

Согласовано  
Руководитель Центра  
*С.И. Коренева*  
«01» 09 2021

Утверждено  
Директор школы  
*Г.В. Бакушина*  
«11» 09 2021



ПРОГРАММА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ  
«Информатика»

Учитель: Фирсова М.Н.

## 1. Пояснительная записка

Программа данного курса посвящена обучению школьников началам программирования на примере графического языка Scratch, а также умению работать с данными в текстовых документах. Занятия курса направлены на развитие мышления, логики, творческого потенциала учеников. Программа ориентирована на использование получаемых знаний для разработки реальных проектов. Курс содержит большое количество творческих заданий (именуемых Кейсами).

### Цель и задачи обучения

Целью изучения предмета «Информатика» является получение теоретических и практических знаний, умений и навыков в области современной информатики; формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира.

Для достижения поставленной цели необходимо решение следующих задач:

- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
- формирование информационной и алгоритмической культуры;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей;
- овладение важнейшими общеучебными умениями и уни-

версальными учебными действиями (формулировать цели деятельности, планировать ее, находить и обрабатывать необходимую информацию из различных источников, включая Интернет и др.).

## 2. Общая характеристика учебного предмета

Программа по предмету «Информатика» предназначена для изучения курса информатики учащимися основной школы.

Она включает в себя два блока:

- Программирование в Scratch
- Работа с текстовым процессором LibreOffice.org Writer

Важная задача изучения этих содержательных линий в курсе – добиться систематических знаний, необходимых для самостоятельного решения задач, в том числе и тех, которые в самом курсе не рассматривались. На протяжении первой части курса учащиеся изучают базовые основы программирования на примере графического языка Scratch.

### Технологии, используемые в образовательном процессе:

- Технологии традиционного обучения для освоения минимума содержания образования в соответствии с требованиями стандартов; технологии, построенные на основе объяснительно-иллюстративного способа обучения. В основе – информирование, просвещение обучающихся и организация их репродуктивных действий с целью выработки у школьников общеучебных умений и навыков.
- Технологии компьютерных практикумов.
- Игровые технологии.
- Тестовые технологии.
- Технологии реализации межпредметных связей в образова

тельном процессе.

- Технологии дифференцированного обучения для освоения учебного материала обучающимися, различающимися по уровню обучаемости, повышения познавательного интереса.
- Технология проблемного обучения с целью развития творческих способностей обучающихся, их интеллектуального потенциала, познавательных возможностей. Обучение ориентировано на самостоятельный поиск результата, самостоятельное добывание знаний, творческое, интеллектуально-познавательное усвоение учениками заданного предметного материала.
- Личностно-ориентированные технологии обучения, способ организации обучения, в процессе которого обеспечивается всемерный учет возможностей и способностей обучаемых и создаются необходимые условия для развития их индивидуальных способностей.
- Информационно-коммуникационные технологии.
- Технология коллективных методов обучения (работа в парах постоянного и сменного состава)

**Формы организации образовательного процесса:** фронтальные, групповые, индивидуальные, индивидуально-групповые, практикумы; урок-консультация, урок-практическая работа, уроки с групповыми формами работы, уроки-конкурсы.

### 3. Место курса в учебном плане.

Данная программа предусматривает на реализацию программы по информатике в 8 классе 34 часа. Рабочая программа может реализовываться с использованием электронного обучения (ЭО) и дистанционных образовательных технологий (ДОТ).

### 4. Личностные, метапредметные и предметные результаты освоения учебного предмета.

**Личностными результатами, формируемыми при изучении предмета информатика, являются:**

- формирование ответственного отношения к учению, готовности и способности обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;
- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.

**Метапредметные результаты изучения предмета «Информатика»:**

- умение самостоятельно определять цели своего обучения, ставить и формулировать для себя новые задачи в учёбе и познавательной деятельности, развивать мотивы и интересы своей познавательной деятельности;
- умение самостоятельно планировать пути достижения целей, в том числе альтернативные, осознанно выбирать наиболее эффективные способы решения учебных и познавательных задач;
- умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности в процес-

- се достижения результата, определять способы действий в рамках предложенных условий и требований, корректировать свои действия в соответствии с изменяющейся ситуацией;
- умение оценивать правильность выполнения учебной задачи, собственные возможности её решения;
  - владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
  - умение определять понятия, создавать обобщения, устанавливать аналогии, классифицировать, самостоятельно выбирать основания и критерии для классификации, устанавливать причинно-следственные связи, строить логическое рассуждение, умозаключение (индуктивное, дедуктивное и по аналогии) и делать выводы;
  - умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебных и познавательных задач;
  - умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками; работать индивидуально и в группе: находить общее решение и разрешать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;
  - формирование и развитие компетентности в области использования информационно-коммуникационных технологий.

#### **Предметные результаты изучения предмета «Информатика»:**

- формирование представления об основных изучаемых понятиях курса;
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;

- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для решения конкретной задачи;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей, с использованием соответствующих программных средств обработки данных;
- формирование представления о том, что значит “программировать” на примере языка Scratch, формирование умения составлять сценарии проектов среды Scratch;
- знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- формирование умения тестировать и оптимизировать алгоритмы исполнителей;
- формирование умения создавать и редактировать документы в текстовом процессоре;
- формирование умения размещать документы в облачном хранилище. организовывать коллективную работу с документами, настраивать права доступа к документам;
- формирование умения формализации и структурирования информации,
- использование готовых прикладных компьютерных программ по выбранной специализации;
- развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы информационной этики и права.

## 5. Содержание курса

### **Программирование в Scratch (24 часа)**

Знакомство со средой программирования Scratch. Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Библиотека персонажей. Исполнитель Scratch. Основные инструменты встроенного графического редактора программной среды SCRATCH. Линейный алгоритм. Создание блок-схемы. Рисование линий исполнителем Scratch. Конечный и бесконечный циклы. Цикл в цикле. Анимация исполнителя Scratch на основе готовых костюмов. Дублирование исполнителей. Алгоритмы с ветвлением. Цикл с условием. Перемещение исполнителей между слоями. Программирование клавиш. Управление событиями. Координатная плоскость. Создание списков. Использование подпрограмм. Отладка программ с ошибками.

### **Работа с текстовым процессором LibreOffice.org Writer (10 часов)**

Загрузка и установка LibreOffice. Интерфейс редактора. Стандартные действия. Форматирование документа: шрифты, стили, размер шрифта. Работа с цветом. Сложное форматирование. Использование списков. Колонтитулы. Изображения в текстовых документах. Графика в текстовых документах. Таблицы в документах. Работа с Google-docs.

## 6. Поурочное планирование

### **Модуль 1. Программирование в Scratch**

#### **Урок 1. Знакомство со средой программирования Scratch**

Свободное программное обеспечение. Авторы программной среды Scratch. Параметры для скачивания и установки программной среды на домашний компьютер.

Основные элементы пользовательского интерфейса программной среды Scratch. Внешний вид рабочего окна. Блочная структура систематизации информации. Функциональные блоки. Блоки команд, состояний, программ, запуска, действий и исполнителей. Установка русского языка для Scratch.

Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Очистка экрана.

Библиотека персонажей. Сцена и разнообразие сцен, исходя из библиотеки данных. Систематизация данных библиотек персонажей и сцен. Иерархия в организации хранения костюмов персонажа и фонов для сцен. Импорт костюма, импорт фона.

#### **Кейс 1. Научи кота бегать и мяукать.**

#### **Урок 2. Исполнитель Scratch, цвет и размер пера. Основные инструменты встроенного графического редактора программной среды SCRATCH.**

Понятие ИСПОЛНИТЕЛЯ.

Команды управления пером: «Опустить перо», «Поднять перо», «Очистить», «Установить цвет пера», «Установить размер пера.

Инструменты растрового графического редактора — кисточка, ластик, заливка, квадрат, круг, линия.

Копирование, поворот, горизонтальное отражение, вертикальное отражение во встроенном редакторе программной среды Scratch. Командный блок внешность (фиолетовый) — команды начальной установки эффекта цвет «Установить эффект цвет в значение 0 и команда начальной установки размера «Установить размер 100%». Команды: «Изменить цвет эффект на \_», «Изменить размер на \_»

Командный блок управления пером (зелёный) — команда «Печатать» для копирования графического изображения исполните-

ля в нужном месте экрана.

Эффекты, которые могут быть применены к графическим изображениям действующего исполнителя. В графическом редакторе можно создавать сложные графические изображения, почти не прикладывая собственных усилий, а лишь правильно применяя встроенные возможности программной среды.

**Кейс 2. Рисуем разноцветные лужи и облака для прогулки кота.**

**Урок № 3. Алгоритм. Линейный алгоритм. Создание блок-схемы. Основные графические примитивы векторного редактора LibreOffice.Draw.**

Создание собственных изображений в других программах (например, LibreOfficeDraw) и импортирование их в программную среду Scratch.

Знакомство с основными графическими примитивами векторного редактора LibreOfficeDraw. Возможность создания геометрических фигур без внутренней заливки, но с текстовым блоком внутри. Стрелки, их направление. Алгоритм, блок-схема как способ записи.

**Кейс 3. Запиши мой алгоритм!**

Ребята разбиваются на пары, в паре формулируют друг другу какую-то повседневную задачу, для решения которой нужен линейный алгоритм, придумывают этот алгоритм и рисуют блок-схему для задачи, заданной им напарником.

**Урок № 4. Линейный алгоритм. Рисование линий исполнителем Scratch.**

Решение поставленной задачи в виде последовательного выполнения команд.

Создание блок-схемы линейного алгоритма средствами редактора векторной графики.

Последовательное выполнение команд.

Изменение параметров пера.

**Кейс 4. Что бывает полосатое?**

Придумайте линейный алгоритм для создания полосатого объекта (заранее придумайте, что это будет – шарф, зебра, забор?) Создайте блок-схему этого алгоритма и реализуйте алгоритм в среде Scratch.

**Урок № 5. Линейный алгоритм. Исполнитель Scratch рисует квадраты и прямоугольники линейно.**

Создание алгоритма для рисования исполнителем квадрата путем последовательного выполнения команд.

Рисование линейного алгоритма, состоящего из двух колонок блоков команд.

Использование векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем.

Выбор нужного значения из предлагаемого списка вариантов.

Отладка программы для получения верного результата. Команда «повернуть в направление».

Пошаговое выполнение программы для её отладки.

Центр костюма исполнителя Scratch.

**Урок №6. Конечный цикл. Scratch рисует несколько линий и фигур. Копирование фрагментов программы.**

Использование операции цикла для решения учебных задач. Применение поворота на прямой угол (90°) при создании геометрических фигур и перемещении исполнителя. Оптимизация линейного алгоритма за счёт использования циклической конструкции в программной среде Scratch. Тело цикла.

Конечный и бесконечный циклы.

Блок-схема бесконечного цикла.

Имя спрайта и костюма.

Изменение костюма исполнителя.

Копирование фрагмента программы.

Команды: «Следующий костюм», «Перейти к костюму».

Рекомендуется объяснить, что циклические конструкции могут применяться не только для рисования фигур. Например, в цикле можно менять внешность исполнителя.

**Урок №7. Циклический алгоритм. Цикл в цикле.** Использование операции цикла в цикле для решения учебных задач.

Использование поворота на прямой угол (90°) по часовой и против часовой стрелки.

Оптимизация алгоритма за счёт использования конструкции «цикл в цикле».

Блок-схема конструкции цикл в цикле.

Например, расписание уроков составляется на неделю, в месяце в среднем четыре недели. Если расписание не меняется в течение учебного года, то получаем цикл из девяти месяцев по четыре недели с одинаковым расписанием.

**Упражнение.** Использование циклов «повторить \_» Нарисуйте квадраты, как показано на рисунке, используя конструкцию цикл в цикле.

### **Кейс 5. Мой необычный дом.**

Предложите ребятам нарисовать дом и что-то рядом с ним, используя линейные и циклические алгоритмы, а также смену костюмов Исполнителя. Может быть, над домом будут звезды разного размера, а около дома будет пунктирная разноцветная дорога, или деревья разного размера.

### **Урок №8. Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.**

Бесконечный цикл.

Анимация исполнителя с помощью смены костюмов.

Эффект «призрак».

Изменение размера исполнителя.

Интерактивное взаимодействие с исполнителем с помощью клавиатуры.

Команды «спросить» и «думать».

### **Урок №9. Сцена как исполнитель. Создаем модель таймера.**

**Параллельное выполнение действий несколькими исполнителями.** Анимация сцены (фона).

Синхронная анимация сцены и исполнителя.

### **Урок №10. Одинаковые программы для нескольких исполнителей.**

Анимация с помощью вращения.

Дублирование исполнителей.

При копировании исполнителей копируются и их скрипты.

Использование одинаковых программ, но разных костюмов у исполнителей.

### **Урок №11. Дублирование исполнителей.**

Синхронное выполнение скриптов.

При параллельном выполнении скриптов результат работы всех исполнителей получается гораздо быстрее, чем в тех задачах, когда работал один исполнитель.

### **Урок №12. Разбиение программы на части для параллельного выполнения исполнителями. Таймер.**

Планирование действий разных исполнителей во времени для решения общей задачи.

Сенсор «таймер»

Сброс «таймера»

Команда «ждать до ...»

### **Кейс 6. Мини-проект «Смена времени суток».**

Создайте скрипт, в котором будет четыре сцены: «утро», «день», «вечер», «ночь». Используйте «таймер» для своевременного изменения сцен, своевременного запуска скриптов разных исполнителей.

### **Урок №13. Два исполнителя со своими программами.**

**Мини-проект «Часы с кукушкой».** Координаты в Scratch.

Команда «идти в x: \_ y: \_» (переместиться в точку с координатами (x,y))

Команда «плыть \_ секунд в точку x: \_ y: \_» (плавно двигаться в точку с координатами (x,y))

Команды «спрятаться», «показаться».

### **Кейс 7. Мини-проект «Часы с кукушкой».**

Создайте сцену в виде часов, и три исполнителя: часовую стрелку, минутную стрелку и кукушку. Стрелки должны вращаться на циферблате. При наступлении каждого часа (т.е. в \_ часов : 00 минут) должна появляться кукушка.

### **Урок №14. Алгоритмы с ветвлением. Условие ЕСЛИ.**

Еще из сказок мы знаем, что на камне было написано: «Нале-во пойдешь – коня потеряешь, направо пойдешь – себя не найдешь, прямо пойдешь – в сказку попадешь» Блок-схема алгоритма с ветвлением.

Полная форма ветвления. Команда «если \_ или»

Неполная форма ветвления. Команда «если \_»

Сенсор «касается» (касания края, или другого спрайта, или указателя мыши)

### **Урок №15. Цикл с условием. Мини-проект «Шарики в лабиринте»**

Сенсор «касается цвета»

Программируем отскок шариков от стен .

### **Кейс 8. Мини-проект «Шарики в лабиринте»**

Создайте сцену в виде бильярдного стола, на котором три шарика будут двигаться с разными скоростями и отталкиваться от бортиков.

### **Урок №16. Цикл с условием. Исполнитель определяет цвет.**

Сенсор «касается цвета»

Программируем поведение исполнителя в зависимости от цвета фона.

### **Урок № 17. Оператор случайных чисел.**

Команда «Выдать случайное число от \_ до \_». Случайные перемещения исполнителя в координатной плоскости. Случайное количество шагов.

Случайные координаты.

Поворот на случайный угол.

### **Урок №18. Перемещение исполнителей между слоями.** Команда «Перейти в верхний слой» Команда «Перейти назад на \_ слоев» Эффект «Призрак»

### **Урок №19. Действия исполнителей в разных слоях.**

### **Кейс 9. Мини-проект «Дорога».**

Создайте сцену, на которой изображена дорога с двумя полосами, по обочинам дороги расположите кусты и деревья. По дороге должны двигаться автомобили. Правильно расположите автомобили, деревья и кусты по слоям.

### **Урок №20. Взаимодействие исполнителей.**

Исполнители касаются друг друга.

Команда «Касается \_»

Поведение исполнителей при столкновении.

### **Урок №21. Последовательное выполнение команд исполнителями.**

Команда «Передать»

Команда «Когда я получу»

Связи между программами разных исполнителей.

### **Урок №22. Программирование клавиш. Мини-проект «Лабиринт»**

Взаимодействие пользователя с программой.

Клавиши управления перемещением исполнителя

Команда «Изменить значение x на \_»

Команда «Изменить значение у на \_»

### **Кейс 10. Игра «Лабиринт»**

Создайте сцену в виде лабиринта. С помощью клавиш со стрелками управляйте движением маленькой мышки. Мышка должна добраться до выхода, не задевая стенок лабиринта. В случае касания стенок мышка возвращается в исходную точку.

### **Урок №23. Координатная плоскость. Геометрические фигуры. Переменные.**

Рисование геометрических фигур на координатной плоскости.

Последовательное выполнение команд одним исполнителем.

Параллельное выполнение команд несколькими исполнителями. Блок «Переменные». Рисование геометрических фигур на координатной плоскости с использованием переменных.

Построение перпендикуляров к координатным осям.

### **Кейс 11. Мини-проект «Викторина».**

Создайте проект «Викторина» в котором будет задано 10 во-просов.

Правильный ответ надо выбрать из списка. В случае правильного ответа очки увеличиваются на единицу. В конце выставляется оценка: «отлично» за 9 или 10, «хорошо» за 7 или 8, «удовлетворительно» за 5 или 6 правильных ответов.

### **Урок №24. Использование подпрограмм. Сообщество Scratch.**

Как сделать программу структурированной и более понятной.

Команда «Передать \_ и ждать»

Команда «Играть звук \_»

Команда «Ноту \_ играть \_ тактов»

Регистрация на сайте.

Личный кабинет.

Публикация проектов.

### **Итоговый проект.**

Учащиеся самостоятельно или в парах выполняют индивидуальный проект, согласованный с учителем. Как итог, можно провести конкурс проектов.

### **Обучающийся научится:**

- составлять сценарии проектов среды Scratch;
- составлять алгоритмы, определять последовательность выполнения команд;
- создавать и редактировать рисунки в графическом редакторе;
- использовать обширную библиотеку готовых сцен и исполнителей;
- изменять размер, костюм, прозрачность исполнителя;
- создавать линейные алгоритмы для исполнителя;
- создавать циклические алгоритмы;
- создавать ветвящиеся алгоритмы;
- управлять одновременной работой нескольких исполнителей;
- передавать сообщения между исполнителями;
- внедрять звуковые эффекты в алгоритмы исполнителей;
- создавать алгоритмы, которые будут выполняться одновременно (параллельно) несколькими исполнителями;
- тестировать и оптимизировать алгоритмы исполнителей.

## **Модуль 2. Работа с текстовым процессором LibreOffice.org Writer.**

### **Урок №25. Знакомство с офисным пакетом**

**LibreOffice.** LibreOffice — офисный пакет, совместимый с 32/64-битными системами.

Поддерживает большинство популярных операционных систем, включая GNU/Linux, Microsoft Windows и Mac OS X.

LibreOffice бесплатен и имеет открытый исходный код, следовательно, его можно бесплатно скачивать и использовать.

Загрузка и установка LibreOffice. Интерфейс редактора.

Меню, панели инструментов, строка состояния.

Режимы работы с документом.

Создание нового документа, ввод текста.

Сохранение, открытие и закрытие документа.

Редактирование текста: копирование, вставка, перемещение, удаление фрагментов текста.

Форматирование документа: шрифты, стили, размер шрифта.

Работа с цветом.

Отмена действий.

### **Урок №26. Сложное форматирование.**

Межстрочный интервал, выравнивание, отступ, поля.

Вставка специальных символов.

Отображение непечатаемых символов. Буквица. Водяные знаки.

### **Урок №27. Использование списков. Колонтитулы.**

Верхний и нижний колонтитулы.

Форматирование колонтитулов.

Прозрачность.

Нумерация страниц.

Нумерованный список.

Маркированный список.

Многоуровневый список.

Параметры списка.

### **Урок №28. Изображения в текстовых документах.**

Вставка рисунка в документ.

Форматирование рисунка.

Размер и положение рисунка.

Обтекание текстом.

Вставка фигур: линии, стрелки, многоугольники и т.п.

### **Кейс 1. Мини-Европа.**

Начнем работу над проектом Мини-Европа. Предыстория, которую рассказывает учитель: в некоторых европейских городах, например, в Брюсселе, есть парки, в которых расположены интересные европейские объекты в миниатюре (показать видео или презентацию). Мы создадим мини-книжку, в которой будут представлены те европейские достопримечательности, которые вам бы хотелось увидеть. Выберите 4-6 стран и найдите в интернете наиболее интересные достопримечательности (например, Франция - Эйфелева башня и т.п.), а также неболь-

шой рассказ про них (можно воспользоваться Википедией). Расположите информацию про каждую страну на отдельной странице, отформатируйте текст, добавьте нумерацию и колонтитулы (в колонтитулы можно вынести название страны или города).

### **Урок №29. Графика в текстовых документах.**

Текстовые эффекты.

Надписи.

Схемы.

Редактор формул.

### **Урок №30. Создание таблиц.**

Вставка таблицы в документ.

Добавление строк и столбцов в таблицу.

Ширина столбцов и высота строк таблицы.

Внешние и внутренние границы таблицы.

### **Урок №31. Работа с таблицами.**

Название таблицы.

Объединение ячеек. Разбиение ячеек.

Направление текста в ячейке.

Изменение фона ячейки, таблицы.

Изменение границ таблицы.

### **Урок №32. Знакомство с Google-docs.**

Бесплатный сервис. Облачное хранилище.

Обмен файлами.

Кроссплатформенность.

Коллективная работа.

### **Урок №33. Начало работы с Google-docs.**

Учетная запись Gmail.

Google – диск.

## **Кейс 2 (большой, на всю тему изучения Google-документов).**

### **Создание папки нашего класса.**

Каждый ребенок создает свой Google-документ, в котором будет собирать небольшой рассказ о себе, своих увлечениях. Учитель создает общую папку, а также Google-таблицу, в которой содержатся ссылки на странички ребят.

### **Урок №34. Работа с текстом в Google-docs.**

Ввод и редактирование текста.

Панель инструментов.

- **Обучающийся научится:** создавать и редактировать документы в текстовом процессоре;
- работать с панелями инструментов текстового процессора;
- работать с блоками текста: выделять, копировать, удалять;
- использовать необходимые шрифты;
- форматировать документ;
- вставлять в документ таблицы, схемы, рисунки;
- создавать и редактировать документы в Google – docs;
- работать с инструментами Google – docs;
- размещать документы в облачном хранилище;
- организовывать коллективную работу с документами;
- настраивать права доступа к документам.

