

Муниципальное казенное общеобразовательное учреждение
«Средняя общеобразовательная школа №12»
Центр цифрового и гуманитарного профилей «Точка роста»

Принято решением
Педагогического совета
№1 от «28»09.2024

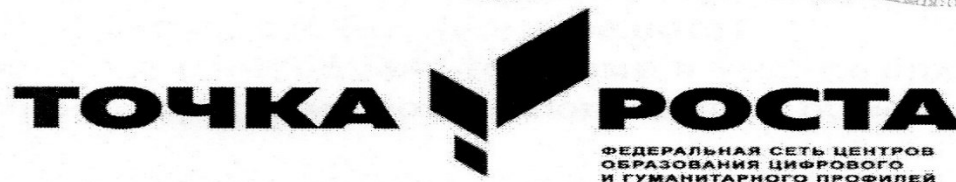
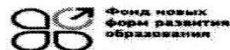


Утверждаю:

Директор МКОУ СОШ №12

Дутова И.И.

«28» 09.2024г.



Дополнительная образовательная (общеразвивающая) программа

технической направленности

«Информатика»

Возрастная категория: 11-15 лет

Срок реализации: 1год

Составитель:

Чаплыгин Иван Андреевич
педагог дополнительного образования

с.Красногвардейское
2024 год.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа «Информатика, компьютерная графика и графические редакторы» имеет техническую направленность, поскольку направлена на развитие личности обучающегося, его творческой самореализации посредством погружения в компьютерный мир и компьютерных технологий.

Актуальность и новизна программы

Программа данного курса посвящена обучению школьников умению работать с растровой и векторной графикой, умению создавать трехмерные модели в различных программах, а также умению создавать презентации в различных компьютерных программах. Занятия курса направлены на развитие мышления, логики, творческого потенциала учеников. Программа ориентирована на использование полученных знаний для разработки реальных проектов. Содержит большое количество творческих заданий (именуемых Кейсами).

Программа способствует развитию познавательной активности обучающихся и творческого мышления, а также профориентации. Визуальная составляющая современных информационных технологий базируется на красочных графических элементах, разнообразных видах анимации, интерактивных элементах управления. Любой продукт информационных технологий не сможет привлечь внимание пользователя без графической и анимационной составляющих. Создание продукта, содержащего коллекции изображений; текстов и данных, сопровождающихся звуком, видео, анимацией и другими визуальными эффектами, составляет основу компьютерной графики и анимации. Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися в таких областях знаний, как физика, химия, биология и других, они также являются фундаментом для дальнейшего совершенствования мастерства в области трёхмерного моделирования, анимации и видеомонтажа.

Отличительные особенности программы

Содержание данной программы ориентировано на учащихся образовательных организациях основного общего и среднего образования. Программа является личностно-ориентированной, что позволяет каждому ученику выбрать наиболее интересный объект работы, который ему больше подходит.

Адресат программы

Программа ориентирована на обучающихся возрастной категории 11-15 лет, имеющих базовый уровень владения ИКТ.

Объем и срок освоения программы

Программа рассчитана на 5 лет. Количество учебных часов по программе: 360

Форма обучения

Очная. Занятия будут проводиться на базе центра образования цифрового и гуманитарного профилей «Точка роста».

Особенности организации образовательного процесса

Специального отбора детей в детское объединение для обучения по дополнительной общеобразовательной общеразвивающей программе «Информатика, компьютерная графика и графические редакторы» не предусмотрено. Подростки приходят в группу по желанию и интересам. Программа может реализовываться с использованием электронного обучения (ЭО) и дистанционных образовательных технологий (ДОТ).

Состав группы – от 3-7 человек

Режим занятий, периодичность и продолжительность занятий

Программа «Информатика, компьютерная графика и графические редакторы» рассчитана на 5 лет обучения. Длительность и количество занятий - 2 раза в неделю. Общий объем – 360 часов, по 72 часа в год.

Возрастные особенности учащихся

Программа рассчитана для обучающихся 11-15 лет – это *подростки*.

Подросток отличает стремление к самостоятельности, независимости, к самопознанию, формируются познавательные интересы. Задача педагога доверять подростку решение посильных для него вопросов, уважать его мнение. Общение предпочтительнее строить не в форме прямых распоряжений и назиданий, а в форме проблемных вопросов. У подростка появляется умение ставить перед собой и решать задачи, самостоятельно мыслить и трудиться.

Подросток проявляет инициативу, желание реализовать и утвердить себя. В этот период происходит окончательное формирование интеллекта, совершенствуется способность к абстрактному мышлению. Для старшего подростка становится потребностью быть взрослым. Проявляется стремление к самоутверждению себя в роли взрослого. Задача педагога побуждать учащегося к открытию себя как личности и индивидуальности в контексте художественного творчества, к самопознанию, самоопределению и самореализации. Совместная деятельность для подростков этого возраста привлекательна как пространство для

общения.

Учет возрастных особенностей детей, занимающихся по образовательной программе «Информатика, компьютерная графика и графические редакторы», является одним из главных педагогических принципов.

Цель программы:создать условия, обеспечивающие личностное, познавательное и творческое развитие обучающегося в процессе изучения основ графики и анимации с использованием компьютерных технологий.

Задачи:

Образовательные

- освоение базовых понятий и методов компьютерной графики;
- изучение популярных графических программ;
- обеспечение глубокого понимания принципов построения и хранения изображений.
- формирование базы практических знаний, необходимых для самостоятельной разработки объектов растровой и векторной графики, а также коротких анимаций, интерактивных элементов для web-публикаций и различных приложений.
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей;
- овладение важнейшими общеучебными умениями и универсальными учебными действиями (формулировать цели деятельности, планировать ее, находить и обрабатывать необходимую информацию из различных источников, включая Интернет и др.).

Развивающие:

- создание благоприятных условий для развития творческих способностей детей, математического и логического мышления.
- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе;
- развитие умений составить и записать алгоритм.

Воспитательные:

- формирование информационной и алгоритмической культуры.

Образовательный процесс строится с учетом следующих принципов:

1.Индивидуального подхода учащимся.

Этот принцип предусматривает взаимодействие между педагогом и воспитанником. Подбор индивидуальных практических заданий необходимо производить с учетом личностных особенностей каждого учащегося, его заинтересованности и достигнутого уровня подготовки.

2.Культуросообразности и природосообразности.

В программе учитываются возрастные и индивидуальные особенности подростков.

3. Системности.

Полученные знания, умения и навыки учащиеся системно применяют на практике, создавая творческую работу. Это позволяет использовать знания и умения в единстве, целостности, реализуя собственный замысел, что способствует самовыражению ребенка, развитию его творческого потенциала.

4. Комплексности и последовательности.

Реализация этого принципа предполагает постепенное введение учащихся в мир компьютерных технологий и программирования.

5. Цикличности.

Обучающиеся осваивают последовательно одни и те же разделы, существует возможность предлагать вновь пришедшим детям задания сначала более простые, соответствующие этому году обучения, а затем более сложные.

6. Наглядности.

Использование наглядности повышает внимание учащихся, углубляет их интерес к изучаемому материалу, способствует развитию внимания, воображения, наблюдательности, мышления.

Основные формы и методы

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

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.

Технологии реализации межпредметных связей в образовательном 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 интеллектуально-го 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.

Методы обучения

В процессе реализации программы используются различные методы обучения.

Методы организации и осуществления учебно-познавательной деятельности:

- словесные (беседа; речевая инструкция; объяснение нового материала и способов выполнения задания; объяснение последовательности действий и содержания; обсуждение; педагогическая оценка процесса деятельности и ее результата);
- наглядные (показ видеоматериалов и иллюстраций, показ педагогом приёмов исполнения, показ по образцу, демонстрация, рассматривание фотографий, слайдов);
- практически-действенные (упражнения на развитие моторики пальцев рук (пальчиковая гимнастика, физкультминутки);
- проблемно-поисковые (создание проблемной ситуации, коллективное обсуждение, выводы);
- методы самостоятельной работы и работы под руководством педагога (создание творческих проектов);
- инструкторский метод (парное взаимодействие, более опытные учащиеся обучают менее подготовленных);
- информационные (объяснение, инструктаж, консультирование, использование средств массовой информации, анализ различных носителей информации, в том числе Интернет-сети, демонстрация, экспертиза, обзор, отчет, иллюстрация).
- побудительно-оценочные (педагогическое требование и поощрение порицание и создание ситуации успеха; самостоятельная работа).

Методы контроля и самоконтроля за эффективностью учебно- познавательной деятельности:

- устный контроль и самоконтроль (беседа, рассказ ученика, объяснение, устный опрос);
- практический контроль и самоконтроль (анализ умения работать с дидактические тесты (набор стандартизованных заданий по определенному материалу);
- наблюдения (изучение обучающихся в процессе обучения).

Выбор метода обучения зависит от содержания занятий, уровня подготовленности и опыта учащихся.

Информационно-рецептивный метод применяется на теоретических занятиях.

Репродуктивный метод обучения используется на практических занятиях по отработке приёмов и навыков определённого вида работ.

Исследовательский метод применяется в работе над тематическими творческими проектами.

Для создания комфортного психологического климата на занятиях применяются следующие педагогические приёмы: создание ситуации успеха, моральная поддержка, одобрение, похвала, поощрение, доверие, доброжелательно-требовательная манера.

В ходе реализации программы используются следующие типы занятий:

- комбинированное (совмещение теоретической и практической частей занятия; изложение нового материала, закрепление новых знаний, формирование умений переноса и применения знаний в новой ситуации, на практике; отработка навыков и умений);
- теоретическое (сообщение и усвоение новых знаний при объяснении новой темы, изложение нового материала, основных понятий, определение терминов, совершенствование и закрепление знаний);
- диагностическое (проводится для определения возможностей и способностей ребенка, уровня полученных знаний, умений, навыков с использованием тестирования, собеседования, выполнения заданий);
- контрольное (проводится в целях контроля и проверки знаний, умений и навыков учащегося через самостоятельную работу, индивидуальное собеседование).

– практическое (является основным типом занятий, используемых в программе, как правило, содержит повторение, обобщение и усвоение полученных знаний, формирование умений и навыков, их осмысление и закрепление на практике, инструктаж при работе на компьютере, ноутбуке, принтером).

вводное занятие (проводится в начале учебного года с целью знакомства с образовательной программой на год, составление индивидуальной траектории обучения; а также при введении в новую тему программы),

итоговое занятие (проводится после изучения большой темы или раздела, по окончании полугодия, учебного года)

Планируемые результаты:

Важнейшими умениями/знаниями являются следующие:

- умение пользоваться персональным компьютером и его периферийным оборудованием;
- умение следовать требованиям техники безопасности, гигиены, эргономики и ресурсосбережения при работе со средствами информационных и коммуникационных технологий:
- умение осуществлять взаимодействие посредством электронной почты, чата, форума;
- умение искать информацию с применением правил поиска(построения запросов), в компьютерных сетях, некомпьютерных источниках информации (справочниках и словарях, каталогах, библиотеках) при выполнении заданий и проектов по различным учебным дисциплинам;
- умение создавать и редактировать растровые и векторные изображения; понимать преимущества и недостатки растровых и векторных изображений;
- понимание принципов построения трехмерного изображения, принципов полигонального моделирования;
- умение работать в программе трёхмерного моделирования Autodesk Fusion 360;
- умение работать в программе трёхмерного моделирования Blender
- умение работать в программе трёхмерного моделирования 3ds Max

- умение создавать и редактировать презентации в Impress;
- умение создавать и редактировать презентации в Google Slides;
- умение размещать документы в облачном хранилище, организовывать коллективную работу с документами, настраивать права доступа к документам;

умение пользоваться сервисами Интернета, а также понимание основ безопасности при работе в Интернете • понимание того, что такое внешние API сайтов, и умение с ними работать с помощью соответствующих библиотек языка Python

- понимание того, как происходит процесс создания сайтов, что такое backend и frontend
- умение писать на языке разметки HTML5, пользоваться каскадными таблицами стилей
- умение создавать веб-страницы
- знание языка программирования JavaScript и его применение для работы с backend'ом
- знакомство с процессом разработки сайтов, умение работать в команде, разрабатывать и реализовывать идеи в рамках технического задания
- умение обрабатывать видеоизображения, добавляя титры, несложные эффекты и переходы
- умение работать с каналами на сервисе YouTube
- умение выбирать способ представления своего проекта с использованием соответствующих программных средств.

- **Механизм оценивания образовательных результатов.**

1. Уровень теоретических знаний.

- Низкий уровень. Обучающийся знает фрагментарно изученный материал. Изложение материала сбивчивое, требующее корректировки наводящими вопросами.
- Средний уровень. Обучающийся знает изученный материал, но для полного раскрытия темы требуются дополнительные вопросы.

- Высокий уровень. Обучающийся знает изученный материал. Может дать логически выдержанный ответ, демонстрирующий полное владение материалом.

2. Уровень практических навыков и умений.

Владение компьютерными технологиями

- Низкий уровень. Требуется помощь педагога.

- Средний уровень. Требуется периодическое напоминание о том, какие технологии и как применять.

- Высокий уровень. Самостоятельный выбор технологии в подготовке творческой работы.

- Низкий уровень. Не может создать проект или творческую без помощи педагога.

- Средний уровень. Может создать проект при подсказке педагога.

- Высокий уровень. Способен самостоятельно создать творческий проект, проявля творческие способности.

Степень самостоятельности при подходе к материалу.

- Низкий уровень. Требуются постоянные пояснения педагога.

- Средний уровень. Нуждается в пояснении последовательности работы, но способен после объяснения к самостоятельным действиям.

- Высокий уровень. Самостоятельно создает предложенные виды работ на основе повтора, импровизации.

Формы подведения итогов реализации программы

Отслеживание результатов образовательного процесса осуществляется посредством аттестации. Дети, обучающиеся по данной программе, проходят аттестацию 2 раза в год:

– аттестация в начале учебного года (сентябрь-октябрь);

– аттестация в конце учебного года (итоговая, май).

При подведении итогов освоения программы используются:

– опрос;

– наблюдение;

– анализ, самоанализ,

– собеседование;

– тестирование и анкетирование;

– выполнение творческих заданий;

– презентации;

- викторина;
- участие детей в выставках, конкурсах и фестивалях различного уровня.

Результаты аттестации отражаются в индивидуальной карте ребенка для отслеживания динамики его развития, что помогает проводить необходимую коррекцию в ходе реализации программы и конструирования учебных занятий.

Учебный планпервого года обучения:

Данная программа предусматривает на реализацию программы по информатике в группе обучающихся 11-12 лет- 72 часа. Рабочая программа рассчитана на 34 учебных недели, 2 часа в неделю, общее количество часов — 68

Содержание программы

1.Программирование в Scratch (34 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.

2.Работа с текстовым процессором LibreOffice.org Writer (34часа)

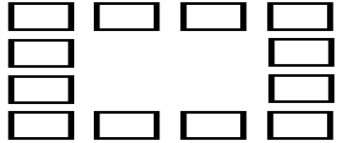
Загрузка и установка LibreOffice. Интерфейс редактора. Стандартные действия. Форматирование документа: шрифты, стили, размер шрифта.Работа с цветом. Сложное форматирование. Использование списков. Колонтитулы. Изображения в текстовых документах. Графика в текстовых документах. Таблицы в документах. Работа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

КАЛЕНДАРНО -ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ программы «Информатика» 1 –й год обучения(5 кл)2 часа

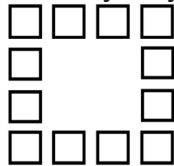
№ п /п	Модули, кейсы,Тема	Деятельность уч-ся	Дата проведения
Модуль 1. Программирование в Scratch(34ч)			
1	Знакомство со средой программирования Scratch Кейс 1. Научи кота бегать и мяукать.	<p>Свободное программное обеспечение. Авторы программной среды Scratch. Параметры для скачивания и установки программной 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.</p> <p>Создание и сохранение документа. Понятия спрайта, сцены, скрипта. Очистка экрана.</p> <p>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.</p>	

		действующего исполнителя. В графическом редакторе можно создавать сложные графические изображения, почти не прикладывая собственных усилий, а лишь правильно применяя встроенные возможности программной среды.	
4	Урок № 4. Алгоритм. Линейный алгоритм. Создание блок-схемы. Кейс 4. Запиши мой алгоритм!	Основные графические примитивы векторного редактора LibreOffice.Draw. Создание собственных изображений в других программах (на пример, LibreOfficeDraw) и импорт 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. <i>Кейс 4. Запиши мой алгоритм!</i> 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.	22.09
5	Урок № 5. Линейный алгоритм. Рисование линий исполнителем Scratch Кейс 5. Что бывает полосатое?	Решение поставленной задачи в виде последовательного выполнения команд. Создание блок-схемы линейного алгоритма средствами редактора векторной графики. Последовательное выполнение команд. Изменение параметров пера. <i>Кейс 5. Что бывает полосатое?</i> 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	29.09
6	Урок № 6. 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.	06.10
7	Урок №7. Конечный цикл. Scratch рисует квадраты, линии. Кейс 6. Создай картинку из квадратов.	Сохранение готовых программ для дальнейшего использования. Рисование блок-схемы циклического алгоритма. Использование векторного редактора офисного пакета LibreOffice в качестве инструмента для создания блок-схем; Использование команд поворота на прямой угол (90°) по часовой и против часовой стрелки; Использование циклического алгоритма для рисования исполнителем квадрата;	13.10

		<p>Оптимизация линейного алгоритма за счёт использования циклической конструкции в программной среде Scratch; Команда открыть... из пункта меню File; Команда сохранить как... из пункта меню File:Циклический алгоритм; Блок-схема циклического алгоритма;</p> <p>Команды: «повернуться на 90° по часовой стрелке», «повернуться на 90° против часовой стрелки», «повторить _».</p> <p><i>Упражнение. Сравнение двух алгоритмов, рисующих квадрат. Идти 100 шагов</i></p> <p>Повторить 4</p> <p>Повернуться на 90° Идти 100 шагов Идти 100 шагов Повернуться на 90°</p> <p>Повернуться на 90°</p> <p>Идти 100 шагов Повернуться на 90° Идти 100 шагов Повернуться на 90°</p> <p>Рекомендуется объяснить, что команды, обеспечивающие повторяющееся выполнение одной или нескольких команд называются циклическими, показать, что использование команды цикла всегда сокращает количество команд в программе, а значит делает её более красивой. В более короткой программе легче обнаружить ошибку или неточность. Рекомендуется объяснить, что циклические конструкции могут применяться не только для рисования фигур. Например, в цикле можно менять внешность исполнителя.</p> <p><i>Кейс 6. Создай картинку из квадратов.</i></p> <p>Предложите ребятам нарисовать свою картинку из квадратов, может быть, это будет лестница, может быть, ковер, может быть, что-то еще.</p>	
8	Урок №8. Конечный цикл. Scratch рисует несколько линий и фигур. Копирование фрагментов программы.	<p>Использование операции цикла для решения учебных задач.</p> <p>Применение поворота на прямой угол (90°) при создании геометрических фигур и перемещении исполнителя.</p> <p>Оптимизация линейного алгоритма за счёт использования циклической конструкции в программной среде Scratch. Тело цикла.</p> <p>Конечный и бесконечный циклы. Блок-схема бесконечного цикла. Имя спрайта и костюма.</p> <p>Изменение костюма исполнителя. Копирование фрагмента программы.</p> <p>Команды: «Следующий костюм», «Перейти к костюму».</p>	20
9	Урок №9. Циклический алгоритм. Цикл в цикле. Использование операции цикла в цикле для решения учебных задач.	<p>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</p>	27.10

		<p>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</p>  <p>конструкцию цикл в цикле</p>	
10	<p>Цикл в цикле. Повторение пунктирной линии с поворотом. Блок-схема цикла.</p> <p>Кейс 7. Мой необычный дом.</p>	<p>Использование конструкции «цикл в цикле».</p> <p>Создание и реализация алгоритма рисования квадрата несплошными линиями;</p> <p>Оптимизация алгоритма за счёт использования конструкции «цикл в цикле» в программной среде Scratch;</p> <p>Использование операции копирования внешности исполнителя путём копирования костюма. Несплошные линии.</p> <p>Алгоритм рисования несплошных линий с использованием циклических конструкций.</p> <p>Рисование квадрата несплошными линиями, используя конструкцию «цикл в цикле».</p> <p>При перемещении исполнителя можно рисовать линии отличные от сплошной. Эти возможности определяются использованием команд из блока ПЕРО.</p> <p><i>Кейс 7. Мой необычный дом.</i></p> <p>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.</p>	10.11
11	Урок №11. Бесконечный цикл. Анимация исполнителя Scratch на основе готовых костюмов.	<p>Бесконечный цикл. Анимация исполнителя с помощью смены костюмов.</p> <p>Эффект «призрак». Изменение размера исполнителя.</p> <p>Интерактивное взаимодействие с исполнителем с помощью клавиатуры.</p> <p>Команды «спросить» и «думать».</p>	17.11
12	Урок №12. Сцена как исполнитель. Создаем модель таймера.	<p>Анимация сцены (фона).</p> <p>Синхронная анимация сцены и исполнителя.</p>	24.11
13	Урок №13. Одинаковые программы для нескольких исполнителей.	<p>Анимация с помощью вращения. Дублирование исполнителей.</p> <p>При копировании исполнителей копируются и их скрипты.</p> <p>Использование одинаковых программ, но разных костюмов у исполнителей.</p>	01.12
14	Урок №14. Параллельное выполнение действий несколькими исполнителями	<p>Дублирование исполнителей. Синхронное выполнение скриптов. При параллельном выполнении скриптов результат работы всех исполнителей</p>	08.12

		получается гораздо быстрее, чем в тех задачах, когда работал один исполнитель.	
15	Урок №15. Разбиение программы на части для параллельного выполнения исполнителями. Таймер. Кейс 8. Мини-проект «Смена времени суток».	Планирование действий разных исполнителей во времени для решения общей задачи. Сенсор «таймер» Сброс «таймера» Команда «ждать до ...» <i>Кейс 8. Мини-проект «Смена времени суток».</i> Создайте скрипт, в котором будет четыре сцены: «утро», «день», «вечер», «ночь». Используйте «таймер» для своевременного изменения сцен, своевременного запуска скриптов разных исполнителей.	15.12
16	Урок №16. Два исполнителя со своими программами. Мини-проект «Часы с кукушкой». Кейс 9. 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». <i>Кейс 9. Мини-проект «Часы с кукушкой».</i> 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.	22.12
17	Урок №17. Алгоритмы с ветвлением. Условие ЕСЛИ.	Еще из сказок мы знаем, что на камне было написано: «Налево 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)	29.12
18	Урок №18. Цикл с условием. Мини-проект «Шарики в лабиринте» Кейс 10. Мини-проект «Шарики в лабиринте»	Сенсор «касается цвета» Программируем отскок шариков от стен <i>Кейс 10. Мини-проект «Шарики в лабиринте»</i> Создайте сцену в виде бильярдного стола, на котором три шарика будут двигаться с разными скоростями и отталкиваться от бортиков.	12/01
19	Урок №19. Цикл с условием.	Исполнитель определяет цвет. Сенсор «касается цвета» Программируем поведение исполнителя в зависимости от цвета фона.	19/01
20	Урок № 20. Оператор случайных чисел.	Команда «Выдать случайное число от _ до _». Случайные перемещения исполнителя в координатной плоскости. Случайное количество шагов. Случайные координаты. Поворот на случайный угол.	26/01
21	Урок №21. Перемещение исполнителей между слоями.	Команда «Перейти в верхний слой» Команда «Перейти назад на _ слоев» Эффект «Призрак»	2/02

22	Урок №22. Действия исполнителей в разных слоях. Кейс 10. Мини-проект «Дорога».	<i>Кейс 10. Мини-проект «Дорога».</i> 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.	9/02
23	Урок №23 Взаимодействие исполнителей.	Исполнители касаются друг друга. Команда «Касается _» Поведение исполнителей при столкновении.	16/02
24	Урок №24. Последовательное выполнение команд исполнителями.	Команда «Передать» Команда «Когда я получу» Связи между программами разных исполнителей.	1/03
25	Урок №25. Программирование клавиш. Мини-проект «Лабиринт» Кейс 10. Игра «Лабиринт»	Взаимодействие пользователя с программой. Клавиши управления перемещением исполнителя Команда «Изменить значение x на _» Команда «Изменить значение y на _» <i>Кейс 10. Игра «Лабиринт»</i> 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.	15/03
26	Урок №26. Управление событиями.	Передача сообщений между исполнителями и фоном. Запуск программ после получения сообщения.	22/03
27	Урок №27. Координатная плоскость. Геометрические фигуры. Кейс 11. Исследование времени выполнения программ.	Рисование геометрических фигур на координатной плоскости. Последовательное выполнение команд одним исполнителем. Параллельное выполнение команд несколькими исполнителями. <i>Кейс 11. Исследование времени выполнения программ.</i> Создайте одного исполнителя, рисующего квадраты, как показано на рисунке. Затем создайте четыре исполнителя, которые будут параллельно рисовать по три квадрата. Сравните время работы в этих двух случаях. 	
28	Урок №28. Координатная плоскость. Переменные.	Блок «Переменные» Рисование геометрических фигур на координатной плоскости с использованием переменных. Построение перпендикуляров к координатным осям.	
29	Урок №29. Создание списков.	Название списка. Элементы списка.	

	Кейс 11. Мини-проект «Викторина».	<p>Длина списка. Команда «Создать список».</p> <p>Выбор элемента списка.</p> <p><i>Кейс 11. Мини-проект «Викторина».</i></p> <p>Создайте проект «Викторина» в котором будет задано 10 вопросов. 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.</p>	
30	Урок №30. Использование подпрограмм.	<p>Как сделать программу структурированной и более понятной.</p> <p>Команда «Передать _ и ждать»</p> <p>Команда «Играть звук _»</p> <p>Команда «Ноту _ играть _ тактов»</p>	
31	Урок №31. Сообщество Scratch.	<p>Scratch 2.0.</p> <p>Регистрация на сайте.</p> <p>Личный кабинет.</p> <p>Публикация проектов.</p>	
32	Урок №32. Отладка программ с ошибками.	<p>http://scratch.mit.edu/projects/10437040</p> <p>http://scratch.mit.edu/projects/10437249</p> <p>http://scratch.mit.edu/projects/1043736622</p> <p>http://scratch.mit.edu/projects/10437439</p> <p>http://scratch.mit.edu/projects/10437476</p>	
33-34	Уроки №33 - 34. Кейс 12. Итоговый проект.	<p>Учащиеся самостоятельно или в парах выполняют индивидуальный проект, согласованный с учителем. Как итог, можно провести конкурс проектов.</p> <p>Обучающийся научится:</p> <ul style="list-style-type: none"> 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.	
35	Модуль 2. Работа с текстовым процессором LibreOffice.org Writer. Урок №35. Знакомство с офисным пакетом LibreOffice.	<p>LibreOffice — офисный пакет, совместимый с 32/64-битными системами. Поддерживает большинство популярных операционных систем, включая GNU/Linux, Microsoft Windows и Mac OS X.</p> <p>LibreOffice бесплатен и имеет открытый исходный код, следовательно, его можно бесплатно скачивать и использовать.</p> <p>Загрузка и установка LibreOffice.</p> <p>Интерфейс редактора.</p> <p>Меню, панели инструментов, строка состояния.</p> <p>Режимы работы с документом.</p> <p>Создание нового документа, ввод текста.</p> <p>Сохранение, открытие и закрытие документа.</p> <p>Редактирование текста: копирование, вставка, перемещение, удаление фрагментов текста.</p> <p>Форматирование документа: шрифты, стили, размер шрифта. Работа с цветом.</p> <p>Отмена действий.</p>	
36	Урок №36. Сложное форматирование. Кейс 1. Создание таинственного документа.	<p>Межстрочный интервал, выравнивание, отступ, поля.</p> <p>Вставка специальных символов.</p> <p>Отображение непечатаемых символов. Буквица. Водяные знаки.</p> <p><i>Кейс 1. Создание таинственного документа.</i></p> <p>Представьте себе, что вы создаете таинственный документ, в котором описано место нахождения сокровища. Придумайте короткий текст, запишите его, отформатируйте его так, чтобы он красиво выглядел на листе. Добавьте водяные знаки, буквицу и какие-нибудь специальные символы.</p>	
37	Урок №37. Использование списков. Кейс 2. О спорт, ты мир!	<p>Нумерованный список. Маркированный список. Многоуровневый список.</p> <p>Параметры списка.</p> <p><i>Кейс 2. О спорт, ты мир!</i></p> <p>Создай многоуровневый список, в котором перечислите несколько наиболее интересных вам видов спорта, у каждого вида спорта (на следующем уровне) перечислите нескольких наиболее известных атлетов, а у каждого атлета (на следующем уровне) – несколько наиболее известных его достижений. Отформатируйте ваш список так, чтобы он выглядел красиво. Для поиска информации воспользуйтесь средой интернет.</p>	
38	Урок №38. Колонтитулы. Кейс 3. Мини-Европа.	<p>Верхний и нижний колонтитулы. Форматирование колонтитулов.</p> <p>Прозрачность. Нумерация страниц.</p> <p><i>Кейс 3. Мини-Европа.</i></p> <p>Начнем работу над проектом Мини-Европа. Предыстория, которую рассказывает</p>	

		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 небольшой 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).	
39	Урок №39. Изображения в текстовых документах. Кейс 3. Продолжение работы над проектом Мини-Европа.	Вставка рисунка в документ. Форматирование рисунка. Размер и положение рисунка. Обтекание текстом. Вставка фигур: линии, стрелки, многоугольники и т.п. <i>Кейс 3. Продолжение работы над проектом Мини-Европа.</i> Найдите в интернете фотографии выбранных вами достопримечательностей и разместите их в тексте.	
40	Урок №40. Графика в текстовых документах. Кейс 3. Завершение работы над проектом Мини-Европа.	Текстовые эффекты. Надписи. Схемы. Редактор формул. <i>Кейс 3. Завершение работы над проектом Мини-Европа.</i> Дополните ваш проект текстовыми эффектами. Покажите вашу работу классу.	
41	Урок №41. Практическая работа. Кейс 4. Создание буклета программы школьного вечера.	Кейс 4. Создание буклета программы школьного вечера.	
42	Урок №42. Практическая работа.	Кейс 5. Создание пригласительных билетов на школьный вечер.	
43	Урок №43. Создание таблиц.	Вставка таблицы в документ. Добавление строк и столбцов в таблицу. Ширина столбцов и высота строк таблицы. Внешние и внутренние границы таблицы.	
44	Урок №44. Работа с таблицами. Кейс 6. Создание таблицы графика дежурств в классе.	Название таблицы. Объединение ячеек. Разбиение ячеек. Направление текста в ячейке. Изменение фона ячейки, таблицы. Изменение границ таблицы. <i>Кейс 6. Создание таблицы графика дежурств в классе.</i>	
45	Урок №45. Печать документа.	Параметры страницы. Нумерация страниц. Выбор страниц для печати. Предварительный просмотр. Печать документа.	

46-47	Уроки №46-47. Практическая работа. Создание сложных таблиц.	<p><i>Кейс 7. Создайте таблицу Расписание уроков. Придумайте, как сделать ее необычной, какого цвета будут ячейки, как будут располагаться столбцы? Самый простой вариант:</i></p> <table border="1" data-bbox="1155 228 1897 555"> <thead> <tr> <th>День</th> <th>Предмет</th> <th>День</th> <th>Предмет</th> </tr> </thead> <tbody> <tr><td rowspan="6">Понедельник</td><td>1</td><td rowspan="6">Четверг</td><td>1</td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td>3</td></tr> <tr><td>4</td><td>4</td></tr> <tr><td>5</td><td>5</td></tr> <tr><td>6</td><td>6</td></tr> <tr><td rowspan="6">Вторник</td><td>1</td><td rowspan="6">Пятница</td><td>1</td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td>3</td></tr> <tr><td>4</td><td>4</td></tr> <tr><td>5</td><td>5</td></tr> <tr><td>6</td><td>6</td></tr> <tr><td rowspan="6">Среда</td><td>1</td><td rowspan="6">Суббота</td><td>1</td></tr> <tr><td>2</td><td>2</td></tr> <tr><td>3</td><td>3</td></tr> <tr><td>4</td><td>4</td></tr> <tr><td>5</td><td>5</td></tr> <tr><td>6</td><td>6</td></tr> </tbody> </table> <p><i>Кейс 8. Создай свою визитку.</i></p>	День	Предмет	День	Предмет	Понедельник	1	Четверг	1	2	2	3	3	4	4	5	5	6	6	Вторник	1	Пятница	1	2	2	3	3	4	4	5	5	6	6	Среда	1	Суббота	1	2	2	3	3	4	4	5	5	6	6	
День	Предмет	День	Предмет																																														
Понедельник	1	Четверг	1																																														
	2		2																																														
	3		3																																														
	4		4																																														
	5		5																																														
	6		6																																														
Вторник	1	Пятница	1																																														
	2		2																																														
	3		3																																														
	4		4																																														
	5		5																																														
	6		6																																														
Среда	1	Суббота	1																																														
	2		2																																														
	3		3																																														
	4		4																																														
	5		5																																														
	6		6																																														
48	Урок №48. Знакомство с Google-docs.	Бесплатный сервис. Облачное хранилище. Обмен файлами. Кроссплатформенность. Коллективная работа.																																															
49	Урок №49. Начало работы с Google-docs.	Кейс 9 (большой, на всю тему изучения Google-документов). Создание папки нашего класса. Начало. 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.																																															
50	Урок №50. Работа с текстом в Google-docs.	Ввод и редактирование текста. Панель инструментов. Кейс 9. Где я ошибся? Работа в парах. Ученик создает гугл-документ, который открывает на доступ своему напарнику. В созданном документе он набирает (или берет из интернета небольшой текст), в котором нарочно делает ошибки. Напарник должен обнаружить все ошибки и их прокомментировать с помощью инструмента Комментарий.																																															
51	Урок №51. Работа с текстом в Google-docs.	Форматирование текста: шрифт, стиль, цвет, размер, выравнивание. Отступ. Межстрочный интервал. Экранная клавиатура. Кейс 9. Продолжение. Начнем наполнять файлы с информацией об учениках. Каждый ученик в своем файле пишет краткий рассказ про себя (меня зовут ..., я учусь ..., я увлекаюсь..., этим летом я был..., я хочу стать ... и т.п.) и красиво форматирует текст.																																															
52	Урок №52. Работа с изображениями.	Вставка рисунка. Размер рисунка, поворот, положение, прозрачность, обтекание																																															

		<p>текстом.</p> <p>Кейс 9. Продолжение.</p> <p>Продолжаем наполнять файлы с информацией об учениках. Ребята дополняют файлы с рассказом о себе изображениями (часть, возможно, потребуется принести из дома, например, свою фотографию, фотографию домашнего животного, или фото из путешествий). К некоторым изображениям стоит добавить интересные эффекты.</p>											
53	Урок №53. Работа со списками.	<p>Маркированный список. Нумерованный список. Шаблоны списков. Сброс нумерации.</p> <p>Кейс 9. Продолжение.</p> <p>Дополните свой файл списком стран, в которых вы бы хотели побывать, к каждой стране добавьте вложенным списком несколько городов, к каждому городу – несколько достопримечательностей этого города (тоже вложенным списком</p>											
54	Урок №54. Работа с таблицами.	<p>Вставка таблицы в документ. Вставка и удаление строк и столбцов в таблицу. Ширина и высота ячеек таблицы.</p> <p>Кейс 9. Продолжение.</p> <p>Дополните свой файл таблицей с вашим расписанием на неделю, и расписанием кружков.</p>											
55	Урок №55. Работа с таблицами.	<p>Объединение ячеек. Цвет фона. Цвет границ. Толщина границ. Стил.</p> <p>Кейс 9. Продолжение.</p> <p>Красиво раскрасьте таблицу со своим расписанием, созданную на прошлом уроке.</p>											
56	Урок №56. Работа с диаграммами.	<p>Вставка диаграммы в документ. Google – Таблица. Виды диаграмм: линейчатая, столбчатая, круговая, график. Область диаграммы, название, легенда. Расширенные настройки. Функция sparkline для создания миниатюрной диаграммы внутри ячейки.</p> <p>Кейс 11. Как я учусь?</p> <p>Возьмите свой дневник и создайте таблицу вида:</p> <p>Заполните ячейки с помощью дневника, записав туда все свои оценки, полученные за прошедшее время. Какие диаграммы вы сможете построить по этой таблице?</p> <table border="1" data-bbox="922 1102 1989 1289"> <tr> <td></td> <td>Сколькопятерок</td> <td>Сколькочетверок</td> <td>Сколькотроек</td> <td>Сколькодвоек</td> </tr> <tr> <td>Математика</td> <td></td> <td></td> <td></td> <td></td> </tr> </table>		Сколькопятерок	Сколькочетверок	Сколькотроек	Сколькодвоек	Математика					
	Сколькопятерок	Сколькочетверок	Сколькотроек	Сколькодвоек									
Математика													

		Русский язык					
		Английский язык					
		География					
		Физкультура					
57	Урок №57. Работа с рисунками.	Графический редактор внутри Google Docs. Линии, фигуры, текстовые поля. Инструмент Word Art. Кейс 9. Продолжение. С помощью изученным инструментов украсьте свой файл.					
58	Урок №58. Работа с формулами.	Вставка формулы в документ. Редактор формул. Кейс 12. Задачи для друга. Возьмите учебник математики (или найдите задачи в интернете) и создайте для друга гугл-документ с пятью задачами, например, на решение уравнений.					
59	Урок №59. Настройка стилей в Google – docs.	Создание своих стилей. Шрифт, размер, цвет, междустрочный интервал, эффекты.					
60	Урок №60. История изменений в Google – docs.	Хронология изменений документа. Возврат к нужной версии документа. Кейс 12. Вернись назад. Разбейтесь на пары. Зайдите в файл (из кейса 9) своего напарника, и внесите в него несколько изменений. Задача каждого ученика – восстановить версию до исправления					
61	Урок №61. Полезные сервисы в Google – docs.	Статистика. Проверка правописания. Автозамена. Голосовой ввод. GOOGLETRANSLATE и DETECTLANGUAGE. Кейс 13. Таинственный текст. Разбейтесь на пары. Создайте Google-таблицу. Запишите таинственное послание на русском языке, затем с помощью любого онлайн-переводчика переведите это послание на какой-нибудь язык, отличный от русского и английского, запишите результат в ячейке своей таблицы. Откройте доступ к таблице вашему напарнику. С помощью функций DETECTLANGUAGE определите, на каком языке записан текст в файле, который вам открыли. С помощью функции GOOGLETRANSLATE переведите этот текст на русский язык.					
62	Урок №62. Настройки доступа в Google – docs.	Доступ по ссылке. Уровни доступа: выключено, просматривать, комментировать, редактировать. Доступ определенным пользователям. Общий доступ. Доступ к папке с файлами.					

63	Урок №63. Совместная работа над документом.	Комментарии. Посоветовать правки. Закладки. Интеграция с Google Keep. Веб-буфер обмена.	
64-65	Урок №64-65 Горячие клавиши.	Сервис. Навигация. Форматирование текста. Форматирование абзаца. Редактирование.	
66-67	Урок №66-67 Плагины для Google – docs.	Установка плагинов из интернет-магазина Chrome Google Docsoffline – позволяет создавать документы при отсутствии подключения к Интернет.	
68-69	Галерея шаблонов в Google – Docs.	При создании стандартного документа, например, письмо или резюме можно воспользоваться заранее настроенным шаблоном. Кейс 14. Шаблон-реферат С помощью шаблона Реферат подготовьте небольшой рассказ о своем увлечении, может быть, это любимый вид спорта, решение задач, а, возможно, вы напишете о любимом домашнем животном.	

Учебный план 3-го года обучения

Данная программа предусматривает на реализацию программы по информатике в группе обучающихся 13-14 лет- 68 часов. Рабочая программа рассчитана на 34 учебных недели, 2 часа в неделю, общее количество часов — 68

Содержание программы

1. Графический язык программирования Blockly (16 часов)

Седа обучения. Демоверсии. Игры. Черепаша. Лабиринт. Учимся программировать: Робот. Blockly Duino – среда программирования роботов.

2. Введение в язык программирования Python (28 часа)

История создания языка. Установка Python. Структура программы. Типы данных. Ввод-вывод. Линейные алгоритмы. Алгоритмы светового течения. Циклы. Вложенные циклы. Списки. Функции. Модули. Работа с текстовыми файлами. Графический модуль PyTurtle. Графика с модулем tkinter. Создание приложения Painter.

3. Работа с табличным процессором LibreOffice.org Calc (24 часа)

Знакомство с офисным пакетом LibreOffice. Этапы работы с документом. Форматирование таблиц. Работа с листами. Навигация в электронных таблицах. Формат ячейки. Панели. Копирование данных и автозаполнение. Относительная и абсолютная адресация

я. Обработка данных. Диаграммы графики. Примеры задач моделирования и их решение с помощью электронных таблиц. Возможности Google Sheets.

Календарно-тематическое планирование «Информатика» III год (7 кл)

№п /п	Раздел Модули, кейсы, тема	Деятельность обучающихся	Дата провед
	Модуль 1.	Графический язык программирования Blockly (16 часов)	
1.	Урок №1. Знакомство с Blockly.	Разрабатывается и поддерживается компанией Google с 2012 года. Распространяется свободно. Не требует установки.	04.09
2.	Урок № 2. Кейс. Программирование – в играх. Урок – командная игра.	Разбейтесь на пары, откройте сайт с игрой “Банни идет домой”: http://blockly.ru/apps/bunny/index.html . Пройдите все три этапа обучения, выполнив задания.	11
3.	Урок 3. Кейс. Командная работа “Разберись со средой обучения”.	Ребята разбиваются на команды по несколько человек, на выбор получают одну из задач в демоверсии http://blockly.ru/training/demo1.html . Необходимо разобраться с тем, что в этой задаче происходит и подготовить краткий рассказ классу. Варианты задач: Демон №1 (Цикл со счётчиком, ветвление) Программа создает случайное двузначное положительное целое число и выводит на экран его и 7 следующих за ним нечётных чисел. Демон №2 (Цикл с условием) Программа генерирует случайные числа, пока их сумма остаётся меньше 100. После генерации очередного числа, на экран выводится само число и сумма сгенерированных чисел. Демон №3 (Одномерный числовой массив) Программа создает одномерный числовой массив, состоящий из 10 элементов и заполняет его случайными числами из диапазона -100...100. Каждый элемент массива выводится на экран. После создания весь массив целиком, а также минимальное и максимальное значения его элементов выводятся на экран. Демон №4 (Процедура) Функция принимает в качестве параметров два числа (a и b), заданных случайным образом, генерирует и выводит на экран 10 случайных чисел из диапазона a...b. Демон №5 (Функция) Функция принимает в качестве параметра целое положительное однозначное	18.

		число n и возвращает значение суммы факториалов всех чисел от 1 до n . Программа выводит на экран значения факториалов и суммы факториалов всех чисел от 1 до n . Демо №6 (Прямая рекурсия) Программа выводит на экран все числа в порядке их вычисления рекурсивной функцией $F(n)$ при выполнении вызова $F(9)$. (Задание №11 демоверсии ЕГЭ-2018) Демо №7 (Косвенная рекурсия) Сколько символов «звёздочка» будет напечатано на экране при выполнении вызова $F(11)$ (Задание №11 демоверсии ЕГЭ2016)	
4.	Урок №4	Кейс. Программирование как вызов. Командная работа. Пройдите лабиринт, изучив основные алгоритмические конструкции. Разбейтесь на пары, откройте страницу с игрой http://blockly.ru/apps/panda/index.html . Сколько заданий вы сможете выполнить за урок?	25
5.	Урок №5. Кейс. Исследуем игры для программистов.	Работа в команде. Разбейте школьников на команды, каждая команда получает для изучения одну из игр со страницы http://blockly.ru/games.html (Птица, Пруд, JS Пруд, Фильм). Ребята должны разобраться тем, как решать задачи в каждой игре, пройти как можно больше уровней, подготовить рассказ об игре, ее особенностях, самых интересных моментах для остального класса.	02.10
6.	Урок № 6. «Черепаша»	Команды движения. Повороты. Управление пером.	09.10
7.	Урок № 7. «Черепаша»	Цвет. Смешение цветов. Логические операции. Циклы. Математика.	16.10
8.	Урок № 8. Практическая работа. Кейс. Командная игра:	Кейс. Командная игра:проведи Панду через суперлабиринт. Чья команда быстрее справится с заданием? Немного теории. Правила прохождения лабиринта. Чтобы провести Панду через этот Суперлабиринт, Вам необходимо проявить незаурядные программистские способности. Удачи! Вперёд!	23.10
9.	Урок № 9. Кейс. Учимся программировать. Робот	. Демоверсии. Командная работа. Ребята разбиваются на команды по несколько человек, на выбор получают одну из задач в демоверсии http://blockly.ru/blockly-robot/apps/demo1.html . Необходимо разобраться с тем, что в этой задаче происходит и подготовить краткий рассказ классу. Демоверсии I. №1 (Основы) Робот проходит половину круга против часовой стрелки и останавливается. Демоверсии I. №2 (Неполное ветвление) Робот генерирует случайным образом из диапазона -100...100 целочисленную переменную x и выводит её на экран. Если число x положительное, то Робот вычисляет корень квадратный из x , выводит его на экран и делает несколько шагов вперёд. Демоверсии I. №3 (Полное ветвление) Робот генерирует случайным образом из диапазона -100...100 целочисленную переменную x и выводит её на экран. Если число x положительное, то Робот вычисляет корень квадратный из x , выводит его на экран и делает несколько шагов вперёд, иначе – выводит надпись: «Вычислить выполнении вызова $F(9)$ – (задание №11 демоверсии ЕГЭ-2018). Вывод на экран каждого нового числа сопровождается одним невозможно» и делает несколько шагов назад. Демоверсии I. №4 (Цикл со счётчиком) Робот генерирует случайным образом однозначное положительное число n , а затем делает n шагов по кругу. Демоверсии I. №5 (Цикл с условием) Робот генерирует случайные числа, пока их сумма остаётся меньше 100. После генерации очередного числа, Робот выводит на экран само число x и сумму сгенерированных чисел s , а затем делает шаги по кругу. Робот останавливается, когда сумма чисел шагом Робота.	13.11

		<p>превысит 100. Демоверсии II http://blockly.ru/blockly-robot/apps/demo2.html Демоверсии II. №1 (Одномерный числовой массив) Робот создаёт одномерный числовой массив и заполняет его случайными однозначными числами. Заполнение каждого элемента и вывод его на экран сопровождается одним шагом Робота. После заполнения всего массива, Робот выводит на экран его содержимое и максимальное значение.</p> <p>Демоверсии II. №2 (Процедура) Робот создаёт процедуру, которая принимает в качестве параметров два числа (a и b), заданных случайным образом, генерирует и выводит на экран 10 случайных чисел из диапазона a...b. Вывод на экран каждого нового числа сопровождается одним шагом Робота.</p> <p>Демоверсии II. №3 (Функция) Робот создаёт функцию, которая принимает в качестве параметра целое положительное однозначное число n и возвращает значение суммы факториалов всех чисел от 1 до n. Робот выводит на экран значения числа n, факториалов и суммы факториалов всех чисел от 1 до n. Количество шагов Робота равно значению числа n.</p> <p>Демоверсии II. №4 (Прямая рекурсия) Робот выводит на экран все числа в порядке их вычисления рекурсивной функцией F(n) при выполнении вызова F(9) – (задание №11 демоверсии ЕГЭ-2018). Вывод на экран каждого нового числа сопровождается одним шагом Робота.</p> <p>Демоверсии II. №5 (Косвенная рекурсия) Робот выводит на экран символы «звёздочка» при выполнении вызова F(11) (задание №11 демоверсии ЕГЭ-2016). Необходимо подсчитать количество выводимых на экран символов. Вывод на экран каждого нового символа сопровождается одним шагом Робота.</p> <p>Демоверсии II. №5 (Косвенная рекурсия) Робот выводит на экран символы «звёздочка» при выполнении вызова F(11) (задание №11 демоверсии ЕГЭ-2016). Необходимо подсчитать количество выводимых на экран символов. Вывод на экран каждого нового символа сопровождается одним шагом Робота.</p>	
10.	Урок № 10. Продолжение проекта с прошлого урока.	Презентация рассказов.	20.11
11.	Урок № 11. BlocklyDuino – среда программирования роботов.	Ввод/вывод Индикаторы Серводвигатели.	27.11
12.	Урок № 12. BlocklyDuino – среда программирования роботов.	Связь Логические Циклы Математика	04.12
13.	Урок № 13. BlocklyDuino – среда программирования	Текст Массивы Переменные Функции	11.12

	роботов.		
14.	Урок № 14. Практическая работа	Кейс: Командная работа. Программирование роботов с помощью языка BlocklyDuino.	18.12
15.	Модуль 2. Введение в язык программирования Python Урок № 15. Знакомство с языком программирования Python.	История создания. Python - это интерпретируемый язык программирования. Python - это полноценный язык программирования. Python – свободно распространяемый язык программирования. Дзэн Питона. Команда «importthis» Установка языка Python. Интерактивный режим. Создание первой программы: «Hello, world»	25.12
16.	Урок № 16. Структура программы. Типы данных. Переменные. Ввод и вывод данных.	Ввод и вывод данных. Операции. Создание и редактирование программ в среде IDLE. Типы данных: целые числа, числа с плавающей точкой, строки. Запуск программ. Кейс: создайте учебную задачу для соседа на ввод/вывод в Python или на последовательность выполнения арифметических операций.	15.01
	Модуль2.	Введение в язык программирования Python (28 час)	
17.	Урок № 1 Линейные алгоритмы. Блок-схема линейного алгоритма.	Ввод данных с клавиатуры. Правила записи арифметических выражений. Множественное присваивание. Случайные числа. Порядок выполнения операций. Вывод на экран. Кейс. Работа в парах: запишите выражение по правилам языка Python и предложите напарнику вычислить его, не используя интерпретатора Python. Проверьте себя в среде разработки.	22
18.	Урок № 2 Ветвящиеся алгоритмы. Инструкция ветвления if.	Ветвление. Полная и сокращенная формы. Логический тип данных. Логическое высказывание. Операции сравнения.	29.01
19.	Урок № 3 Ветвящиеся алгоритмы.	Множественное ветвление. Инструкция ветвления if ... else. Мини-кейс: Программа «Открыто / закрыто»	05.02
20.	Урок № 4. Циклические алгоритмы. Цикл с предусловием.	Блок-схема цикла с предусловием. Инструкция while. Мини-кейс: Подсчет суммы цифр целого числа.	12
21.	Урок № 5 Циклические алгоритмы.	Мини-кейс: Поиск НОД двух целых чисел.	19
22.	Урок № 6. Циклические алгоритмы. Цикл с параметром. Блок-схема цикла с параметром.	Генерация диапазона значений. Функция range (). Мини-кейс: вычисление факториала N.	26.02
23.	Урок № 7 Вложенные	Время работы программы, содержащей вложенные циклы. Мини-кейс: написать программу,	04.03

	циклы. Цикл в цикле.	выводящую на экран таблицу Пифагора.	
24.	Урок № 8 Списки. Массивы и списки.	Индекс (номер элемента списка). Создание, ввод и вывод списков. Просмотр элементов списка. Добавление удаление элементов списка. Количество элементов списка может меняться во время выполнения программы. Мини-кейс: Программа «Рекорды»	11
25.	Урок № 9. Списки.	Копирование списков. Линейный поиск. Сортировка списка. Мини-кейс: Создать список учеников класса, отсортировать его и написать программу, которая спрашивает у пользователя число N и выводит фамилию и имя N-ого по алфавиту ученика, или N-ого с конца алфавита.	18
26.	Урок № 10. Функции. Вспомогательные алгоритмы.	Объявление функции. Правила описания функций. Параметры и возвращаемые значения. Области видимости. Чтение глобальной переменной внутри функции.	
27.	Урок № 11. Модули.	Модули из стандартной библиотеки. Инструкция import. Инструкция from. Использование псевдонимов Создание своего модуля.	
28.	Урок № 12. Работа с текстовыми файлами.	Функция open и её аргументы. Чтение из файла. Запись в файл.	
29.	Урок № 13. Практическая работа.	Кейс: Создание игры «Викторина». Вопросы и варианты ответов с указанием правильного хранятся в текстовом файле. Вопросы задаются последовательно. Игроку предлагается выбрать правильный ответ из предложенных вариантов. В случае правильно ответа результат увеличивается на единицу, и игроку выводится сообщение: «верно». В случае неверного ответа выводится сообщение: «не верно». После ответа на последний вопрос на экран выводится результат (количество верных ответов).	
30.	Урок № 14 Практическая работа.	Кейс: Завершение работы над Викторинной. Работа в парах: ребята разбиваются на пары, и тестируют игру, написанную напарником.	
31.	Урок № 15. Графический модуль PyTurtle.	Импорт модуля turtle. Внешность, формы «черепашки». Размер «черепашки». Цвет «черепашки». Отпечаток «черепашки».	
32.	Урок № 16. Графический модуль PyTurtle Команды перемещения.	Команды поворота. Рисуем домики, машинки и т.п. Раскрашиваем картинки. Кейс. Нарисуй самый необычный домик.	
33.	Урок № 17 Графический модуль PyTurtle.	Возможности рисования в PyTurtle. Цвет и размер пера. Решение задач с использованием линейных алгоритмов.	
34.	Урок № 18 Графический модуль PyTurtle. Положение	Координатная плоскость. Направление «черепашки». Команда поставить точку. Циклические конструкции в PyTurtle	

	«черепашки».		
35.	Урок № 19. Графический модуль PyTurtle. Создание геометрических фигур в PyTurtle.	Рисование спиралей в PyTurtle. Конструкции ветвления алгоритмов в PyTurtle.	
36.	Урок № 20 Практическая работа.	Кейс: Рисуем снежинки. Конкурс на самую необычную снежинку.	
37.	Урок № 21. Графика с модулем tkinter в Python.	Создание графического интерфейса. Импорт модуля tkinter. Создание базового окна. Создание рамки. Создание метки. Создание кнопок.	
38.	Урок № 22. Графика с модулем tkinter в Python.	Менеджер размещения Grid. Текстовые поля и области. Применение флажков. Применение переключателей.	
39.	Урок № 23. Виджет Canvas. Холст.	Линии. Цвета. Формы. Рисуем изображение. Рисуем текст.	
40.	Урок № 24. Практическая работа	Кейс: Командная работа. «Создаем приложение «Painter», позволяющее рисовать мышкой на экране (с палитрой цветов и ластиком)»	
41.	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 электронных таблиц. Меню, панели инструментов, строка состояния. Режимы работы с документом.	
42.	Урок № 26. Этапы работы с документом. Создание нового документа, ввод данных.	Сохранение, открытие и закрытие документа. Редактирование таблиц: копирование, вставка, перемещение, удаление содержимого ячеек. Отмена действий.	
43.	Урок № 27 Форматирование таблиц. Шрифты, стили, размер шрифта, границы.	Работа с цветом. Выравнивание, перенос по словам Объединение ячеек.	
44.	Урок № 28. Работа с листами.	Ярлыки листов. Вставка, удаление листов. Переименование листов. Цвет ярлыка листа.	

	Модуль 3.	Работа с табличным процессором LibreOffice.org Calc (28 часов)	
45.	Урок № 1 Навигация в электронных таблицах.	Использование мыши. Использование навигатора. Ссылки на ячейки. Использование клавиш Enter, Tab, влево, вправо, вверх, вниз, Home, End, Page Up и Page Down	
46.	Урок № 2. Навигация по листам Использование клавиатуры.	Использование мыши. Использование Навигатора	
47.	Урок № 3 Строка состояния Номер листа.	Режим вставки. Стилль листа. Статус изменения. Сумма, среднее значение. Масштаб.	
48.	Урок № 4 Боковая панель.	Настройки боковой панели. Свойства. Стили. Галерея. Навигатор. Функции.	
49.	Урок № 5. Выбор ячеек. Диапазоны.	Диапазон смежных ячеек. Диапазон не смежных ячеек. Весь лист. Выделение строк. Выделение столбцов.	
50.	Урок № 6. Формат ячеек. Числовой.	Процентный. Денежный. Дата, время. Текст.	
51.	Урок № 7. Панель формул. Поле «Имя».	Мастер функций « f(x) » Сумма « Σ » Формула « = » Строка ввода.	
52.	Урок № 8. Мастер функций. Категории функций.	Поиск необходимой функции. Работа с аргументами функций.	
53.	Урок № 9. Копирование ячеек. Копирование данных.	Копирование формул. Автозаполнение ячеек.	
54.	Урок № 10 Практическая работа.	Кейс: Создание платежной ведомости. Отобразить начисленную З/П, удержанный НДФЛ, сумму к выдаче, итого. Найти максимальную, минимальную и среднюю З/П	
55.	Урок № 11 Относительная и абсолютная адресация.	Ссылки на ячейки текущего листа. Ссылки на ячейки из других листов. Использование знака « \$ » в ссылках на ячейки. Закрепление номера строки. Закрепление имени столбца.	
56.	Урок № 12 Обработка данных. Сортировка.	Фильтр. Скрыть, отобразить данные.	
57.	Урок № 13. Практическая работа.	Кейс: Создание таблицы расчёта стоимости перевозки грузов. В нашем распоряжении три автомобиля, заданной грузоподъемности. Для каждого из них известна стоимость перевозки на один километр. В таблице десять пунктов назначения. Для каждого из них известно расстояние до склада и необходимое количество грузов, которые будут доставлены со склада. Рас - считать общие затраты на все	

		необходимые перевозки.	
58.	Урок № 14 Диаграммы и графики. Мастер диаграмм.	Типы диаграмм. Диапазон данных. Ряды данных. Элементы диаграммы. Кейс: Построение графиков функций.	
59.	Урок № 15 Примеры задач моделирования и их решение с помощью электронных таблиц.	Кейс: Моделирование движения твердого тела в поле тяжести Земли	
60.	Урок № 16. Возможности Google Sheets.	Бесплатный сервис. Облачное хранилище. Обмен файлами. Кроссплатформенность. Коллективная работа. Учетная запись Gmail. Google – диск	
61.	Урок № 17. Панель инструментов Google Sheets. Шрифт.	Размер шрифта. Форматирование текста Форматирование ячеек Выравнивание текста. Масштаб.	
62.	Урок № 18 Операции с ячейками, строками и столбцами.	Вставка, удаление строк и столбцов. Закрепление строк и столбцов. Перемещение строк и столбцов. История изменений в «Google Sheets».	
63.	Урок № 19. Функции в «Google Sheets».	Категории функций. Поиск необходимой функции. Работа с аргументами функций	
64.	Урок № 20 Относительная и абсолютная адресация. Ссылки на ячейки текущего листа.	Ссылки на ячейки из других листов. Распространение формул. Использование знака « \$ » в ссылках на ячейки. Закрепление номера строки. Закрепление имени столбца.	
65.	Урок № 21. Обработка данных. Сортировка.	Фильтр. Скрыть, отобразить данные.	
66.	Урок № 22. Диаграммы и графики.	Редактор диаграмм. Типы диаграмм. Диапазон данных. Ряды данных. Элементы диаграммы.	
67.	Урок № 23. Настройки доступа в Google Sheets.	Доступ по ссылке. Уровни доступа: выключено, просматривать, комментировать, редактировать. Доступ определенным пользователям. Общий доступ. Доступ к папке с файлами.	
68.	Урок № 28 Отчет	Выступление обучающихся со своими творческими работами	

Учебный план IV года обучения

Данная программа предусматривает на реализацию программы по информатике в группе обучающихся 14-15 лет
Рабочая программа рассчитана на 34 учебных недели, 2 часа в неделю, общее количество часов — 68 часов.

Содержание программы

1. Основы языка Python (12 часов)

Ввод-вывод данных. Типы данных. Работа со строками. Списки. Условная инструкция. Циклы for и while. Функции. Разработка несложных консольных приложений.

2.Создание приложений с помощью tkinter (16 часов)

Работа с модулем tkinter. Виджеты. Конфигурация виджетов. Реакция на события. Упаковщики виджетов. Рисование на холсте canvas. Управление нарисованными объектами с помощью клавиатуры. Разработка и создание GUI-приложений “пингпонг”, “сапер”.

3.Криптография (26 часов)

История криптографии. Знаменитые шифры (атбаш, сцитала, шифр Цезаря, квадрат Полибия, решетка Кардано). Создание криптографического приложения с помощью tkinter. Шифры, практически невозможно разгадать (шифр Виженера). Современные алгоритмы шифрования. Открытый и закрытый ключи. Электронная подпись. Кодирование текста. Работа с файлами в Python. Продвинутое возможности Python: словари. Дополнительные библиотеки языка Python для работы с датами и временем. Разработка игрового приложения “Мемори”.

4.Искусственный интеллект (6 часа)

Что такое ИИ? Алан Тьюринг и его работы. Вычислительная сложность алгоритма. Идея двоичного поиска. Создание приложения, отгадывающего возраст.

5.Продвинутые библиотеки языка Python. Pygame (12 часов)

Обзор дополнительных библиотек для работы с графическим интерфейсом. Библиотека Pygame. Шаблон программы. Геометрические примитивы в Pygame. Простая анимация в Pygame. События клавиатуры. События мыши. Дополнительные поверхности. Работа с текстом. Музыка.

Календарно-тематическое планирование «Информатика» Криптография IV года обучения(8 кл)

№п /п	Модули, кейсы, тема	Деятельность обучающихся	Дата провед
	Модуль1.	Основы языка Python (12 часов)	
1.	Модуль 1. Основы языка Python. Урок 1. Вводное занятие. Техника безопасности.	Знакомство с программой курса. Краткая история языка Python, кто создал и почему так назвали. Что можно сделать, зная язык программирования Python? Где в принципе программисты могут писать программы и как они их потом запускают? Отличие компилируемых и интерпретируемых языков программирования. Демонстрация установки интерпретатора Python. Используем Python как калькулятор: интерактивный режим работы с интерпретатором Python, вычисляем сумму чисел,	07.09

		вычисляем 2 в степени 100. Запуск встроенной среды разработки IDLE, работа в командной строке Shell – аналог интерактивного режима. Тест по технике безопасности и простым фактам о языке Python.	
2.	Урок 2. Основы языка Python. Ввод-вывод данных, числа и строки, операции с числами и строками.	Встроенная среда разработки IDLE. Знакомство с функциями print() и input(). Особенность ввода данных на языке Python – данные считываются в виде строки. Первая программа: считываем число и выводим его же. Простые операции со строками, конкатенация строк. Создание простой программы, спрашивающей имя пользователя и затем приветствующей его. Создание простой программы, спрашивающей у пользователя имя, количество лет, а затем выводящей имя столько раз, сколько пользователю лет.	14
3.	Урок 3. Основы языка Python. Операции с числами и строками. Типы данных int и str.	Преобразование типов с помощью операторов int() и str(). Простые вычисления в Python, например, “Чему равно str(2 + 3) * int('2' + '3')? Постарайтесь дать ответ, не используя интерпретатора Python”. Особенности функций input() и print(). Изменение поведения функции print() с помощью параметров sep и end. Символ перевода строки '\n'. Тест на понимание работы функций print, input, str, int. Простые программы на взаимодействие с системой. Два возможных примера описаны ниже. Пример 1. Поле Чудес Программа приветствует Буратино и спрашивает, сколько у него монет, после ответа пользователя программа сообщает Буратино, сколько денег у него будет завтра. print('Привет, Буратино!') print('Зарой все свои деньги здесь.') n = input('Сколько у тебя монет?') n = int(n) print('Завтра у тебя будет', n * 100, 'монет') print('Приходи завтра!') Пример 2. Инопланетянин Программа приветствует пользователя, узнает его возраст и печатает соответствующее количество сердечек. print('Привет, землянин!') n = input('Сколько тебе лет?') n = int(n) print('Шлю тебе', n, 'сердечек') print('\u2764' * n)	21
4.	Урок 4. Основы языка Python. Условная инструкция в Python.	Условная инструкция if-else в Python. Блок-схема ветвления. Неполное и полное ветвление. Отступы в Python – почему это важно? Операторы сравнения ==, !=, <=, >, >=, . Простые программы на использование условной инструкции. Несколько возможных примеров описаны ниже. Предложите школьникам придумать самим подобные программы и реализовать их. Программа 1. print('На улице темно? да/нет') ans = input() if ans == 'да': print('спокойной ночи!') Программа 2. ans = input('У тебя есть щупальца? да/нет') if ans == 'да': print('Привет, осьминог!') else: print('Привет, человек!')	28.09
5.	Урок 5. Основы языка Python. Условная инструкция в Python, elif, логические операции.	Как быть, если одновременно нужно проверить истинность нескольких условий? Или то, что верно хотя бы одно условие из нескольких? На помощь приходят логические операции or и and. Простые программы, например, Программа 1. Пройти на аттракцион rost = input('Каков твой рост в сантиметрах?\n') vozrast = input('Сколько тебе лет?\n') if int(rost) > 120 and int(vozrast) >= 3: print('Проходи!') else: print('Подрасти еще немного!') Если нужно больше двух ветвлений, стоит использовать оператор elif. Показать несколько простых примеров с использованием elif и без использования elif. Создание простых программ, где удобно использовать elif, например, Программа 2. Погода weather = input('Какая сегодня	05.10

		погода? дождь/снег/солнце’) ifweather == ‘дождь’: print(‘Захватите зонтик!’) elifweather == ‘снег’: print(‘Не забудьте варежки!’) else: print(‘Ура! Берем солнечные очки!’) Что такое модули и как их подключать? Модуль random и некоторые его функции. Примеры использования модуля random. Программа 3. Чье число больше? importrandom n = input(‘Загадай число от 1 до 5, и я тоже загадаю\n’) n = int(n) m = random.randrange(1, 10) if m > n: print(‘Мое число’, m, ‘больше твоего, ура!’) else: print(‘Я загадал’, m, ‘и проиграл...’) Вопрос к школьникам: модернизируйте программу так, чтобы рассматривался и случай равенства чисел.	
6.	Урок 6. Цикл for в Python.	Использование цикла для перебора объектов (конструкция foreleminobject, где object – строка, кортеж, иной итерируемый объект). Использование цикла для выполнения заданного количества операций: три способа записи for i inrange(n); for i inrange(a, b); for i inrange(a, b, d)). Простые программы, например: • Как напечатать слово привет столько раз, какое число введет пользователь (каждый “привет” должен быть в отдельной строке)? • Напечатайте числа от 1 до n (n определяется пользователем) в прямом и в обратном порядке, только четные, только кратные трем, и т.п. • Пользователь задает число n, программа выводит n строк, в первой строке одна звездочка, второй две и т.д.	12.10
7.	Урок 7. Цикл while Цикл с предусловием. Блок-схема.	Бесконечный цикл. Оператор break для выхода из цикла. Переменная-счетчик для подсчета количества операций в цикле. Простые программы, например, такие, как приведены ниже: Программа 1. Сладкоежка candies = 0 ans = ‘да’ whileans == ‘да’: candies = candies + 1 print(‘Съедено конфет:’, candies) ans = input(‘Хочешь еще конфетку? да/нет\n’) Программа 2. Бесконечный цикл: while True: print(‘я буду работать вечно!’) Программа 3. Надоедливая программа: while True: ans = input(‘я тебе надоела? да/нет\n’) ifans == ‘да’: print(‘Как некрасиво! Ухожу...’) break Программа 4. Угадай, как меня зовут? print(‘Угадай, как меня зовут!’) ans = input() whileans != ‘python’: print(‘Не угадал! Попробуй еще раз’) ans = input() print(‘Правильно, я python!’)	19.10
8.	Урок 8. Строки Нумерация символов строк. Отрицательная нумерация. Функция длины строки.	Перебор символов строки с помощью цикла for (два способа: forsmbinstr и по индексу). Срезы строк. Простые программы: напечатать заданную пользователем строку задом наперед; напечатать все символы заданной пользователем строки по одному в строке; изменить заданную пользователем строку так, чтобы символы были разделены знаком звездочка, и т.п.	26.10
9.	Урок 9. Списки. Что такое список в Python? Нумерация элементов списка.	Длина списка. Аналогия со строками. Модуль random для работы со списками, перемешивание элементов списка, выбор произвольного элемента. Простые программы, например, в списке можно хранить имена членов вашей семьи и с помощью модуля random выбирать того, кто моет посуду сегодня после ужина.	09.11
10.	Урок 10. Функции. Встроенные функции в	Функции как инструмент многократного использования одного и того же кода. Локальные и глобальные переменные. Самостоятельное создание простых функций, например, функции,	16.11

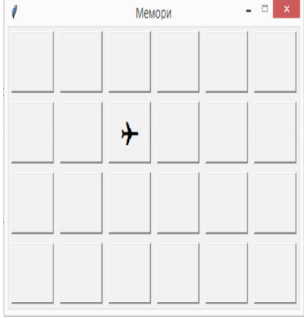
	Python.	переводящей количество дней в количество секунд в этих днях. Некоторые встроенные функции в Python: max, min, sum, reverse, встроенные функции изменения регистра букв lower и upper.	
11.	Урок 11. Кейс 1. Создание программы-теста из нескольких вопросов.	Создание тематического теста. Программа ведет диалог с пользователем, задавая ему 1-3 вопроса, каждый по одной и той же схеме (задается вопрос, далее дается три попытки на ответ, если одна из попыток заканчивается верным ответом, выводится строка “Ответ верный!”, если ни в одной попытке не было дано верного ответа, просто показывается верный ответ.) Программа должна засчитывать верный ответ без учета регистра, для реализации этого удобно использовать встроенную функцию lower или встроенную функцию upper. Пример подобной программы, задающей пользователю вопрос про футбол, приведен ниже. <pre>print('Тест про футбол') ans = input('Вопрос 1. В какой стране проходил последний чемпионат мира по футболу?\n') checker = True attempt = 0 while checker and attempt < 3: if ans.lower() == «россия» or ans.lower() == «в россии»: print('Ответ верный!') checker = False else: if attempt < 2: ans = input('Попробуй еще раз!\n') attempt = attempt + 1 if attempt == 3: print('Попытки закончились... Верный ответ: Россия или В России')</pre>	23.11
12.	Урок 12. Завершение работы над программой-тестом.	Добавление в тест еще одного-двух вопросов. Добавление счетчика очков (за каждый верный ответ пользователю начисляется определенное количество очков, после прохождения теста — это количество очков выводится на экран). Обсуждение со школьниками того, как использование функций могло бы помочь сделать код короче (все вопросы теста однотипны), реализация при достаточном уровне подготовленности школьников.	30.11
	Модуль 2.	Создание приложений с помощью tkinter (16 часов)	
13.	Урок 1 Знакомство с модулем tkinter. Подключение модуля tkinter.	Краткий обзор возможностей, демонстрация уже готовых приложений, разработанных с помощью tkinter. Создание простого GUI-приложения, состоящего из одного виджета, например, кнопки. Обсуждение общей последовательности действий, необходимой для создания GUI-приложения. Совершенствуем наше приложение, добавив реакцию на событие, например, щелчка левой кнопки мыши (например, виджет меняет цвет).	07.12
14.	Урок 2. Что такое виджеты, конфигурация виджетов.	Некоторые виджеты: кнопка, текстовое поле, поле ввода, метка. Конфигурация виджетов, методы config и configure. Получение информации о состоянии виджета.	14.12
15.	Урок 3. События в tkinter.	Как научить виджет реагировать на события – различные способы. Методы bind и unbind. Кейс 2. Создайте интеллект-карту на тему “События в tkinter”.	21.12
16.	Урок 4. Создание простых приложений.	Продолжаем знакомиться с tkinter на простых примерах. Создание приложения, состоящего из кнопки, подсчитывающей количество нажатий, и исчезающей после определенного количества нажатий. Создание простого приложения, состоящего из двух кнопок “Привет!” и “Пока...” и текстового поля: при нажатии на одну из кнопок в текстовом поле печатается приветствие, при нажатии на вторую кнопку, приложение закрывается	28.12
17.	Урок 5. Создание простых приложений.	Приложение “Радуга”: простое приложение, состоящее из кнопок, окрашенных в основные цвета радуги и текстового поля. При нажатии на кнопку, в текстовом поле появляется название цвета	11.01

		нажатой кнопки.	
18.	Урок 6.Создание простых приложений.	Упаковщики виджетов: pack(), grid(), place(). Создание простого калькулятора из нескольких кнопок и текстового поля для вывода результата вычислений.	18
19.	Урок 7. Продвинутое рисование в tkinter.	Рисование на холсте canvas в tkinter. Создание различных геометрических фигур.	25
20.	Урок 8.Движение нарисованных объектов.	Движение нарисованных объектов с помощью клавиатуры.	01.02
21.	Урок 9.Кейс 3. Создание игрового приложения “Пинг-понг”.	Создание простой версии игры “пинг-понг”. Создаем мячик и ракетку с помощью canvas. “Учим” ракетку двигаться влево-вправо при нажатии кнопок со стрелками влево-вправо соответственно. Как сделать так, чтобы ракетка двигалась все время и при нажатии кнопки только меняла направление движения. Как сделать так, чтобы ракетка не выходила за левую и правую границы игрового поля.	08
22.	Урок 10. Создание игрового приложения “Пинг-понг”. Доделываем приложение пинг-понг.	Добавляем правильное движение мячика – с отскоком от стен и от ракетки. Добавляем счетчик количества ударов мяча о ракетку.	15
23.	Урок 11.Завершение работы над приложением “пинг-понг”.	Работа в парах: ребята делятся на пары программист-тестер, и тестируют созданные приложения.	22
24.	Урок 12. Общая идея игры сапер. Двумерные списки, вложенные циклы.	Как расположить виджеты Label в виде прямоугольника. Раскрасим минное поле в шахматном порядке.	29
25.	Урок 13. Приложение “Сапер”.	Информация о минах. Двумерный битовый список для хранения информации о минах в игре. Создаем и заполняем его случайным образом. Как с помощью модуля random можно контролировать количество мин в игре? (Например, выбираем случайное целое число от 1 до 100, если оно оказалось меньшим, чем 25, в соответствующую ячейку пишем 1 – это будет мина, иначе 0 – пустая ячейка. Обсуждение со школьниками, много или мало мин в этом случае стоит ожидать).	07.03
26.	Урок 14. Приложение “Сапер”.	Как узнать, сколько мин среди соседей данной клетки и как эту информацию хранить? По созданному двумерному списку мин создаем список, в каждой ячейке которого находится информация о том, сколько мин среди соседей данной клетки (соседями считаются клетки, имеющие с данной клеткой общую вершину или сторону).	14
27.	Урок 15. Создаем “мозг” игры. Пишем “мозг” игры сапер.	Что происходит в тот момент, когда пользователь нажимает на клетку, описываем реакцию виджета на событие.	21.03
28.	Урок 16. Завершение	Доделываем приложение, тестируем, убираем недостатки. Кейс 5. Как представить свой проект на	

	работы над приложением “Сапер”.	конференции? Ребята разбиваются на группы, выбирают лучшее из созданных ими приложений, рассказывают о процессе создания.	
	Модуль 3.	Криптография (26 часов)	
29.	Урок 1. История криптографии.	Старинные шифры Что такое криптография и чем она занимается? Как шифровали сообщения в древности? Старинные шифры атбаш и считала (шифр Древней Спарты).	
30.	Урок 30. История криптографии.	Старинные шифры Чем отличается стеганография от криптографии. Диск и линейка Энея. Шифр Цезаря.	
31.	Урок 31. История криптографии. Старинные шифры.	Шифрование с использованием таблиц. Квадрат Полибия. Решетка Кардано.	
32.	Урок 32. Игровое занятие. Разгадываем шифры.	Создание и расшифровка “таинственных посланий” с использованием вспомогательных материалов – карандаша и полоски бумаги для шифра считала, диска для шифра Цезаря и т.д.	
33.	Урок 33. Кейс 6. Создание криптографических приложений с помощью tkinter.	Диалоговые окна <code>simpledialog</code> и <code>messagebox</code> в <code>tkinter</code> . Создание приложения, запрашивающего текст для шифровки и возвращающего зашифрованное сообщение для шифра атбаш. Творческое задание: придумать свой несложный шифр перестановки и изменить приложение так, чтобы оно зашифровывало текст вашим способом (вариант шифра: переставить местами соседние буквы сообщения). Модернизация программы: пользователю сообщается название шифра, далее на выбор можно выбрать, хочет ли пользователь зашифровать текст, или, наоборот, расшифровать его.	
34.	Урок 34. Создание криптографического приложения «Шифр Цезаря».	Продвинутое криптографическое приложение Шифр Цезаря. Создание приложения, которое по выбору пользователя может как зашифровать сообщение шифром Цезаря с данным ключом (значение ключа тоже задает пользователь), либо расшифровать сообщение, зашифрованное шифром Цезаря с произвольным ключом. Идея перебора всех возможных ключей.	
35.	Урок 35. Завершение работы над приложением.		
36.	Урок 36. Модификация и развитие шифра Цезаря.	Шифры, которые практически невозможно разгадать. Шифр Виженера.	
37.	Урок 37. Современные алгоритмы шифрования.	Современные алгоритмы шифрования и основные принципы их работы. Кто и зачем придумал RSA? Идея открытого и закрытого ключа.	
38.	Урок 38. Современные алгоритмы шифрования.	Еще раз обсуждение того, что такое открытый и закрытые ключи. Идея односторонней функции с лазейкой (трудно вычислимой информацией). Обсуждение нескольких несложных способов шифрования с лазейкой (например, берется конкретная книга и буквы алфавита шифруются с помощью этой книги – шифром для буквы А может быть первое слово на первой странице, ..., для буквы Я – тридцать третье слово на тридцать третьей странице и так далее. В данном случае лазейкой является конкретная книга, используемая для шифрования.)	
39.	Урок 39. Игровое соревновательное занятие	Придумывание школьниками своих шифров для одноклассников и последующее разгадывание сообщений.	

	“Взламываем шифр с открытым ключом”.		
40.	Урок 40. Что такое электронная подпись?	Протокол аутентификации сообщений. Протокол электронно-цифровой подписи. Алгоритмы проверки электронной подписи. Алгоритм генерации электронной подписи.	
41.	Урок 41. Кодирование текста. Кодирование информации в компьютере.	Сколько информации можно закодировать с помощью 8 бит? Таблица ASCII. Как кодируют русские буквы и почему вместо понятного текста иногда в электронных сообщениях мы видим “крокозябры”. Стандартные кодовые таблицы для русского алфавита. Unicode. Вывод символа с помощью языка Python по номеру в кодовой таблице.	
42.	Урок 42. Работа с файлами в Python.	Чтение данных из файла. Методы <code>readline()</code> , <code>readlines()</code> и <code>read()</code> . Удаление концевых символов строки с помощью метода <code>rstrip()</code> . Вывод данных файл. Создание программы, считывающей данные из файла посимвольно.	
43.	Урок 43 . Кейс 7 Разработка приложения, сохраняющего данные в файле.	Подготовка к созданию приложения, позволяющего открывать текстовые файлы в текстовом поле в <code>tkinter</code> или создавать новые, а также редактировать их и сохранять. Многострочное текстовое поле <code>Text</code> в <code>tkinter</code> . Конфигурация виджета <code>Text</code> (размеры, шрифт, цвет).	
44.	Урок 44. Разработка приложения, сохраняющего данные в файле.	Методы <code>insert()</code> , <code>get()</code> и <code>delete()</code> виджета <code>Text</code> . Создание простого приложения, содержащего многострочное текстовое поле и две кнопки, позволяющие добавлять текст и удалять его.	
45.	Урок 45. Разработка приложения, сохраняющего данные в файле.	Модуль <code>filedialog</code> в <code>tkinter</code> (диалоговые окна открытия и сохранения файлов). Создание приложения, позволяющего открывать текстовые файлы в текстовом поле в <code>tkinter</code> или создавать новые, редактировать их и сохранять. Использование полос прокрутки <code>Scrollbar</code> .	
46.	Урок 46. Продвинутое возможности Python.	Структура данных с идентификацией элемента не по числовому, а по произвольному ключу: словарь. Как создать словарь, как заполнить словарь, работа с элементами словаря. Возможный пример: создадим в Python словарь, в котором индексом является название страны, а значение – названием столицы этой страны. Затем добавим диалог с пользователем: программа спрашивает название страны, в котором живет пользователь, если такая страна уже есть в словаре, то выводится сообщение о том, как называется столица этой страны, в противном случае пользователю задается вопрос о том, как называется столица его страны и полученная информация сохраняется в словарь.	
47.	Урок 47. Создание продвинутого интерфейса к программе, разработанной на предыдущем уроке	. Вопрос и ответ задаются и обрабатываются с помощью диалоговых окон, а информация о странах и столицах хранится в файле.	
48.	Урок 48. Завершение		

	работы над программой.		
49.	Урок 49. Работа с датой и временем Модуль date и datetime.	Как с помощью языка Python получить сегодняшнюю дату и текущее время? Как узнать день недели для какой-то даты? Несложные практические задания, например, А.С.Пушкин родился 6 июня 1799 года. С помощью модуля datetime определите, какой был день недели. Удобное использование словаря для получения названия дня недели по номеру дня: days = {0: 'понедельник', 1: 'вторник', 2: 'среда', 3: 'четверг', 4: 'пятница', 5: 'суббота', 6: 'воскресенье'}	
50.	Урок 50. Кейс 8. Создание приложения	<p>Календарь дней рождения моей семьи Обсуждение проекта. Общая идея приложения: в файле хранится информация, например, в таком виде: День рождения мамы,05.05.1984 День рождения папы,09.08.1983 Приложение обрабатывает эту информацию, с помощью модуля datetime и его возможностей вычисляет, сколько дней между текущей датой и днем рождения каждого члена семьи, упорядочивает данные по близости к текущей дате и выводит информацию, например, в таком виде:</p> <div style="background-color: #00b0f0; color: #ffcc00; padding: 10px; text-align: center;"> <p><u>Календарь дней рождений</u></p> <p>День рождения папы через 51 д.</p> <p>День рождения дедушки через 54 д.</p> <p>День рождения бабушки через 128 д.</p> <p>День рождения мамы через 260 д.</p> </div>	
51.	Урок 51. Завершение работы над проектом.		
52.	Урок 52. Кейс 9. Разработка и создание игры "Мемори".	Подготовка игрового поля. Использование символов Unicode в качестве картинок, например, u'\u2702' это символ с изображением ножниц. Создание сетки из кнопок с закрытым текстом с помощью упаковщика grid. Конфигурация цвета поля и размера текста. Создание функции, открывающей текст-изображение при нажатии на кнопку. Обсуждение дальнейшей реализации.	

			
53.	<p>Урок 53. Разработка и создание игры “Мемори”.. Полезные ресурсы: 1. https://younglinux.info/pygame/pygame 2. https://habr.com/ru/post/347138/ Что получит ученик по окончании модуля: • Обучающийся получит представление о некоторых дополнительных библиотеках языка Python, позволяющих разрабатывать приложения с GUI (PyQt, wxPython, Pygame). • Обучающийся познакомится с принципами работы основных элементов библиотеки Pygame и научится понимать код, использующий Pygame. • В процессе обучения</p>	<p>Использование словарей для хранения информации о том, какая кнопка нажата. Использование <code>grid_info()['row']</code> и <code>grid_info()['column']</code> для получения информации о том, в какой строке и в каком столбце находится нажатая кнопка. Модификация функции, обрабатывающей событие нажатие кнопки – как сделать так, чтобы запоминалась информация о предыдущей нажатой кнопке и обработка этой информации. Использование задержки <code>sleep</code> из модуля <code>time</code> для отображения изображений, если при двух последовательных нажатиях были открыты разные картинки.</p>	

	ученик разработает и напишет игровое приложение с использованием Pygame		
54.	Урок 54. Разработка и создание игры “Мемори”. Завершение работы над проектом.	Тестирование проекта и доработка мелочей (если изображения уже открыты, то на кнопки нельзя нажимать, если открыты все изображения, то игра завершена и т.п.). Тестирование проектов одноклассников. Добавление счетчика ходов. Работа с изображениями в Python. Замена текстовых надписей в приложении “Мемори” изображениями.	
	Модуль 4.	Искусственный интеллект (6 часа)	
55.	Урок 1. Искусственный интеллект. История.	Что такое искусственный интеллект и что такое интеллект вообще? Есть ли IQ у компьютерных программ?	
56.	Урок №2 Искусственный интеллект. История.	Как можно сравнить человеческий и компьютерный интеллекты? Алан Тьюринг и его работы. Немного про вычислительную сложность.	
57.	Урок 3 Искусственный интеллект. Идея двоичного поиска.	Обсуждение задачи угадывания числа: учитель загадал натуральное число от 1 до 100. Ученик хочет его отгадать, и задает вопросы, на которые учитель отвечает только “да” или “нет”. За какое наименьшее количество вопросов ученик сможет отгадать загаданное число? Если перебирать числа последовательно (такой способ в программировании называется линейный поиск), за 99 вопросов точно можно угадать, но наверняка можно и за меньшее количество. Обсуждение со школьниками, школьники задают вопросы, учитель фиксирует на доске вопросы и отвечает “наихудшим” возможным образом. Обсуждение того, какие вопросы были хорошими, и почему. Обсуждение идеи сокращения количества подозрительных чисел вдвое одним вопросом. Общая формулировка идеи двоичного поиска. Обсуждение того, сколько вопросов потребуется, если загадано число от 1 до N, где N – степень двойки. Обсуждение того, почему для N = 10 трех вопросов не хватит.	
58.	Урок 4. Продолжение обсуждения идеи двоичного поиска.	Напомнить про то, что такое двоичный поиск (для того, чтобы отгадать число от 1 до 100 потребовалось бы всего 7 вопросов с ответом да/нет). Дополнительно можно обсудить, что на прошлом уроке все вопросы задавались последовательно, и каждый последующий вопрос задавался после того, как был известен ответ на предыдущий. А вот хватит ли семи вопросов, если все вопросы формулируются сразу, например, на листке, и даются учителю? (Учитель сразу дает ответы на все записанные вопросы, после чего ученик должен сказать, каким было 30 Информатика 31 загаданное число). Ответ: да, семи вопросов для загаданного числа от 1 до 100 тоже хватит, например, можно спросить “какова первая цифра в двоичной записи загаданного числа, если его дополнить слева нулями до двоичного числа длины 7”, “какова первая цифра в двоичной записи загаданного числа, если его дополнить слева нулями до двоичного числа длины 7”, и т.п. Создание программы, отгадывающей загаданное число. Пример подобной программы перед вами. <code>from random import * print (“Я загадал число</code>	

		от 1 до 20. Попробуй угадать!') secret = randint(1, 20) ans = int(input()) while ans != secret: if ans > secret: print('Слишком большое число!') else: print('Маловато...') ans = int(input('Попробуй еще раз!\n')) print('Молодец, угадал!') Предложите школьникам модернизировать программу так, чтобы загадывалось число от 1 до n, где n задает пользователь. Кроме того, пусть программа подсчитывает количество заданных вопросов и после того, как число отгадано, сообщает это количество.	
59.	Урок 5. Кейс 10: приложение, угадывающее возраст.	Создание приложения на языке Python, отгадывающее возраст пользователя. Идея: допустим, человек родился 16 июля. Умножим день рождения на 2, получим 32 (16*2). Прибавим к результату 5, получим 37 (16*2+5). Умножим результат на 50, получим 1850 ((16*2+5)*50=16*100+250). К результату прибавим номер месяца рождения, получим 1857. Результат вычислений попросим сообщить нам. Для того, чтобы узнать дату рождения, достаточно вычесть из результата вычислений число 250, получится число 1607, первые две его цифры соответствуют дню, последние две – месяцу. Для удобства реализации создадим словарь, ключами в котором являются числовые номера месяцев, а значениями – названия месяцев.	
60.	Урок 6. Кейс 10: приложение, угадывающее возраст.	Пример возможной реализации приведен ниже. D = {1: 'января', 2: 'февраля', 3: 'марта', 4: 'апреля', 5: 'мая', 6: 'июня', 7: 'июля', 8: 'августа', 9: 'сентября', 10: 'октября', 11: 'ноября', 12: 'декабря'} input('Привет! Я отгадаю дату твоего рождения. Нажми enter, как будешь готов.') input('Умножь день, в который ты родился, на 2. Если ты родился 15 февраля, умножай на 2 число 15. Нажми enter, как будешь готов. ') input('К результату прибавь 5. Нажми enter, как будешь готов.') input('Полученное число умножь на 50. Нажми enter, как будешь готов.') input('К результату прибавь номер месяца, в который ты родился (число от 1 до 12). Нажми enter, как будешь готов.') ans = int(input('А теперь скажи, что у тебя получилось. ')) ans = ans - 250 print('Ты родился', ans//100, D[ans % 100]) Что получит ученик по окончании модуля: • Обучающийся познакомится с понятием искусственного интеллекта и с историческими фактами, касающимися искусственного интеллекта. • Разберется с идеей двоичного поиска. • Напишет несколько простых консольных приложений.	
Модуль 5.		Продвинутые библиотеки языка Python. Pygame (12 часов)	
61.	Урок 1. Знакомство с продвинутыми фреймворками для разработки GUI-приложений.	Обзор дополнительных библиотек, позволяющих создавать приложения с графическим интерфейсом (PyQt, wxPython, 32 Информатика 33 Pygame). Примеры приложений. Установка библиотеки Pygame. update(), quit(). События.	
62.	Урок 2. Шаблон программы на Pygame Как устроен шаблон программы на Pygame.	Подключение модулей, инициализация, главный цикл, корректное завершение работы приложения. Функции init(), set_mode(), update(), quit(). События,	
63.	Урок 3. Геометрические примитивы в Pygame	Рисуем и раскрашиваем линии, прямоугольники, многоугольники, круги, овалы.	

	Цветовые модели.		
64.	Урок 4. Кейс 11. Простая анимация в Pygame.	Создание анимации движения геометрической фигуры от левой границы главного окна вправо. Как изменить программу так, чтобы фигура не исчезала, а отражалась от стен.	
65.	Урок 5. События клавиатуры.	Модуль pygame.event. Кейс 12: создание приложения, реализующего перемещение геометрической фигуры с помощью стрелок вправо/влево и вверх/вниз.	
66.	Урок 6. События мыши.	Обработка событий нажатия кнопки мыши, отпускания кнопки мыши, движения мыши. Координаты мыши. Кейс 13: “Звездное небо”. Создание приложения, в котором при щелчке мыши по экрану на экране появляются звездочка.	
67.	Урок 7 Дополнительные поверхности в Pygame.	Surface и blit(). Простые примеры.	
68.	Работа с творческими работами	Работа с подготовкой творческих работ Выступление обучающихся со своими творческими работами.	

Рабочий план Vгода обучения

Данная программа предусматривает на реализацию программы по информатике в группе обучающихся 15-16 лет

Рабочая программа рассчитана на 34 учебных недели, 1 часа в неделю, общее количество часов — 34 часов.

Содержание программы

Содержание программы:

1. Big Data (10 часов)

Что такое большие данные и как с ними работать? Модель обработки данных MapReduce. Примеры задач, решаемых с помощью парадигмы MapReduce (задача Word Count, обработка логов рекламной системы). Стек технологий Hadoop для работы с большими данными.

2. Сетевые технологии. Интернет (10 часов)

История возникновения компьютерных сетей. MAC-адрес. IP-адрес, типы IP-адресов. Адрес сети и адрес узла. Маска подсети. WWW и Интернет – в чем отличие? URL-адреса. Протоколы передачи данных. Внутреннее устройство WWW. Запросы и ответы. Взаимодействие клиент-сервер. Установка веб-сервера. Безопасность в Интернете. Службы и сервисы Интернета.

3. Работа с внешними API сайтов (10 часов)

Тестирование API. Форматы передачи данных XML и JSON. Определение IP с помощью api.ipify.org. Структура данных словарь в языке Python. Модуль requests языка Python и http запросы.

4. Как создаются сайты (32 часа)

Основы HTML5. Гипертекст. Таблицы. Верстка простых веб-страниц. Вставка изображений. Аудио- и видео-контент на веб-странице. Дизайн веб-страницы. Каскадные таблицы стилей. Верстка страниц с помощью блоков. Формы в HTML. CSS и анимация. Основы программирования на языке JavaScript. Frontend и backend-программирование.

5. Видео (10 часов)

Форматы видеофайлов. Простой видеоредактор. Обработка

видео: нарезка, создание титров, добавление простых эффектов. Экспорт видео. Продвинутое эффекты. Работа с сервисом YouTube.

КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ Vгода обучения(9 кл)

№ п /п	Модули, кейсы, тема	Деятельность обучающихся	Дата провед
	Модуль1	Big Data(9 часов)	
1	Урок 1. Big Data: большие данные или сложные данные? Что такое Big Data, почему важно уметь с ними работать?	При-меры источников данных, для которых необходимы методы работы с большими данными, обсуждения. Характеристики больших данных, три V (volume, velocity, variety).	02.09
2	Урок 2. Большие данные в мире и в России В каких проектах и задачах возникают большие	Еще немного истории и подробностей. Основные принципы работы с большими данными. Как умение работать с большими данными помогает развитию электромобилей Tesla? Кейс 1. Подготовьте презентацию на тему “”Что такое Big Data	09.09

	данные?		
3	Урок 3. MapReduce	Модель распределенной обработки данных MapReduce. Стадии Map, Shuffle и Reduce.	16. 09
4	Урок 4. Задача Word Count Классическая задача, решаемая с помощью парадигмы	MapReduce – задача WordCount (для каждого слова, хотя бы раз встречающегося в наборе документов вычислить сколько именно раз это слово встретилось). Вспомнить, как решить подобную задачу в простом варианте (есть текстовый файл, нужно определить частоту появлений каждого слова или каждого символа) – идея сортировки подсчетом, словари в языке Python.	23. 09
5	Урок 5. MapReduce и сложная задача Word Count	Как MapReduce может помочь решить задачу Word Count в трудном варианте. Другие примеры задач, решаемых с помощью парадигмы MapReduce (например, задача обработки логов рекламной системы)	30. 09
6	Урок 6. Стек технологий Hadoop для работы с	История. Что такое Hadoop, основные компоненты. Пример запуска MapReduce-задачи на Hadoop. Кейс 2. Составьте интеллект-карту на тему “MapReduce и Word Count”.	07. 10

	большим и данными		
7	Урок 7. Реализация на Python задачи подсчета количества слов в простом варианте .	Подготовка Словари в языке Python. Простые задачи на словари.	14. 10
8-9	Урок 8. Реализация на Python задачи подсчета количества слов в простом варианте . Подготовка	Словари в языке Python. Задача подсчета количества символов в тексте. Идея сортировки подсчетом. Полезные ресурсы: 1. https://habr.com/ru/company/dca/blog/267361/ 2. https://in-scale.ru/blog/big-data 3. https://pythonworld.ru/typy-dannyx-v-python/slovari-dict-funkcii-i-metody-slovaroj.html 4. https://habr.com/ru/company/yandex/blog/332688/ 5. Марк Лутц, Изучаем Python, 4ое издание (2011 год).	21. 10
10	Урок 10 Кейс 3: “Подсчет частоты встречающихся	Что получит ученик по окончании модуля: • познакомится с понятием BigData, проблемами, связанными с обработкой больших данных и способами решения этих проблем • научится пользоваться продвинутыми возможностями языка python (словари) для решения различных задач • потренируется в практическом программировании, написав	11. 11

	слов в произведениях уроков литературы” (реализация в простом варианте, с использованием словарей).	программу подсчета частоты встречающихся слов в тексте	
	Модуль 2.	2.Сетевые технологии. Интернет (10часов)	
11	1 Что такое Интернет?	История возникновения компьютерных сетей. Для чего нужно соединять компьютеры в сети? Как соединить в сеть несколько компьютеров (по кругу? каждый с каждым? все компьютеры к одному серверу? недостатки этих идей). Сетевая карта. Принципы работы сетевого хаба и свитча. Интернет= объединение нескольких сетей.	18.11
12	2 Как «подписывают» устройства в сети?	MAC-адрес: уникальность, где выдается, почему нельзя общаться по MAC-адресам. IP-адрес, как способ «пронумеровать» устройства. Сколько памяти выделяется на один IP-адрес? Сколько адресов можно в принципе записать? IPv4 и IPv6 - зачем нужно увеличивать длину IP-адреса?	25.11
13	3 Структура IP-адреса	Типы IP-адресов: публичные/частные; статические/динамические, etc. Белый IP и адреса внутри сети. Как устройства ищут друг друга по IP-адресам? Адрес сети и адрес узла. Маска подсети. Два способа записи маски: префиксный и десятичный. <i>Кейс 4. Создайте интеллект-карту на тему “Структура IP-адреса”</i>	02.12
14	4. Как путешест	Передача информации из сети через основной шлюз. ipconfig Трассировка маршрута с помощью tracert. Передача пакетов с	09.12

	вует информация?	помощью ping. Для чего нужны DNS-серверы. ipconfig с параметрами.	
15	5 URI и URL-адреса	Домены разного уровня. Доменные регистраторы (например, reg.ru). В чем отличие Интернета и WWW? WWW - всемирная паутина, страницы, связанные гиперссылками, Интернет - объединение сетей, протоколы и т.п. <i>Кейс 5. Подготовьте презентацию "Интернет и WWW. История: факты и выдумки."</i>	16.12
16	6 Как работает WWW?	Браузер. Веб-сервер. Взаимодействие браузера и веб-сервера: запросы (requests) и ответы (responses). Языки для веб-программирования: php, python, java, ruby. Взаимодействие клиент-сервер. Протокол HTTP. Коды возврата. <i>Кейс 6. Составьте интеллект-карту на тему: "Как работает WWW?"</i>	23.12
17	7 Методы передачи данных	Как устроены и чем отличаются GET и POST запросы. Как установить себе веб-сервер?	13.01
18	8 Безопасность передачи данных	Безопасность в интернете. SSL и HTTPS (SSLoverHTTP). Авторизация и cookies. Сетевые угрозы. Мошенничество. Правила личной безопасности в Интернете. <i>Кейс 7. Составьте интеллект-карту на тему: "Данные в интернете. Передача и безопасность."</i>	20
19	9 Службы Интернета	Электронная почта. Обмен файлами (FTP). Форумы. Общение в реальном времени. Интернет-магазины. Электронные платёжные системы. Полезные ресурсы: 1. https://habr.com/ru/post/253803/ 2. https://fb.ru/article/367974/post-get-chem-otlichayutsya-zaprosyi-drug-ot-druga	27.03
20	10 Службы Интернета	Что получит ученик по окончании модуля: • Разберется с тем, чем отличаются Интернет и WWW. • Познакомится с устройством сетей и принципами передачи информации по сетям. • Узнает, что такое IP-адреса, какие они бывают и для чего	03.02

		нужны. • Познакомится с тем, как следят за безопасностью в Интернете.	
	Модуль 3.	3.Работа с внешними API сайтов (10 часов)	
2 1	1 Что такое API?	Тестирование API. Анализ запроса с помощью сервиса aritester.com, например, на примере запросов https://ru.wikipedia.org/wiki/Python , https://ru.wikipedia.org/wiki/Python , https://ru.wikipedia.org/wiki/python . Коды ответов сервера. Обсудить, каковы коды ответа для каждого из приведенных выше запросов. Ошибки клиента и ошибки сервера. Что записано в строке https://yandex.ru/search/?text=Python&lr=213 после знака вопроса? Параметры запросов.	10. 02
2 2	2 Форматы передачи данных	Формат данных XML и формат данных JSON. Аналогия с правильными скобочными последовательностями.	17
2 3	3 Сервис ipify.org	Сервис ipify.org. Определение IP с помощью api.ipify.org. Определение геолокации по IP с помощью geo.ipify.org. Регистрация на сервисе для того, чтобы иметь возможность получать данные геолокации.	24. 02
2 4	4 Структура данных словарь	Для того чтобы удобно обрабатывать запросы, нужно знать, как работать со словарями в Python. Ключи и значения. Создание словаря в Python. Добавление и удаление элемента. Перебор элементов словаря.	02. 03
2 5	5 Модуль requests языка Python и http-запросы	Установка модуля requests. Запись запроса с помощью requests. Функция get (). Создание консольного приложения, отправляющего запрос на сервис https://api.ipify.org и получающего IP-адрес вашего компьютера.	09. 03
2 6	6 Кейс 8: Созданы	Создание консольного приложения, по IP-адресу компьютера определяющее страну, область, город, почтовый индекс, временную зону, широту и долготу. Запись запроса с параметрами в	16

	е приложе ния «Опреде ление данных геолокац ии по IP»	библиотеке requests. Начало работы над программой может быть, например, таким: import requests url_1 = "https://api.ipify.org" answer_1 = requests.get(url_1) ip_address = answer_1.text print(ip_address)url_2 = "https://geo.ipify.org/api/v1" params = {"apiKey": "тут должен быть ваш apiKey, полученный при регистрации на сервисе", "ipAddress": ip_address} answer_2 = requests.get(url_2, params = params) print(answer_2.json())	
2 7	7Кейс 8: Создани е приложе ния «Опреде ление данных геолокац ии по IP»	Превращение консольного приложения, созданного на прошлом уроке, в графическое.	23. 03
2 8	8Кейс 8: Создани е приложе ния «Опреде ление данных геолокац ии по IP»	Превращение консольного приложения, созданного на прошлом уроке, в графическое.	
2 9	9Кейс 8: Создани	Дополним созданное приложение возможностью сохранить найденные данные в файл.	

	е приложения «Определение данных геолокации по IP».		
30	10Создание приложения «Определение данных геолокации по IP».	<p>Что получит ученик по окончании модуля:</p> <ul style="list-style-type: none"> • Познакомится с тем, что такое внешние API сайтов, и научится с ними работать с помощью соответствующих библиотек языка Python. 	
	Модуль 4.	4.Как создаются сайты (32 часа)	
31	1 Как пишут веб-сайты?	<p>Содержание, оформление, работа с данными. Первые примеры на языке HTML. Средства разработки: онлайн-инструменты и IDE. Инструменты разработчика в Google Chrome.</p> <p>Урок 29. Структура HTML-документа. Структура html-документа. Основные теги. Отличия HTML5 и HTML4.</p>	
32	2 Работа с текстом и гиперссылками	Текстовые веб-страницы. Простейшая веб-страница. Заголовки. Абзацы. Специальные символы. Списки.	
33	3Кейс 9: Создани		

	е простой веб- страниц ы.		
3 4	4Графика а на веб- страница х Вставка изображе ний.	Форматы графических файлов и их особенности, важные для веба.	

Учебно-методическое обеспечение

1. Примерная основная образовательная программа образовательного учреждения. Основная школа. – М.: Просвещение, 2011.
2. Бешенков С.А. Примерные программы по информатике для основной и старшей школы. – М.: БИНОМ. Лаборатория знаний, 2011.
3. Белова Г.В. Программирование в среде ЛОГО. Первые шаги. – М.: Солон, 2007
4. Великович Л., Цветкова М. Программирование для начинающих. – М.: Бином, 2007
5. Полат Е.С. Новые педагогические и информационные технологии в системе образования. – М. Академия. – 2006.
6. Патаракин Е.П. Учимся готовить в среде Скретч - Версия 2.0
7. <http://scratch.ucoz.net>
8. <http://scratch.mit.edu>- официальный сайт проекта Scratch
9. Сорокина В.В. Психологическое неблагополучие детей в начальной школе. – М.: Генезис, 2005

Материально-техническое обеспечение

Методичка с разработками по каждому разделу, С.М. Шпынева

CD - диск с файлами-проектами для лабораторных и практических работ, С.М. Шпынева

Кабинет, оборудованный компьютерами, столами, стульями, общим освещением, интерактивной доской, стендами, мультимедийным оборудованием (проектор, экран, аудиоустройства)

- Материалы и инструменты: таблицы, компакт-диски с обучающими и информационными программами по основным темам программы;
ОС Windows и Linux; учебные компьютерные программы Gimp, Inkscape; презентации

Полезные ресурсы:

1. https://www.movavi.ru/videoeditor/?admitad_uid=c992c13512ce65caba8700737f1ca220&tagtag_uid=c992c13512ce65caba8700737f1ca220
2. <https://lifehacker.ru/besplatnye-videoredactory/>
3. <https://ru.wikipedia.org/wiki/YouTube>
4. <https://iklife.ru/youtube/kak-sozdat-kanal.html>
5. <https://videosmile.ru/lessons/read/kak-pravilno-obrabatyivat-video-luchshie-programmyi-dlya-obrabotki-video.html>.

В данном документе пронумеровано: пятьдесят (.....) страниц

Директор МКОУ СОШ №12: _____ /Дутова И.И./