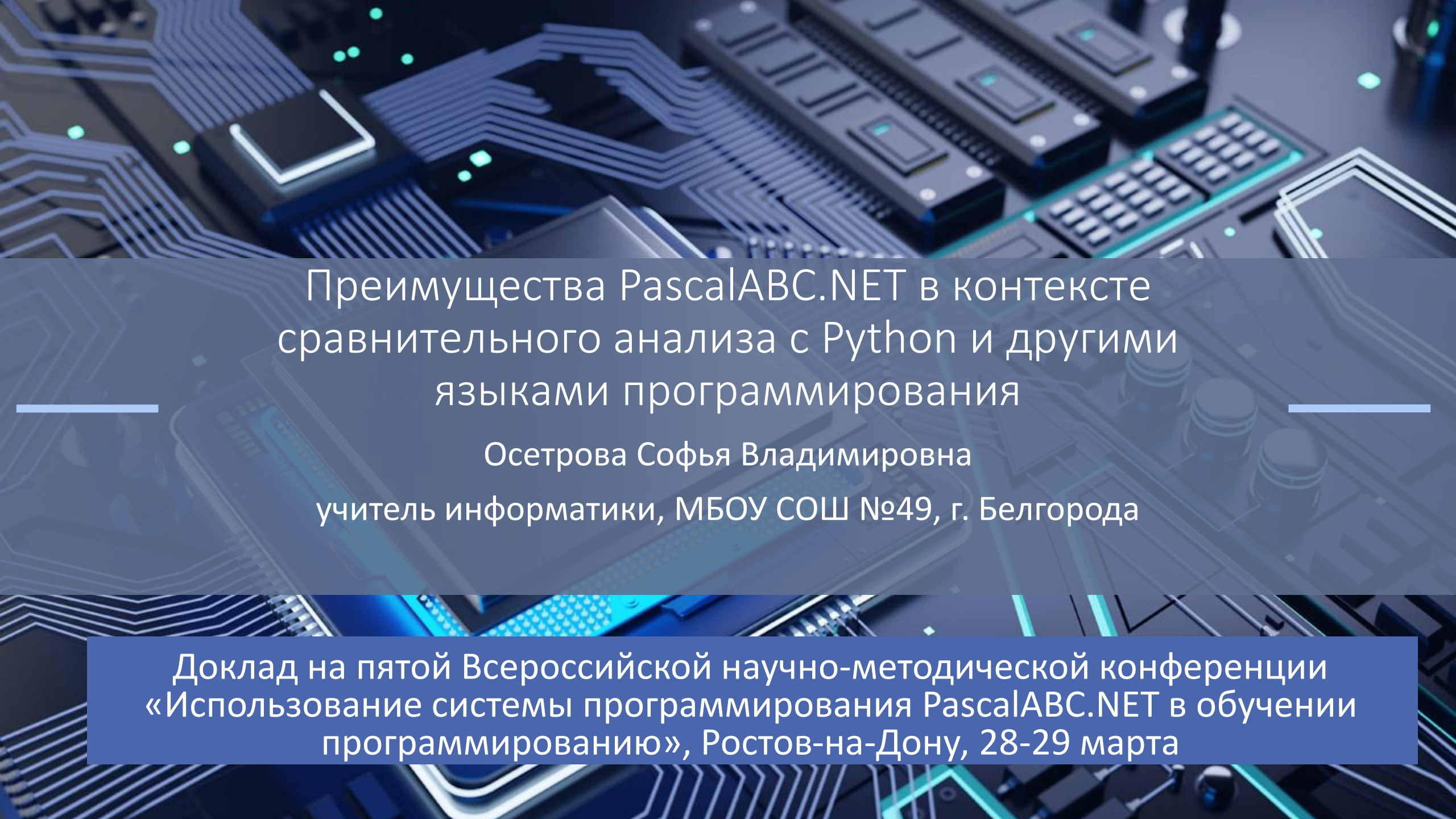




Преимущества PascalABC.NET в контексте сравнительного анализа с Python и другими языками программирования

Осетрова Софья Владимировна
учитель информатики, МБОУ СОШ №49, г. Белгорода

Доклад на пятой Всероссийской научно-методической конференции
«Использование системы программирования PascalABC.NET в обучении
программированию», Ростов-на-Дону, 28-29 марта

Основная цель

Выявление особенностей и преимуществ языка PascalABC.NET в сравнении с другими языками программирования



Мы стремимся ответить на следующие вопросы

Как изучение различных языков программирования влияет на развитие алгоритмического мышления у учащихся?



Какие преимущества предоставляет PascalABC.NET в контексте обучения программированию?



Как язык Паскаль способствует развитию логического мышления и навыков решения задач у учащихся, изучающих сразу несколько языков программирования?



Как использование PascalABC.NET в учебном процессе может повлиять на выбор профессии программистом в будущем и подготовить учащихся к требованиям современного рынка труда?



Обзор языков программирования

PascalABC.NET



PascalABC.NET — это современная реализация языка Паскаль, которая сочетает в себе простоту и мощные возможности. Язык отличается строгой типизацией, что позволяет учащимся лучше понимать структуру данных и логику программирования. Ясная синтаксическая структура делает его доступным для начинающих, а также способствует формированию правильных привычек в написании кода

Обзор языков программирования

Python



Python — это язык, который также широко используется в образовательных учреждениях благодаря своей простоте и читаемости. Он предлагает множество библиотек и инструментов, что делает его универсальным для решения различных задач. Однако, его динамическая типизация может усложнить понимание некоторых концепций программирования для начинающих.

Обзор языков программирования

КуМир



Кумир — это язык, разработанный для обучения программированию в России. Он сочетает в себе элементы Паскаля и других языков, что делает его удобным для начинающих. Однако, его популярность и использование ограничены в сравнении с другими языками

Обзор языков программирования

Scratch



Scratch — это визуальный язык программирования, ориентированный на детей. Он позволяет создавать программы с помощью перетаскивания блоков, что делает процесс обучения интерактивным и увлекательным. Однако, его ограниченные возможности могут не подготовить учащихся к более сложным языкам программирования.

Введение

В ходе нашего доклада мы проанализируем, как каждый из этих языков программирования может способствовать развитию алгоритмического мышления у учащихся. Особое внимание будет уделено PascalABC.NET, который, благодаря своей простоте и строгой типизации, создает прочную основу для понимания программирования и подготовки к будущей профессии программиста. Мы уверены, что результаты нашего исследования помогут педагогам выбрать наиболее эффективные подходы к обучению программированию в школе.

Начальная школа

Компьютерная грамотность



Введение в программирование на Scratch

Внеурочная деятельность

МОДУЛЬ 5. АЛГОРИТМЫ

M5Y1(22). Последовательность действий

M5Y2(23). Алгоритмы

M5Y3(24). Свойства алгоритмов

M5Y4(25). Подведение итогов модуля

МОДУЛЬ 4. АЛГОРИТМЫ И ЛОГИКА

M4Y1(18). Введение в логику

M4Y2(19). Истинность простых высказываний

M4Y3(20). Алгоритм и его свойства

M4Y4(21). Линейные алгоритмы. Усложнение.

M4Y5(22). Подведение итогов модуля

МОДУЛЬ 5. АЛГОРИТМЫ. БЛОК-СХЕМЫ

M5Y1(24). Алгоритмы и языки программирования

M5Y2(25). Блок-схемы

M5Y3(26). Циклические алгоритмы.

M5Y4(27). Блок-схема циклического алгоритма

M5Y5(28). Проектный урок. Рисуем блок-схему

M5Y6(29). Подведение итогов модуля

МОДУЛЬ 5. АЛГОРИТМЫ 2

M5Y1(23). Scratch. Знакомство

M5Y2(24). Scratch. Скрипты

M5Y3(25). Scratch. Циклы

M5Y4(26). Scratch. Условия. Организация движения

M5Y5(27). Scratch. Повороты и Вращение

M5Y6(28). Подведение итогов модуля

5-6 класс



11	Алгоритмы и исполнители
12	Алгоритмы и исполнители
13	Работа в среде программирования
14	Работа в среде программирования
15	Работа в среде программирования
16	Работа в среде программирования
17	Работа в среде программирования
18	Работа в среде программирования
19	Работа в среде программирования
20	Контрольная работа по теме «Алгоритмизация и основы программирования»

Урок

5 класс - 10 часов

6 класс – 12 часов

Основы программирования

Внеурочная деятельность



12	Основные алгоритмические конструкции.
13	Среда текстового программирования.
14	Управление исполнителем (исполнитель Чертежник).
15	Управление исполнителем (исполнитель Чертежник).
16	Циклические алгоритмы. Переменные.
17	Практическая работа №4. Разработка программ в среде текстового программирования, реализующих простые вычислительные алгоритмы. «Кумир»
18	Практическая работа №5. Разработка программ для управления исполнителем в среде текстового программирования с использованием циклов
19	Практическая работа №6. Разработка диалоговых программ в средетекстового программирования.
20	Вспомогательные алгоритмы. Разбиение задачи на подзадачи, использование вспомогательных алгоритмов (процедур). Процедуры с параметрами.
21	Практическая работа №7. Разработка программ для управления исполнителем в среде текстового программирования с использованием вспомогательных алгоритмов (процедур).
22	Практическая работа №8. Разработка программ для управления исполнителем в среде текстового программирования, в том числе с использованием вспомогательных алгоритмов (процедур) с параметрами.
23	Обобщение и систематизация знаний по темам: «Основные алгоритмические конструкции». «Вспомогательные алгоритмы». Проверочная работа.

МОДУЛЬ 3. ЗНАКОМСТВО СО СРЕДОЙ ПРОГРАММИРОВАНИЯ SCRATCH

МЗУ1(17). Знакомство с алгоритмами, способы записи алгоритмов

МЗУ2(18). Введение в Scratch: интерфейс, настройка фонов, свойства спрайта

МЗУ3(19). Среда Scratch: скрипты

МЗУ4(20). Циклические алгоритмы

МЗУ5(21). Разветвляющиеся алгоритмы

МЗУ6(22). Повороты

МЗУ7(23). Повороты и движение

МЗУ8(24). Установка начальных позиций: свойства, внешность. Анимация

МЗУ9(25). Подведение итогов модуля

МОДУЛЬ 3. SCRATCH. Логика

МЗУ1(17). Блок-схема как графическая модель алгоритма

МЗУ2(18). Оператор ветвления «если»
МЗУ3(19). Оператор ветвления «если ... , то ... , иначе ... »

МЗУ4(20). Логические операторы

МЗУ5(21). Цикл с условием

МЗУ6(22). Отработка: операторы ветвления и логические операторы

МЗУ7(23). Переменные

МЗУ8(24). Переменные: Закрепление

МЗУ9(25). Практикум по решению задач

МЗУ10(26). Подведение итогов модуля

7-9 класс

Основы программирования на Python Внеурочная деятельность



МОДУЛЬ 2. ОСНОВЫ ЯЗЫКА ПРОГРАММИРОВАНИЯ PYTHON

M2Y1(7). Современные языки программирования

M2Y2(8). Линейные алгоритмы в Python

M2Y3(9). Переменные в Python

M2Y4(10). Ввод данных

M2Y5(11). Типы данных

M2Y6(12). Вычисления

M2Y7(13). Ветвление в Python

M2Y8(14). Множественное ветвление

M2Y9(15). Решение задач на ветвление

M2Y10(16). Проект «Чат-бот»

M2Y11(17). Подведение итогов модуля

МОДУЛЬ 3. ЦИКЛЫ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

M3Y1(18). Логика, логические выражения

M3Y2(19). Логические выражения в Python

M3Y3(20). Логические операторы в Python

M3Y4(21). Цикл while

M3Y5(22). Цикл for

M3Y6(23). Вычисление суммы последовательностей

M3Y7(24). Практикум решению задач

M3Y8(25). Задачи на деление с остатком

M3Y9(26). Проект «Максимум и минимум»

M3Y10(27). Подведение итогов модуля

7 класс
21 час

МОДУЛЬ 2. ГРАФИЧЕСКИЙ МОДУЛЬ TURTLE В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

M2Y1(9). Подключение модуля turtle. Основные команды
M2Y2(10). Создание многоугольников. Цикл while

M2Y3(11). Цикл со счётчиком

M2Y4(12). Координаты

M2Y5(13). Логические операторы

M2Y6(14). Объекты и методы

M2Y7(15). Работа со списками

M2Y8(16). Подведение итогов модуля

8 класс
20 часов

МОДУЛЬ 3. ГРАФИЧЕСКИЙ МОДУЛЬ TURTLE В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

M3Y1(17). Понятие функции

M3Y2(18). Создание функции
M3Y3(19). Глобальные и локальные переменные

M3Y4(20). Фракталы
M3Y5(21). Понятие события. Объект «экран»

M3Y6(22). Логические операторы в Python

M3Y7(23). События мыши

M3Y8(24). События клавиатуры
M3Y9(25). Нахождение расстояния между объектами

M3Y10(26). Рекурсия и фракталы

M3Y11(27). Создание интерактивной игры

M3Y12(28). Подведение итогов модуля

МОДУЛЬ 2. СТРУКТУРЫ ДАННЫХ

M2Y1(7). Понятие базы данных и системы управления базами данных (СУБД)

M2Y2(8). Структурированные и неструктурированные данные

M2Y3(9). Создание и действия над строками

M2Y4(10). Создание списков

M2Y5(11). Действия над элементами списка

M2Y6(12). Циклический просмотр списка

M2Y7(13). Сортировка списков

M2Y8(14). Сумма элементов списка

M2Y9(15). Обработка списков

M2Y10(16). Понятие словаря

M2Y11(17). Подведение итогов модуля

МОДУЛЬ 3. СПИСКИ И СЛОВАРИ В ЯЗЫКЕ ПРОГРАММИРОВАНИЯ PYTHON

M3Y1(18). Создание словаря

M3Y2(19). Работа с элементами словаря

M3Y3(20). Решение задачи на подсчёт с
M3Y4(21). Решение задачи по
вычислению кэшбэка

M3Y5(22). Подведение итогов модуля

9 класс
15 часов

7-9 класс (1 час в неделю)

13	Понятие алгоритма. Исполнители алгоритмов
14	Свойства алгоритма. Способы записи алгоритма
15	Алгоритмическая конструкция «следование». Линейный алгоритм
16	Алгоритмическая конструкция «ветвление»: полная и неполная формы
17	Алгоритмическая конструкция «повторение»
18	Формальное исполнение алгоритма
19	Разработка несложных алгоритмов с использованием циклов для управления формальными исполнителями
20	Разработка несложных алгоритмов с использованием циклов и ветвлений для управления формальными исполнителями
21	Выполнение алгоритмов
22	Обобщение и систематизация знаний. Контрольная работа по теме «Исполнители и алгоритмы. Алгоритмические конструкции»

23	Язык программирования. Система программирования
24	Переменные. Оператор присваивания
25	Программирование линейных алгоритмов
26	Разработка программ, содержащих оператор ветвления
27	Диалоговая отладка программ
28	Цикл с условием
29	Цикл с переменной
30	Обработка символьных данных
31	Обобщение и систематизация знаний по теме «Язык программирования»
32	Анализ алгоритмов. Определение возможных результатов работы алгоритма при заданном множестве входных данных
33	Анализ алгоритмов. Определение возможных входных данных, приводящих к данному результату



8 класс
21 час
Урок



9 класс
27 часов



	Управление и алгоритмы (12 часов)
2.	Кибернетика. Кибернетическая модель управления
3.	Понятие алгоритма и его свойства. Входной контроль
4.	Линейные, ветвящиеся и циклические алгоритмы. Языки для записи алгоритмов (язык блок-схем, учебный алгоритмический язык)
5.	Исполнитель алгоритмов: назначение, среда исполнителя, система команд исполнителя, режимы работы
6.	Структурная методика алгоритмизации. Практическое задание: «Работа с учебным исполнителем алгоритмов». ТБ
7.	Практическое задание: «Составление линейных алгоритмов управления исполнителем». ТБ
8.	Вспомогательные алгоритмы. Метод пошаговой детализации. Практическое задание: «Использование вспомогательных алгоритмов (процедур, подпрограмм)». ТБ
9.	Практическое задание: «Составление циклических алгоритмов управления исполнителем». ТБ
10.	Практическое задание: «Составление циклических алгоритмов управления исполнителем». ТБ
11.	Практическое задание: «Составление ветвящихся алгоритмов управления исполнителем». ТБ
12.	Практическое задание: «Составление алгоритмов со сложной структурой». ТБ
13.	Текущий контроль по теме: «Управление и алгоритмы»

	Введение в программирование (15 часов)
14.	Языки программирования высокого уровня (ЯПВУ), их классификация
15.	Представление данных в программе. Алгоритмы работы с величинами: константы, переменные, понятие типов данных, ввод и вывод данных
16.	Структура программы на языке Паскаль. Правила записи основных операторов: присваивания, ввода, вывода
17.	Этапы решения задачи с использованием программирования: постановка, формализация, алгоритмизация, кодирование, отладка, тестирование. Отказы. Практическое задание: «Знакомство с системой программирования на языке Паскаль; ввод, трансляция и исполнение данной программы». ТБ
18.	Практическое задание: «Разработка и исполнение линейных программ». ТБ
19.	Правила записи основных операторов: ветвление
20.	Практическое задание: «Разработка и исполнение ветвящихся программ». ТБ
21.	Практическое задание: «Разработка и исполнение ветвящихся программ». ТБ
22.	Текущий контроль по теме: «Введение в программирование»
23.	Правила записи основных операторов: цикл
24.	Практическое задание: «Разработка и исполнение циклических программ». ТБ
25.	Практическое задание: «Разработка и исполнение циклических программ». ТБ
26.	Структурированный тип данных – массив. Способы описания и обработки массивов
27.	Практическое задание: «Программирование обработки массивов». ТБ
28.	Практическое задание: «Программирование обработки массивов». ТБ
29.	Текущий контроль по теме: «Введение в программирование»

7-9 класс (2 часа в неделю)

26	Понятие алгоритма. Исполнители алгоритмов
27	Свойства алгоритма
28	Способы записи алгоритма
29	Алгоритмические конструкции. Конструкция «следование». Линейный алгоритм
30	Знакомство с исполнителем
31	Конструкция «ветвление»: полная и неполная формы
32	Простые и составные условия. Создание алгоритмов с использованием ветвлений для управления исполнителем
33	Конструкция «повторение»: циклы с заданным числом повторений
34	Конструкция «повторение»: с условием выполнения
35	Переменная. Конструкция «повторение»: с переменной цикла
36	Вспомогательные алгоритмы
37	Вспомогательные алгоритмы с параметрами
38	Создание и выполнение на компьютере алгоритмов с использованием вспомогательных алгоритмов для управления исполнителем
39	Создание и выполнение на компьютере алгоритмов для управления исполнителем Робот
40	Анализ алгоритмов для исполнителей. Синтаксические и логические ошибки. Отказы
41	Контрольная работа по теме "Алгоритмы и программирование"

7 класс

16 часов

Урок

21	Язык программирования. Система программирования
22	Целые, вещественные и символьные переменные
23	Оператор присваивания. Арифметические выражения
24	Операции с целыми числами
25	Проверка делимости одного целого числа на другое
26	Операции с вещественными числами. Встроенные функции
27	Случайные (псевдослучайные) числа
28	Ветвления
29	Нахождение минимума и максимума из двух, трёх и четырёх чисел
30	Составные условия
31	Контрольная работа по теме «Начала программирования»
32	Решение квадратного уравнения, имеющего вещественные корни
33	Логические переменные. Диалоговая отладка программ
34	Цикл с условием
35	Алгоритм Евклида для нахождения наибольшего общего делителя двух натуральных чисел
36	Разбиение записи натурального числа в позиционной системе с основанием, меньшим или равным 10, на отдельные цифры
37	Разложение натурального числа на простые сомножители
38	Цикл с переменной. Алгоритм проверки натурального числа на простоту
39	Анализ алгоритмов. Определение возможных результатов работы алгоритма при заданном множестве входных данных
40	Анализ алгоритмов. Определение возможных входных данных, приводящих к данному результату
41	Обработка потока данных: вычисление количества, суммы
42	Обработка потока данных: вычисление среднего арифметического
43	Вычисление минимального и максимального значений элементов последовательности
44	Вычисление значений элементов последовательности, удовлетворяющих заданному условию
45	Обработка символьных данных. Посимвольная обработка

	строка
46	Поиск в символьных строках
47	Подсчёт частоты появления символа в строке
48	Встроенные функции для обработки строк
49	Табличные величины (массивы). Одномерные массивы
50	Заполнение числового массива случайными числами, в соответствии с формулой или путём ввода чисел
51	Нахождение суммы элементов массива
52	Линейный поиск заданного значения в массиве
53	Контрольная работа по теме «Программирование»
54	Подсчёт элементов массива, удовлетворяющих заданному условию
55	Нахождение минимального (максимального) элемента массива
56	Понятие о сложности алгоритмов

8 класс
36 часов



	Алгоритмизация и программирование
16.	Этапы решения задачи на компьютере. Задача о пути торможения автомобиля
17.	Решение задачи на компьютере
18.	Одномерные массивы целых чисел. Описание массива. Использование циклов
19.	Различные способы заполнения и вывода массива
20.	Вычисление суммы элементов массива
21.	Последовательный поиск в массиве
22.	Сортировка массива
23.	Решение задач с использованием массивов
24.	Проверочная работа «Одномерные массивы»
25.	Последовательное построение алгоритма
26.	Разработка алгоритма методом последовательного уточнения для исполнителя Робот
27.	Вспомогательные алгоритмы. Исполнитель Робот
28.	Запись вспомогательных алгоритмов на языке Паскаль. Процедуры
29.	Функции
30.	Алгоритмы управления. Робототехника. Управление роботом
31.	Текущий контроль по теме: «Алгоритмизация и программирование»
32.	Обобщение и систематизация основных понятий темы «Алгоритмизация и программирование»

9 класс
17 часов

10-11 класс

	Введение
1.	Введение. ТБ и правила поведения в кабинете информатики
	Структура программы на языке Паскаль
2.	Знакомство с программой на языке Паскаль
3.	Переменные и их имена
4.	Команда присваивания
5.	Операторы write и writeln. Процедуры вывода информации
6.	Операторы read и readln. Процедуры ввода информации
	Ввод и выполнение программ
7.	Императивные (процедурные) среды программирования: Turbo Pascal, PascalABC. Система программирования PascalABC.NET
8.	Lazarus - свободная среда разработки программного обеспечения на языке Object Pascal для компилятора Free Pascal
9.	Императивный, структурированный, объектно-ориентированный, высокоуровневый язык программирования Delphi
	Арифметические операции с целыми числами. переменные целого типа
10.	Операции с целыми числами и переменными целого типа в языке Паскаль
11.	Комментарии
12.	Операции с целыми числами и переменными целого типа в языке Паскаль. Решение задач
	Условный оператор
13.	Условный оператор if... then... else
14.	Блок-схемы, изображающие условные операторы
15.	Вещественный тип real в операторах if... then... else. Решение задач
16.	Формы записи вещественных чисел
17.	Арифметические операции с вещественными переменными. Решение задач
18.	Форматированный вывод информации
19.	Форматированный вывод информации. Решение задач
20.	Некоторые стандартные функции и процедуры, применимые к целым типам. Логический тип Boolean
21.	Некоторые стандартные функции и процедуры, применимые к целым типам. Логический тип Boolean. Решение задач
22.	Вычисление значений функции. Решение задач
	Операторы цикла. Цикл с предусловием
23.	Что такое цикл
24.	Целый тип Longint
25.	Условия в программах с циклами. Решение задач
26.	Досрочное прерывание цикла. Метка. Оператор безусловного перехода goto
27.	Преобразование типов. Решение задач
28.	Циклы в циклах
29.	Циклы в циклах. Решение задач
30.	Системы счисления
31.	Системы счисления. Решение задач
	Повторение
32.	Повторение. Программы линейной структуры
33.	Повторение. Условный оператор
34.	Повторение. Операторы цикла. Цикл с предусловием

10 класс

11 класс

	Операторы цикла. Цикл с постусловием
1.	Цикл с последующим условием. ТБ и правила поведения в кабинете информатики
2.	Цикл с последующим условием. Решение задач
3.	Различие между циклом while и циклом repeat
4.	Программы с совместным использованием циклов repeat и while... do...
5.	Программы с совместным использованием циклов repeat и while... do.... Решение задач
6.	Разные задачи. Решение задач
7.	Разные задачи. Решение задач
	Операторы цикла. Циклы с параметром
8.	Оператор цикла for... to... do...
9.	Оператор цикла for... to... do... Решение задач
10.	Оператор цикла for... downto... do...
11.	Оператор цикла for... downto... do... Решение задач
12.	Разные задачи. Решение задач
13.	Преобразование типов
14.	Преобразование типов. Решение задач
15.	Операторы цикла. Решение задач
	Подпрограммы. Рекурсия
16.	Дедуктивный метод программирования. Процедуры
17.	Дедуктивный метод программирования. Вызов процедуры из программы
18.	Дедуктивный метод программирования. Решение задач
19.	Дедуктивный метод программирования. Решение задач
20.	Вызов процедуры из другой процедуры
21.	Вызов процедуры из другой процедуры. Решение задач
22.	Рекурсия
23.	Рекурсия. Решение задач
24.	Функции

Внеурочная деятельность «Программирование»

1 час в неделю



25.	Функции. Решение задач
26.	Решение уравнений с одной переменной методом половинного деления. Отделение корней
27.	Решение уравнений с одной переменной методом половинного деления. Решение задач
28.	Разные задачи. Решение задач
	Повторение
29.	Повторение. Операторы цикла. Цикл с постусловием
30.	Повторение. Операторы цикла. Цикл с параметром
31.	Повторение. Подпрограммы. Рекурсия
32.	Резерв
33.	Резерв
34.	Резерв

10-11 класс

МОДУЛЬ 1. ОБРАБОТКА И ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ	МОДУЛЬ 2. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ
M1 Y1(1). Функции электронных таблиц для обработки больших массивов данных	M2Y1(18). Алгоритмы обработки текстовых данных M2Y2(19). Анализ текстовых данных. Динамический подход M2Y3(20). Метод двух указателей для строк M2Y4(21). Анализ текстовых файлов. Частотный словарь M2Y5(22). Генерация слов заданного алфавита M2Y6(23). Генерация всех слов, удовлетворяющих условию M2Y7(24). Обработка целочисленных данных: практикум
M1 Y2(2). Визуализация данных	M2Y8(25). Числовые автоматы
M1 Y3(3). Возможные направления исследований. Открытые данные. Формат CSV	M2Y9(26). Регулярные выражения
M1 Y4(4). Обработка массивов данных в электронных таблицах	M2Y10(27). Регулярные выражения
M1 Y5(5). Обработка данных в файлах формата CSV	M2Y11(28). Регулярные выражения
M1 Y6(6). Корреляционный анализ	M2Y12(29). Решение задач на шифрование M2Y13(30). Практикум по решению уравнений в электронных таблицах M2Y14(31). Решение финансовых задач в электронных таблицах M2Y15(32). Исследование графиков функций в полярных координатах M2Y16(33). Практикум по вычислению длины кривой M2Y17(34). Практикум по вычислению площадей фигур
M1 Y7(7). Восстановление зависимостей	
M1 Y8(8). Введение. Задачи анализа данных	
M1 Y9(9). Подготовка данных	
M1 Y10(10). Базы данных в электронных таблицах M1 Y11(11). Базы данных в электронных таблицах	
M1 Y12(12). Среда Jupyter Notebook	
M1 Y13(13). Пакет Pandas	
M1 Y14(14). Статистические характеристики данных	
M1 Y15(15). Визуализация и преобразование данных	
M1 Y16(16). Визуализация и преобразование данных	
M1 Y17. Итоговый проект	

10 класс

Практическая информатика

элективный курс

1 час в неделю

МОДУЛЬ 1. ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И МАШИННОЕ ОБУЧЕНИЕ
M1 Y1(1). Алгоритмы обработки текстовых данных
M1 Y2(2). Анализ текстовых данных. Динамический подход
M1 Y3(3). Метод двух указателей для строк M1 Y4(4). Анализ текстовых файлов. Частотный словарь
M1 Y5(5). Генерация слов заданного алфавита
M1 Y6(6). Генерация всех слов, удовлетворяющих условию M1 Y7(7). Обработка целочисленных данных: практикум
M1 Y8(8). Числовые автоматы
M1 Y9(9). Регулярные выражения
M1 Y10(10). Регулярные выражения
M1 Y11(11). Регулярные выражения M1 Y12(12). Решение задач на шифрование M2 Y13(30). Практикум по решению уравнений в электронных таблицах M2 Y14(31). Решение финансовых задач в электронных таблицах M2 Y15(32). Исследование графиков функций в полярных координатах M2 Y16(33). Практикум по вычислению длины кривой M2 Y17(34). Практикум по вычислению площадей фигур

МОДУЛЬ 2. МОДЕЛИРОВАНИЕ И ОПТИМИЗАЦИЯ
M3Y1(18). Среда Scilab
M3Y2(19). Среда Scilab: построение графиков
M3Y3(20). Построение трёхмерных графиков в среде Scilab
M3Y4(21). Среда Scilab: решение уравнений
M3Y5(22). Моделирование в среде Xcos пакета Scilab
M3Y6(23). Моделирование физических моделей
M3Y7(24). Моделирование систем управления
M3Y8(25). Моделирование популяций
M3Y9(26). Создание графических приложений в среде Scilab
M3Y10(27). Решение задач оптимизации с помощью электронных таблиц
M3Y11(28). Решение задач оптимизации финансовых процессов
M3Y12(29). Решение задач оптимизации в среде Scilab
M3Y13(30). Оптимальный выбор: задачи линейного программирования
M3Y14(31). Оптимальный выбор: динамическое программирование
M3Y15(32). Оптимальная упаковка
M3Y16(33). Оптимизация вычислительных процессов: практикум
M3Y17(34). Оптимизация вычислительных процессов: практикум

11 класс

Введение программирования в школьном курсе

Класс	Язык программирования (урок)	Язык программирования (внеурочная деятельность)
1, 2, 3, 4		Scratch
5, 6	Кумир	Scratch
7,8,9 (база)	Кумир, Паскаль	Python
7,8,9 (профиль)	Кумир, Паскаль	Python
10, 11 (база)	Python	Паскаль
10, 11 (профиль)	Python	Паскаль

Урок «Разработка программ, содержащих операторы ветвления», 8 класс

Время	I) Постановка темы урока		Самооценивание
2 мин 9.55-9.57		Тема урока: _____ _____ _____	1 балл, если тема сформулирована верно:
	Кто автор картины? Как она называется?	_____ _____ _____	1 балл, если дан хотя бы один верный ответ:

Время	II) Постановка целей урока (Сформулируйте свою цель, кратко запишите высказывания одноклассников)		Самооценивание
2 мин 9.57-9.59	Паша:	Кирилл:	1 балл, при активной работе на этапе: цель сформулирована, высказывания одноклассников выписаны:
	Денис:	Настя:	
	Олег:	Ира:	
	Аня:	Миша:	
	Дима:	Никита:	

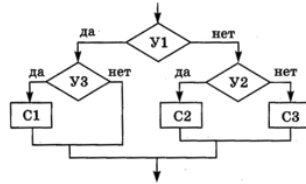
Актуализация знаний. Повторение теории, формулировка основных определений

Время	Облако слов	Основные понятия, определения	Самооценивание
2 мин 9.59-10.01			За верный ответ- 1 балл
			За верный ответ- 1 балл
			За верный ответ- 1 балл

Проверка Д/З Структура алгоритмов и программ

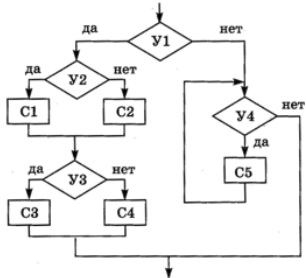
(IV) Проверка Д/З Структура алгоритмов и программ
Соедини стрелочками соответствующие
блок-схему и программу

Время
3 мин
10.01-10.04



```

If <Y1>
Then
Begin
  If <U2>
  Then <C1>
  Else <C2>;
  If <Y3>
  Then <C3>
  Else <C4>
End
Else While <Y4> Do <C5>;
  
```

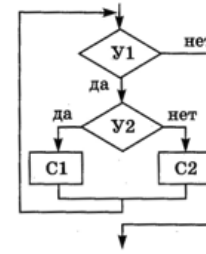
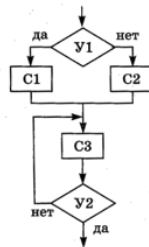


```

If <Y1>
Then
  If <Y2>
  Then <C1>
  Else
    If <Y3>
    Then <C2>
    Else <C3>;
  End
End
  
```

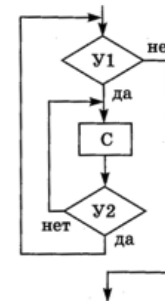
```

While <Y1> Do
  Repeat
    <C1>
  Until <Y2>;
End
  
```



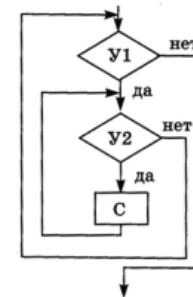
```

If <Y1>
Then <C1>
Else <C2>;
Repeat
  <C3>
Until <Y2>;
  
```



```

While <Y1> Do
  While <Y2> Do
    <C>;
  End
End
  
```



```

While <Y1> Do
  If <Y2>
  Then <C1>
  Else <C2>;
End
  
```

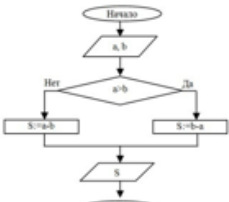
Самооценивание 1 балл за каждые два верных ответа
Максимальное количество 3 балла

Практикуемся в решении задач на ветвление первого уровня сложности

Задание	V) Практикуемся в решении задач на ветвление первого уровня сложности Время 5 мин 10.04-10.09			Самооценивание
Зачеркни лишнее	Program Zd5_1; Var a : integer; Begin ReadLn(a); If (a mod 2) = 0 Then WriteLn('Четное') Else WriteLn('Не четное') End.	Program Zd5_2; Var x, y : integer; Begin ReadLn(x,y); If x>y Then WriteLn('x - макс') Else WriteLn('y - мин') End.	Program Zd5_3; Var x, y : integer; Begin ReadLn(n); S:=0; For i:=1 To n Do S:=S+1; WriteLn(S) End.	Верный ответ – 1 балл
Составь условие				Верно составленное условие для задач Zd5_1 и Zd5_2 - 1 б. Для Zd5_3 еще 1 б.
Исправь ошибки	<pre> graph TD Start([Начало]) --> ReadA[/a/] ReadA --> Cond{a mod 5 = 0} Cond -- Да --> Out1[/a не четное/] Cond -- Нет --> Out2[/a четное/] Out1 --> End([Конец]) Out2 --> End </pre> <pre> graph TD Start([Начало]) --> ReadXY[/x, y/] ReadXY --> Cond{x > y} Cond -- Да --> Out1[/x - макс/] Cond -- Нет --> Out2[/x - мин/] Out1 --> End([Конец]) Out2 --> End </pre>			Все ошибки исправлены в первой блок-схеме – 1 балл Все ошибки исправлены во второй блок-схеме – еще 1 балл

Дополнительное задание Практикуемся в решении задач на ветвление первого уровня повышенной сложности

Дополнительное задание выполняется при наличии свободного времени!
Проверяется учителем в индивидуальном порядке, при работе за компьютером.

Задание	V) Дополнительное задание Практикуемся в решении задач на ветвление первого уровня повышенной сложности Время 5 мин 10.04-10.09			Критерии оценки
Зачеркни лишнее				Верный ответ – 1 балл
Составь условие				Верно составленное условие для задач Zd5_5 и Zd5_6 - 1 б. Для Zd5_4 еще 1 б.
Исправьте ошибки	<pre> Program Zd5_4; Var r, S, L : integer; Begin ReadLn(S); S:=Pi*Sqr(r); L:=2*Pi*r; WriteLn(L,S) End.</pre>	<pre> Program Zd5_5; Var a, b : integer; Begin ReadLn(a); If a>b Then WriteLn('Четное') Else WriteLn('Не четное'); WriteLn('S'); End.</pre>	<pre> Program Zd5_6; Var a : integer; S : real; Begin ReadLn(x,y); If x>y Then S:=a*2 Else S:=a/2; WriteLn(S) End.</pre>	За каждую верно составленную программу 1 балл Максимум – 3 балла

Решение задач второго уровня сложности. Работа за компьютером

VI) Решение задач второго уровня сложности. Работа за компьютером.

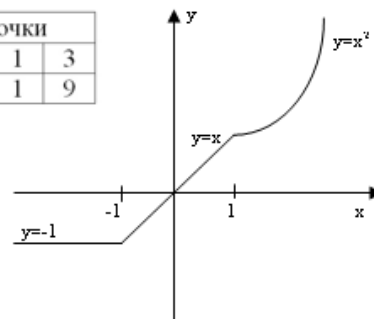
Выберите для решения одну из двух задач. Вторая задача повышенного уровня сложности. За правильно составленную систему уравнений можно получить дополнительный балл. Задачу решаем на компьютере. Решение проверяем самостоятельно по контрольным точкам.

Задача №6_1

$$y = \begin{cases} x^2 + 1, & \text{при } x \geq 1 \\ 2, & \text{при } -2 \leq x < 1 \\ -x, & \text{при } x < -2 \end{cases}$$

Контрольные точки					
x	-3	-2	0	1	5
y	3	2	2	2	26

Контрольные точки					
x	-5	-1	0	1	3
y	-1	-1	0	1	9



Задача №6_2

Самопроверка: 5 контрольных точек выдают верное значение – 3 балл;
3,4 точки – 2 балла;
1 точка – 1 балл.

Дополнительный балл, если для решения выбрана вторая задача и система составлена правильно.

Выписанная программа в опорный конспект +1 балл.

Еще один дополнительный балл можно заработать верно составленную блок-схему.

Максимум – 6 баллов.

Время: 7 мин
10.09-10.16

Программирование алгоритмов с ветвлением, уровень 2, повышенная сложность. Работа по группам

VII) Программирование алгоритмов с ветвлением, уровень 2, повышенная сложность. Работа по группам.

В группах распределите обязанности:

Время: 9 мин
10.09-10.25

- 1) Составление математической модели, словесного описания.
- 2) Составление блок-схемы (У доски!)
- 3) Запись программы (Копируем в папку Учитель)
- 4) Составление контрольных точек для задачи другого варианта, тестирование задачи второй команды
- 5) Совместно оцените работу другой группы: максимум два балла за каждый этап. (2 балла – этап выполнен верно, 1 балл – с ошибками, 0 баллов – нет решения.)
- 6) Баллы, которые выставили вам, распределите между участниками группы в соответствии с приложенными усилиями для решения задачи.

Вариант 1

Определите, имеется ли в заданном трехзначном числе четная цифра.

Вариант 2

Напишите программу, которая по заданным трем числам определяет, является ли сумма каких-либо двух из них положительной

Изучение нового оператора Case

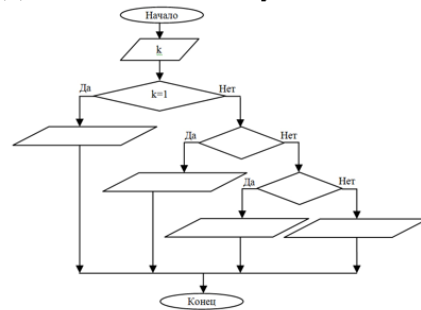
VIII) Изучение нового оператора Case.

Время 5
мин

Работа с учебником стр. 78-79 Пример 1, пример 2.

Задача №3 стр. 51 из практикума (только начальная школа)

Дополнить блок-схему:



Открыть в Pascal файл Case1.

Дополнить программу.

Обратите внимание - ввод данных осуществляется через файлы.

Запустите на исполнение

Program Case_;

Var

f1,f2 : text;

k : integer;

S : string;

Begin

Assign(f1,'input.txt');

Assign(f2,'output.txt');

reset(f1);

rewrite(f2);

Самооценивание блок
схема (2 балла – составлена
верно, 1 балл – с ошибками,
0 баллов – нет решения)

Откройте файл
C:\FPC\2.6.4\bin\output.txt
Если там записано «Privet,
vtoroklassnik» - программа
составлена верно – 1 балл.
Дополнительный балл, если
программу записали в
полный консект

ReadLn(f1,k);

Case k Of

1 : s:='Privet, pervoklassnik!';

2 :

3 :

Else

End;

WriteLn(f2,s);

Close(f1);

close(f2);

End.

Время	IX) Интеллектуальная игра «Алгоритмический код» Ответьте на вопрос: Да - 1; Нет - 0. Работа в парах. Подумайте, как вы распределите работу.		Да - 1; Нет - 0.
5 мин (10.30 - 10.35)	1) В результате выполнения программы: Повтори 4 [Вперед 50 Направо 90] Черепашка начертит квадрат	Исполнитель Черепашка перемещается на экране компьютера, оставляя след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существуют две команды: Вперед n (где n – целое число), вызывающая передвижение Черепашки на n шагов в направлении движения; Направо m (где m – целое число), вызывающая изменение движения на m градусов по часовой стрелке. Запись Повтори k [Команда 1 Команда 2 Команда 3] означает, что последовательность команд в скобках повторится k раз.	
	2) Фигура, которая получится в результате выполнения программы: Повтори 12 [Вперед 50 Направо 90] квадрат.		
	3) Фигура, которая получится в результате выполнения программы: Сместиться на (3, 2) Сместиться на (2, -3) Сместиться на (1, -5) пятиугольник	Исполнитель Чертёжник перемещается на координатной плоскости, оставляя след в виде линии. Чертёжник может выполнять команду Сместиться на (a, b) (где a, b – целые числа), перемещающую Чертёжника из точки с координатами (x, y) в точку с координатами (x + a, y + b). Если числа a, b положительные, значение соответствующей координаты увеличивается; если отрицательные, уменьшается. Запись Повтори k раз Команда1 Команда2 Команда3 Конец означает, что последовательность команд Команда1 Команда2 Команда3 повторится k раз.	
	4) В результате выполнения программы: Повтори 2 раз Сместиться на (3, 2) Сместиться на (2, -3) Сместиться на (1, -5) Конец получится пятиугольник		
	5) Кузнечик начал выполнение программы в точке 0 и закончил в точке 2. Затем программа была выполнена еще раз. Вернется ли он в точку 0?	Исполнитель «Кузнечик» «живет» на горизонтальной оси и умеет выполнять две команды Вперед 3 , Назад 2 . Повтори k раз Команда1 Команда2 Команда3 Конец означает, что последовательность команд Команда1 Команда2 Команда3 повторится k раз.	
	6) Кузнечик начал движение из точки 0. Повтори 2 раз Вперед 3, Назад 2, Вперед 3 Верно ли, что Кузнечик после выполнения программы оказался в точке 4?		

Интеллектуальная игра «Алгоритмический код»

	7) Попадет ли Робот из начальной точки (◇) в точку A в результате выполнения программы: вниз, влево, вниз.	По командам вверх, вниз, вправо, влево Робот перемещается на одну клетку в указанном направлении. При попытке Робота пройти через стену возникает ошибка в работе программы, выполнение которой при этом прерывается.					
	8) Попадет ли Робот из начальной точки (◇) в точку A в результате выполнения программы: влево, вверх, вправо.						
9) К числу 167 применили программу <u><=, x2, x2, <=,</u> В ответе будет 6?		Вычислитель умеет: <table border="1" data-bbox="1832 862 2226 918"><tr><td>x 2</td><td>Умножить на 2</td></tr><tr><td><=</td><td>Стереть последнюю цифру</td></tr></table>	x 2	Умножить на 2	<=	Стереть последнюю цифру	
x 2	Умножить на 2						
<=	Стереть последнюю цифру						
10) К числу 453 применили программу <u><=, x2, <=,</u> В ответе будет 9?							
Запиши получившийся двоичный код: _____			Код составлен верно – 3 балла				
Переведите в десятичную систему:			Верный ответ 16.				
Как этот год связан с жизнью ученого аль-Хорезми			Верный ответ 16.				

Подведение итогов

X) Дополнительно он-лайн тестирование

XI) Подведение итогов

Баллы:

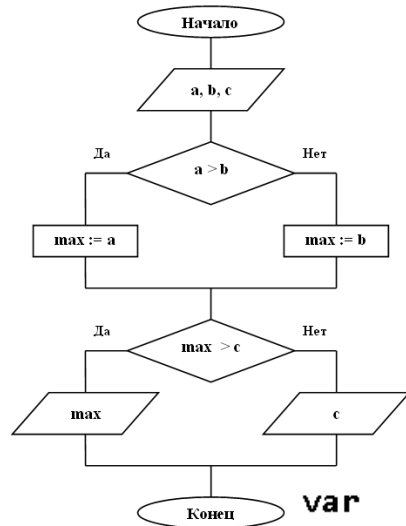
Оценка:

Д/З Закончить Задачу №3 стр.
51 из практикума.
Решить вторым способом,
используя оператор If.
Оптимизировать ввод



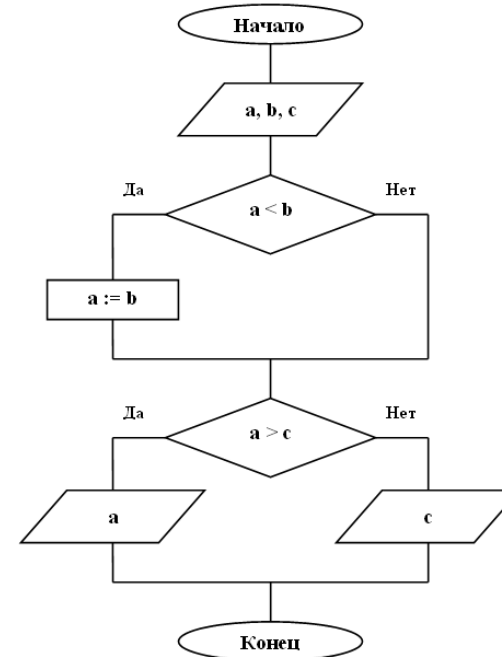
Время: 5 мин
10.35-10.40

Найти наибольшее число из трех. Если числа равны, то вывести любое из них



```
var
  a,b,c,max:integer;
begin
  read(a,b,c);
  if a > b then
    max:=a
  else
    max:=b;
  if max > c then
    write(max)
  else
    write(c);
end.
```

```
var
  a,b,c:integer;
begin
  read(a,b,c);
  if (a > b) and (a > c) then
    write(a)
  else
    if (b > a) and (b > c) then
      write(b)
    else
      write(c);
end.
```



```
var
  a,b,c:integer;
begin
  read(a,b,c);
  if a < b then
    a:=b;
  if a > c then
    write(a)
  else
    write(c);
end.
```

Найти наибольшее число из трех (Python)

```
if a > b:
    if a > c:
        print(a)
    else:
        print(c)
else:
    if b > c:
        print(b)
    else:
        print(c)
```

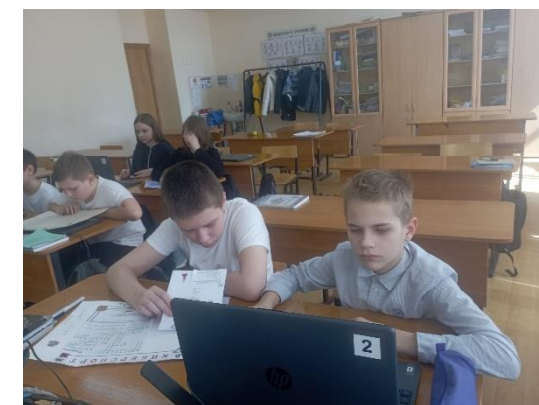
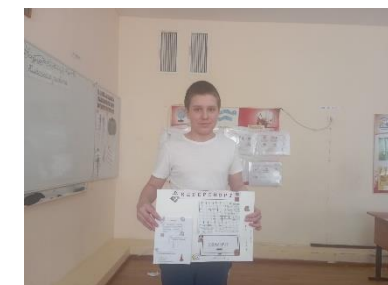
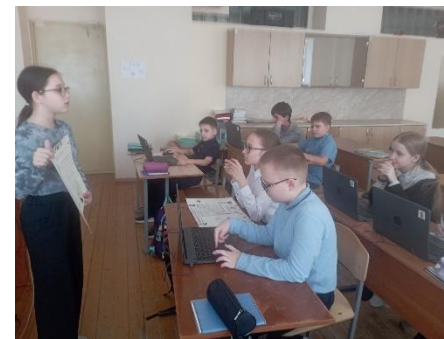
```
if a > c and a > b : print(a)
elif b > a and b > c : print(b)
else: print(c)
```

```
mx = max(a,b,c)
print(mx)
```

```
mx = a
if b > mx:
    mx = b
if c > mx:
    mx = c
print(mx)
```

```
m = sorted([a,b,c])
print(m[-1])
```

Наши уроки



Сравнительный анализ PascalABC.NET и Python



Одним из основных преимуществ PascalABC.NET является его строгая типизация. Это означает, что типы данных должны быть явно определены, что помогает учащимся лучше понимать структуру данных и логику программирования. Строгая типизация способствует выявлению ошибок на этапе компиляции, что позволяет учащимся быстрее осваивать основные концепции и избегать распространенных ошибок, связанных с неправильным использованием типов данных. В результате, учащиеся, изучающие PascalABC.NET, получают более глубокое понимание основ программирования, что является важным для их дальнейшего обучения.

Python?

С другой стороны, Python, хотя и является более гибким и мощным языком, может вводить учащихся в заблуждение из-за своей динамической типизации. В Python типы данных определяются во время выполнения программы, что может привести к трудностям в понимании того, как данные обрабатываются и хранятся. Это может затруднить процесс обучения для начинающих программистов, так как они могут не осознавать важность правильного выбора типов данных и их влияния на работу программы.



Простота синтаксиса и структурное программирование



Еще одним значительным преимуществом PascalABC.NET является его ясная и лаконичная синтаксическая структура. Язык был разработан с учетом удобства для начинающих, что позволяет учащимся сосредоточиться на логике программирования, а не на сложностях синтаксиса. Это особенно важно для младших школьников, которые только начинают знакомиться с программированием. Простота синтаксиса PascalABC.NET позволяет учащимся быстрее создавать рабочие программы и видеть результаты своих усилий, что, в свою очередь, повышает их мотивацию и интерес к обучению.

Python?

В отличие от этого, Python, хотя и имеет читаемый синтаксис, может быть более сложным для понимания из-за своей гибкости и множества доступных библиотек. Учащиеся могут столкнуться с трудностями при попытке разобраться в различных подходах к решению одной и той же задачи, что может привести к путанице и затруднениям в обучении.



Развитие логического мышления и навыков структурного программирования

Использование PascalABC.NET в учебном процессе также способствует развитию навыков структурного программирования и логического мышления. Структурное программирование, основанное на использовании функций и процедур, помогает учащимся разбивать сложные задачи на более простые подзадачи, что является важным навыком для решения проблем в программировании. Учащиеся, изучающие Pascal, учатся планировать и организовывать свой код, что является основой для успешного перехода к более сложным языкам и проектам.



Python?

В то время как Python также поддерживает структурное программирование, его гибкость может отвлекать учащихся от основ. Учащиеся могут начать использовать более сложные конструкции и библиотеки, не освоив базовые принципы, что может привести к недостаточному пониманию основ программирования



Заключение

В ходе нашего доклада мы провели сравнительный анализ языков программирования PascalABC.NET и Python, выделив ключевые преимущества языка Паскаль в контексте обучения программированию. Мы рассмотрели, как строгая типизация и простота синтаксиса PascalABC.NET способствуют более глубокому пониманию основ программирования, в то время как гибкость Python может вводить учащихся в заблуждение. Также мы обсудили, как использование PascalABC.NET помогает развивать навыки структурного программирования и логического мышления, что является важным для успешного перехода к более сложным языкам и проектам

Вывод

На основании проведенного анализа можно сделать вывод, что интеграция PascalABC.NET в учебный процесс может значительно повысить качество обучения программированию. Язык Паскаль, благодаря своей ясной структуре и строгой типизации, создает прочную основу для формирования алгоритмического мышления у учащихся, что является критически важным в их дальнейшем обучении и профессиональной деятельности





Рекомендации педагогам

- Введение в основы программирования через PascalABC.NET: Начинать обучение с PascalABC.NET, чтобы учащиеся могли сосредоточиться на базовых концепциях программирования, таких как переменные, циклы и условия, без излишней сложности.
- Комбинирование с другими языками: После освоения основ на PascalABC.NET, постепенно вводить более сложные языки, такие как Python, чтобы учащиеся могли увидеть, как различные языки решают одни и те же задачи.
- Практические проекты: Включение практических проектов, которые требуют применения изученных концепций, поможет учащимся закрепить знания и развить навыки решения реальных задач.
- Обратная связь и поддержка: Регулярное предоставление обратной связи и поддержка учащихся в процессе обучения помогут им преодолевать трудности и сохранять мотивацию.

Спасибо за внимание



Таким образом, наш доклад может стать полезным ресурсом для учителей, стремящихся улучшить процесс обучения и подготовить учащихся к успешной карьере в области информационных технологий. Мы уверены, что правильная интеграция PascalABC.NET в учебный процесс не только повысит качество образования, но и поможет формировать будущих профессионалов, готовых к вызовам современного мира.
