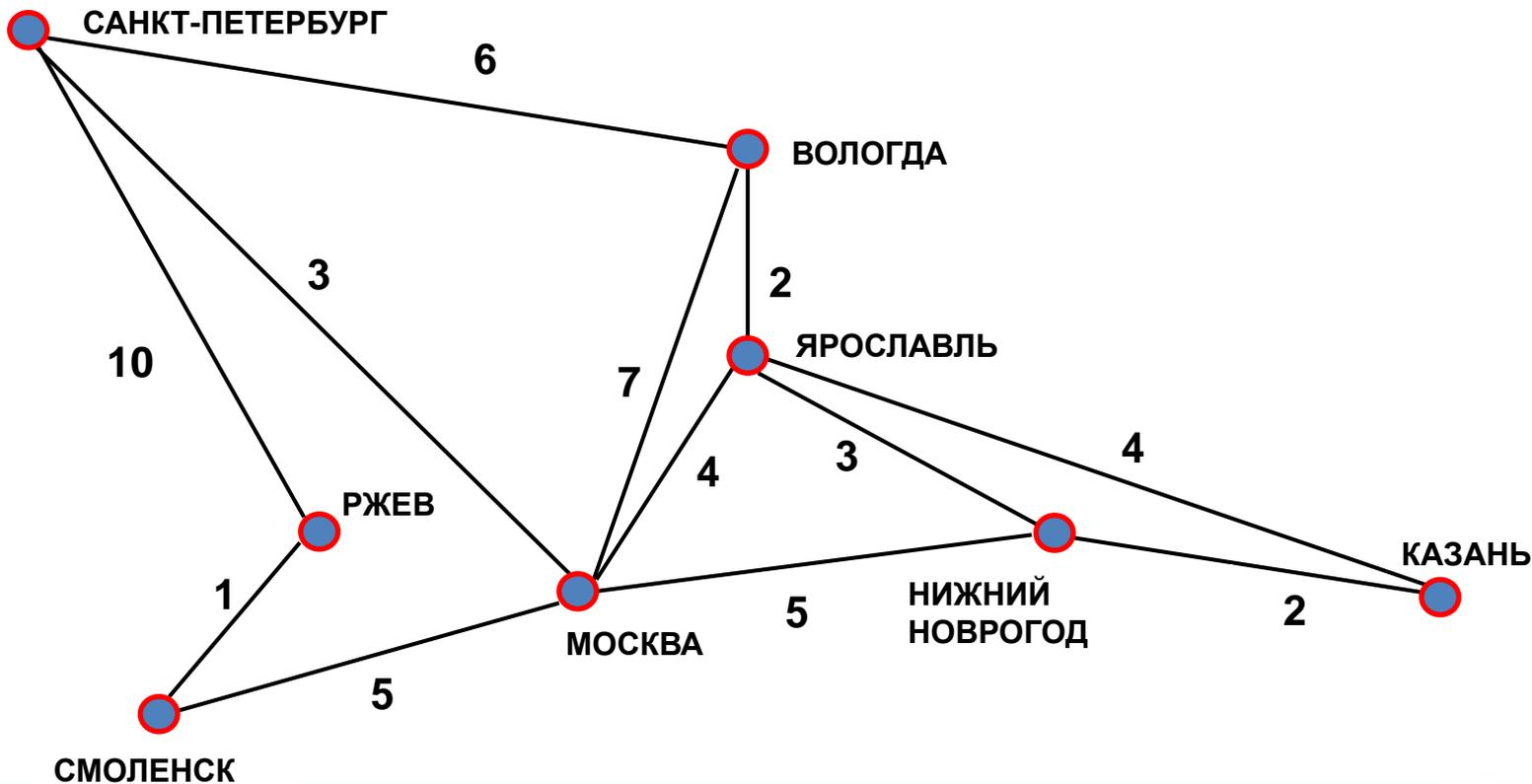
Задача «Путешествуем по России» (алгоритм Дейкстры):

«Требуется определить оптимальный маршрут между двумя заданными городами, начав путешествие в одном из городов и закончив его в другом».





Действие по алгоритмическим предписаниям





ОДОБРЕНА РЕШЕНИЕМ ФЕДЕРАЛЬНОГО УЧЕБНО-МЕТОДИЧЕСКОГО
ОБЪЕДИНЕНИЯ ПО ОБЩЕМУ ОБРАЗОВАНИЮ,
протокол 7/22 от 29.09.2022 г.

ПРИМЕРНАЯ РАБОЧАЯ ПРОГРАММА
СРЕДНЕГО ОБЩЕГО ОБРАЗОВАНИЯ

ИНФОРМАТИКА

УГЛУБЛЁННЫЙ УРОВЕНЬ

(для 10—11 классов образовательных организаций)

МОСКВА
2022

05

Продолжение

Примерная рабочая программа

Примерные темы, раскрывающие данный раздел программы, и количество часов, отводимое на их изучение	Учебное содержание	Основные виды деятельности учащихся при изучении темы (на уровне учебных действий)
<p>графы</p>	<p>Многоразрядные целые числа, задачи длинной арифметики. Словари (ассоциативные массивы, отображения). Хэш-таблицы. Построение алфавитно-частотного словаря для заданного текста. <i>Анализ текста на естественном языке. Выделение последовательностей по шаблону. Регулярные выражения. Частотный анализ.</i> Стеки. Анализ правильности скобочного выражения. Вычисление арифметического выражения, записанного в постфиксной форме. Очереди. Использование очереди для временного хранения данных. <i>Связные списки. Реализация стека и очереди с помощью связанных списков.</i> Алгоритмы на графах. Построение минимального остовного дерева взвешенного связанного неориентированного графа. <i>Обход графа в глубину. Обход графа в ширину.</i> Количество</p>	<ul style="list-style-type: none"> ■ Пояснять принципы обработки многоразрядных целых чисел и реализовывать соответствующие алгоритмы на языке программирования. ■ Применять словари (ассоциативные массивы, отображения) в задачах обработки данных. ■ Выполнять простой анализ текста на естественном языке, в том числе с использованием регулярных выражений. ■ Пояснять принципы работы стека и очереди, использовать стеки и очереди для решения алгоритмических задач. ■ Реализовывать и использовать двоичные (бинарные) деревья и графы для решения задач обработки данных. ■ Использовать динамическое программирование для вычисления рекурсивных функций, подсчёта



ОДОБРЕНА РЕШЕНИЕМ ФЕДЕРАЛЬНОГО УЧЕБНО-МЕТОДИЧЕСКОГО
ОБЪЕДИНЕНИЯ ПО ОБЩЕМУ ОБРАЗОВАНИЮ,
протокол 7/22 от 29.09.2022 г.

ПРИМЕРНАЯ РАБОЧАЯ ПРОГРАММА
СРЕДНЕГО ОБЩЕГО ОБРАЗОВАНИЯ

ИНФОРМАТИКА

УГЛУБЛЁННЫЙ УРОВЕНЬ

(для 10—11 классов образовательных организаций)

МОСКВА
2022

алгоритм Дейкстры

различных путей между вершинами ориентированного ациклического графа. Алгоритм Дейкстры.
Алгоритм Флойда–Уоршелла.
Деревья. Реализация дерева с помощью ссылочных структур. Двоичные (бинарные) деревья. Построение дерева для заданного арифметического выражения. Рекурсивные алгоритмы обхода дерева. Использование стека и очереди для обхода дерева. Динамическое программирование как метод решения задач с сохранением промежуточных результатов. Задачи, решаемые с помощью динамического программирования: вычисление рекурсивных функций, подсчёт количества вариантов, задачи оптимизации.

Практические работы

1. Поиск простых чисел в заданном диапазоне.
2. Реализация вычислений с многозначными числами.
3. Построение алфавитно-частотного словаря для заданного текста.
4. *Анализ текста на естественном языке.*
5. Вычисление арифметического выражения, записанного в постфиксной форме.

количества вариантов и решения задач оптимизации



ОДОБРЕНА РЕШЕНИЕМ ФЕДЕРАЛЬНОГО УЧЕБНО-МЕТОДИЧЕСКОГО
ОБЪЕДИНЕНИЯ ПО ОБЩЕМУ ОБРАЗОВАНИЮ,
протокол 7/22 от 29.09.2022 г.

ПРИМЕРНАЯ РАБОЧАЯ ПРОГРАММА
СРЕДНЕГО ОБЩЕГО ОБРАЗОВАНИЯ

ИНФОРМАТИКА

УГЛУБЛЁННЫЙ УРОВЕНЬ

(для 10—11 классов образовательных организаций)

МОСКВА
2022

25

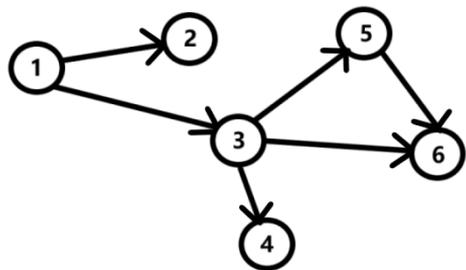
Продолжение

Примерная рабочая программа

Примерные темы, раскрывающие данный раздел программы, и количество часов, отводимое на их изучение	Учебное содержание	Основные виды деятельности учащихся при изучении темы (на уровне учебных действий)
алгоритм Дейкстры	6. Использование очереди. 7. Использование деревьев для вычисления арифметических выражений. 8. Вычисление длины кратчайшего пути между вершинами графа (алгоритм Дейкстры). 9. Вычисление рекурсивных функций с помощью динамического программирования. 10. Подсчёт количества вариантов с помощью динамического программирования. 11. Решение задач оптимизации с помощью динамического программирования	
Основы объектно-ориентированного программирования (16 часов)	Понятие об объектно-ориентированном программировании. Объекты и классы. Свойства и методы объектов. Объектно-ориентированный анализ. Разработка программ на основе объектно-ориентированного подхода. Инкапсуляция, наследование, полиморфизм.	<ul style="list-style-type: none"> Пояснять основные принципы объектно-ориентированного программирования. Проектировать и использовать простые классы объектов. Проектировать иерархии классов для описания предметной области.



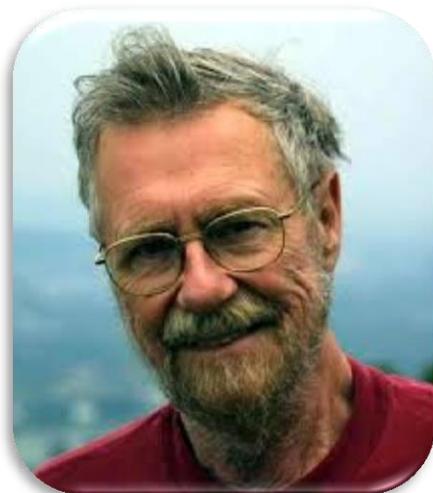
Алгоритм Дейкстры (1959 г.) для нахождения кратчайших путей (модификация поиска в ширину)



Каждой вершине графа $G(V, E)$ из множества вершин графа $V(1-6)$ поставлена в соответствие метка ребра из множества ребер графа E (минимальное известное расстояние от этой вершины до стартовой вершины 1).

Алгоритм работает пошагово на каждом шаге он «посещает» одну вершину и сохраняет пути с минимальными метками.

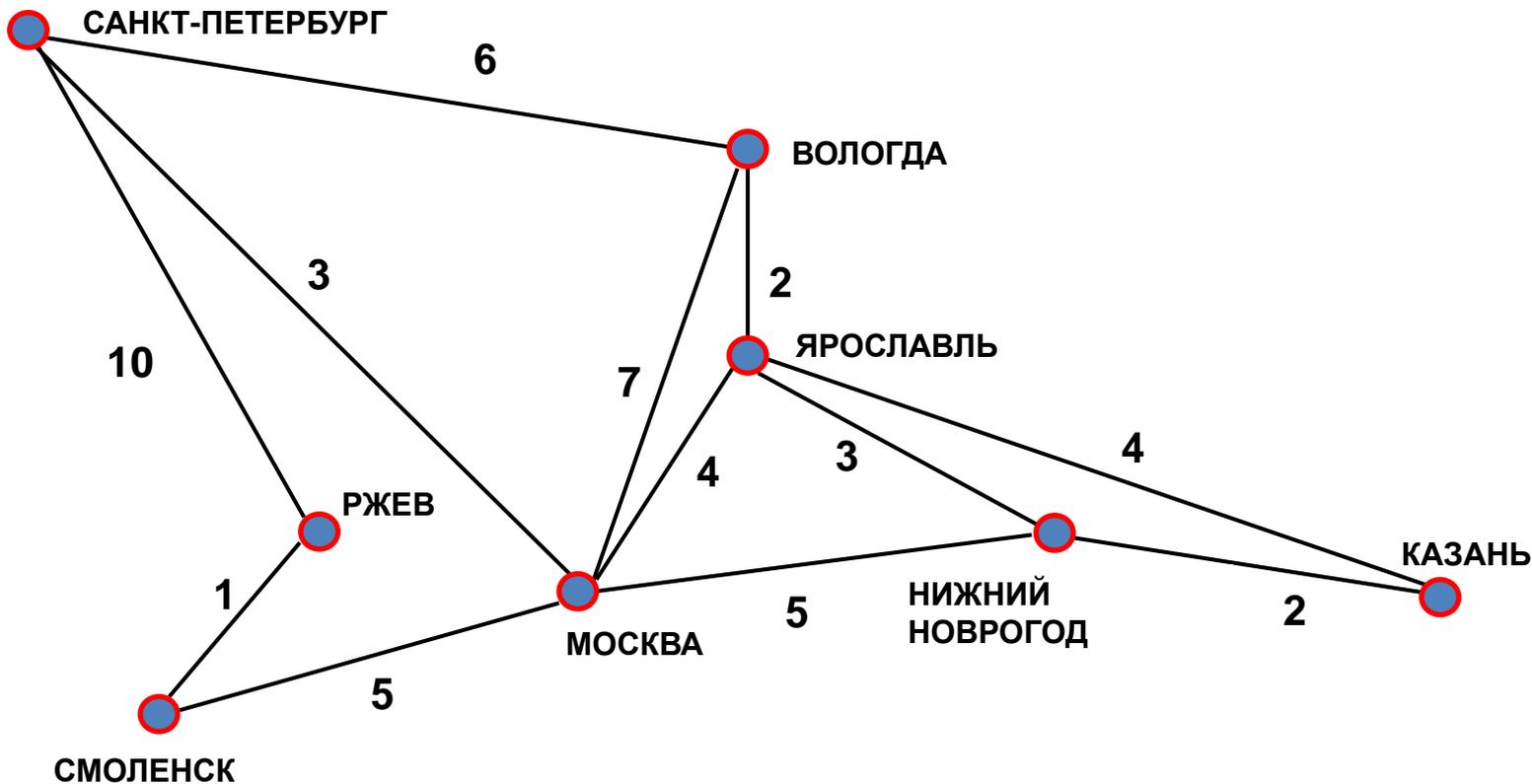
Работа алгоритма завершается, когда пройдены все вершины.



Эдсгер Вибе Дейкстра
(1930-2002)



Действие по алгоритмическим предписаниям





Действие по алгоритмическим предписаниям



Способ предписаний «по выбору» (путём усложнения операций)

Подготовительный этап:

- рисуем граф;
- изучаем маршрут – последовательность вершин и рёбер графа;
- расставляем вес путей (рёбер) между городами (вершинами);
- анализируем возможные оптимальные маршруты «вручную»;
- изучаем методы, необходимые для составления алгоритма в **Python** (метод нахождения пути от начальной вершины к конечной; метод минимизации общего веса маршрута по заданным критериям – его ценности; метод обхода вершин без повторения и пр.);
- выбираем модули, объекты и операторы в **Python** (sys, input, graph, init и др.), обеспечивающие программную реализацию необходимых методов;
- изучаем программные возможности отобранных модулей, объектов и операторов в **Python**.

Этап составления «схемы» (алгоритма) решения задачи:

- задаём вершины и весовые коэффициенты рёбер графа (в текстовой и цифровой форме);
- вводим «вручную» начальный и конечный пункты маршрута;
- обходим вершины с подсчётом суммарных весовых коэффициентов по рёбрам;
- отмечаем вершины как «посещённые»;
- сравниваем соседние весовые коэффициенты для пройденных путей и оставляем путь с минимальным суммарным весом;
- выводим результат на экран.

Этап написания программы (программных кодов) решения задачи.

Этап испытания и корректировки программы.

Этап изменения условий или расширения (усложнения) задачи:

- изменяем начальный и конечный пункты маршрута;
- изменяем весовые коэффициенты рёбер графа.....



```
IDLE Shell 3.12.3
File Edit Shell Debug Options Window Help
Python 3.12.3 (tags/v3.12.3:f6650f9, Apr 9 2024, 14:05:25) [M
SC v.1938 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more in
formation.
>>>
= RESTART: C:\Users\olgak\OneDrive\Рабочий стол\Путешествие по
России.py
Введите пункт отправления: Санкт-Петербург
Введите пункт назначения: Казань
Найден следующий лучший маршрут с ценностью 10.
Санкт-Петербург -> Москва -> Нижний Новгород -> Казань
>>>
==== RESTART: C:\Users\olgak\OneDrive\Рабочий стол\Путешествие
по России.py ====
Введите пункт отправления: Санкт-Петербург
Введите пункт назначения: Казань
Найден следующий лучший маршрут с ценностью 7.
Санкт-Петербург -> Вологда -> Ярославль -> Казань
>>>
Ln: 15 Col: 0
```

Санкт-Петербург – Вологда: 6

Санкт-Петербург – Вологда: 1



Заключение

- Более эффективное **управление учебной деятельностью** учащихся (индивидуализация обучения, сочетание с коллективной работой, рациональное использование всего комплекса средств обучения).
- Облегчается **процесс усвоения УУД** за счёт постепенного накопления запаса частных познавательных навыков и умений, формирующих УУД.
- Обеспечивается концентрация внимания на каждом шаге усвоения действия, осознаётся **роль отдельных деталей** в целостном акте формирования УУД.
- **Ускоренное формирование УУД**, самостоятельности и активности учащихся в процессе усвоения учебного материала.
- За счёт осознанности и более глубокого понимания **повышается интерес и мотивация** к выполнению заданий.



СПАСИБО ЗА ВНИМАНИЕ