

**Общеобразовательная автономная
некоммерческая организация
«Школа и детский сад «МИР»**

**УТВЕРЖДАЮ
Директор**

**СОГЛАСОВАНО
Педагогический совет
Протокол № 08
от «30» июня 2022 г.**



**РАБОЧАЯ ПРОГРАММА ПО КУРСУ
ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ
ИТ Английский + Информатика»**
возраст обучающихся: от 12 до 16 лет, срок реализации: 1 год

**Авторы-составители:
Юхновский М. Молодцов А.**

п. Доброград – 2022г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Программа данного курса посвящена обучению школьников различным аспектам программирования на современном языке Python на английском языке. Занятия курса направлены на развитие языковых навыков, мышления, логики, творческого потенциала учеников. Программа ориентирована на использование получаемых знаний для разработки реальных проектов. Курс содержит большое количество творческих заданий (именуемых Кейсами). Рабочая программа курсу внеурочной деятельности «ИТ Английский + Информатика» составлена в соответствии с:

- Федеральным законом от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказом Министерства Просвещения РФ от 11.02.2022 № 69 «О внесении изменений в Порядок организации и осуществления образовательной деятельности по основным общеобразовательным программам – образовательным программам начального общего, основного общего и среднего общего образования»;
- ФГОС начального общего образования, утвержденным приказом Минпросвещения от 31.05.2021 № 286 и ФГОС основного общего образования, утвержденным приказом Минпросвещения от 31.05.2021 № 287;
- уставом ОАНО «Школа и детский сад «МИР»
- локальными нормативными актами ОАНО «Школа и детский сад «МИР»
- основной образовательной программой начального общего и основного общего образования ОАНО «Школа и детский сад «МИР», утвержденной приказом от 01.07.2022 №ОД-112.

Цель и задачи обучения

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

Для достижения поставленной цели необходимо решение следующих задач:

- создание условий для развития интеллектуальных и творческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
- формирование информационной и алгоритмической культуры;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей;
- формирование умения чтения и понимания текстов технической направленности на английском языке;
- овладение лексикой английского языка, употребляемой в синтаксисе языков программирования;
- 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.);

Общая характеристика учебного предмета

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

Она включает в себя пять блоков:

- Основы языка Python
- Создание приложений с помощью tkinter

- Криптография
- Искусственный интеллект
- Продвинутое библиотеки языка Python. Pygame

Важная задача изучения этих содержательных линий в курсе – добиться систематических знаний, необходимых для самостоятельного решения задач, в том числе и тех, которые в самом курсе не рассматривались, и сформировать навык работы с иноязычным текстом технической направленности. На протяжении всего курса учащиеся изучают различные аспекты программирования на современном языке Python используя оригинальные материалы на английском языке.

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 интеллектуального потенциала, познавательных возможностей. Обучение ориентировано на самостоятельный поиск результата, самостоятельное добывание знаний, творческое, интеллектуально-познавательное усвоение учениками заданного предметного материала.

- Личностно-ориентированные технологии обучения, способ организации обучения, в процессе которого обеспечивается всемерный учет возможностей и способностей обучаемых и создаются необходимые условия для развития их индивидуальных способностей.

- Информационно-коммуникационные технологии.
- Технология коллективных методов обучения (работа в парах постоянного и сменного состава)

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

Место курса в учебном плане

Данная рабочая программа рассчитана на 34 учебные недели, 2 часа в неделю, общее количество часов — 68. Рабочая программа может реализовываться с использованием электронного обучения (ЭО) и дистанционных образовательных технологий (ДОТ)

Личностные, метапредметные и предметные результаты освоения учебного предмета

Личностными результатами, формируемыми при изучении предмета «ИТ Английский + Информатика», являются:

- формирование ответственного отношения к учению, готовности и способности, обучающихся к саморазвитию и самообразованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индивидуальной траектории образования на базе ориентировки в

мире профессий и профессиональных предпочтений, с учётом устойчивых познавательных интересов;

- формирование целостного мировоззрения, соответствующего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое,

духовное многообразие современного мира;

- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творческой и других видов деятельности.

Метапредметные результаты изучения предмета «ИТ Английский + Информатика»:

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

Предметные результаты изучения предмета «ИТ Английский + Информатика»:

- формирование представления об основных изучаемых понятиях курса на русском и английском языках;
- формирование понимания места английского языка как основного языка в сфере программирования
- формирование информационной и алгоритмической культуры; формирование представления о компьютере как универсальном устройстве обработки информации; развитие основных навыков и умений использования компьютерных устройств;
- развитие алгоритмического мышления, необходимого для профессиональной деятельности в современном обществе; развитие умений составить и записать алгоритм для решения конкретной задачи;
- формирование умений формализации и структурирования информации, умения выбирать способ представления данных в соответствии с поставленной задачей, с использованием соответствующих программных средств обработки данных;
- знакомство с базовыми конструкциями языка Python; формирование умения придумывать алгоритмы и их реализовывать на языке Python;
- формирование умений работы с дополнительными библиотеками языка Python (tkinter, ...)
- формирование умений поиска в Интернете на английском языке и работы с инструкциями на английском языке;

- формирования представления о том, что такое криптография, каковы были классические алгоритмы шифрования данных в древности и в чем заключаются их недостатки, каковы современные методы шифрования;
- формирование умения создавать реальные приложения с помощью языка Python, формирование умения применять накопленные знания для решения практических задач;
- использование готовых прикладных компьютерных программ по выбранной специализации;
- развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимости справочных материалов, компьютера;
- формирование навыков и умений безопасного и целесообразного поведения при работе с компьютерными программами и в Интернете, умения соблюдать нормы международной информационной этики и права

Содержание курса

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.

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 “пинг-понг”, “сапер”.

Криптография (26 часов)

История криптографии. Знаменитые шифры (атбаш, сцигала, шифр Цезаря, квадрат Полибия, решетка Кардано). Создание криптографического приложения с помощью tkinter. Шифры, которые 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”.

Искусственный интеллект (4 часа)

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

Продвинутое библиотеки языка Python. Pygame (10 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

ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ.

№	Тема урока	Кол-во часов	Виды деятельности
1	Модуль 1. Основы языка Python. 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, 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	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.
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.
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 ==, !=, <=, >, >=,
5	Основы языка Python. Условная инструкция в Python,		Одновременно нужно проверить истинность нескольких условий? Или то, что, верно хотя бы одно условие из нескольких? На помощь приходят логические операции or и and.

	elif, логические операции.		
6	Цикл for в Python.		Использование цикла для перебора объектов (конструкция for elem in object, где object – строка, кортеж, иной итерируемый объект). Использование цикла для выполнения заданного количества операций: три способа записи for i in
7	Цикл while		Цикл с предусловием. Блок-схема. Бесконечный цикл. Оператор break для выхода из цикла. Переменная-счетчик для подсчета количества операций в цикле
8	Строки		Нумерация символов строк. Отрицательная нумерация. Функция длины строки. Перебор символов строки с помощью цикла for (два способа: for smb in str 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
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. Урок
10	Функции. Встроенные функции в Python		Функции как инструмент многократного использования одного и того же кода. Локальные и глобальные переменные. Самостоятельное создание простых функций, например, функции, 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	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.)
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
13	Модуль 2. Создание приложений с помощью tkinter. Знакомство с модулем		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).
14	Что такое виджеты, конфигурация виджетов		Некоторые виджеты: кнопка, текстовое поле, поле ввода, метка. Конфигурация виджетов, методы config и configure. Получение информации о состоянии виджет
15	События в tkinter		Как научить виджет реагировать на события – различные способы. Методы bind и unbind.
16	Кейс 2. Создайте интеллект-карту на тему “События в tkinter”. 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
17	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 нажатой кнопки.
18	Создание простых приложений.		Упаковщики виджетов: pack(), grid(), place(). Создание простого калькулятора из нескольких кнопок и текстового поля для вывода результата вычислений
19	Продвинутое рисование в tkinter.		Рисование на холсте canvas в tkinter. Создание различных геометрических фиг
20	Движение нарисованных объектов.		Движение нарисованных объектов с помощью клавиатуры.
21	Кейс 3. Создание игрового приложения “Пинг-понг”		“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
22	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
23	Завершение работы над приложением “пинг-понг”		Работа в парах: ребята делятся на пары программист-тестер, и тестируют созданные приложения
24	Общая идея игры сапер.		Двумерные списки, вложенные циклы. Как расположить виджеты Label в виде прямоугольника. Раскрасим минное поле в шахматном порядке
25	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?
26	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)
27	Создаем “мозг” игры.		Пишем “мозг” игры сапер. Что происходит в тот момент, когда пользователь нажимает на клетку, описываем реакцию виджета на событие
28	Завершение работы над приложением “Сапер”		Доделываем приложение, тестируем, убираем недостатки.
29	Модуль 3. Криптография История криптографии. Старинные шифры		Что такое криптография и чем она занимается? Как шифровали сообщения в древности? Старинные шифры атбаш и сцитала (шифр Древней Спарты).
30	История криптографии. Старинные шифры		Чем отличается стеганография от криптографии. Диск и линейка Энея. Шифр Цезаря
31	История криптографии. Старинные шифры		Шифрование с использованием таблиц. Квадрат Полибия. Решетка Кардано.
32	Игровое занятие. Разгадываем шифры		Создание и расшифровка “таинственных посланий” с использованием вспомогательных материалов – карандаша и полоски бумаги для шифра сцитала, диска для шифра Цезаря и т.д
33	Кейс 6. Создание криптографических приложений с помощью tkinter		Диалоговые окна simpledialog и messagebox в tkinter. 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)
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	Завершение работы над приложением		Завершаем работу, устраняем недостатки
36	Модификация и развитие шифра Цезаря.		Шифры, которые практически невозможно разгадать. Шифр Виженера
37	Современные алгоритмы шифрования		Современные алгоритмы шифрования и основные принципы их работы. Кто и зачем придумал RSA? Идея открытого и закрытого ключа.
38	Современные алгоритмы шифрования		Еще раз обсуждение того, что такое открытый и закрытые ключи. Идея односторонней функции с лазейкой (трудновычислимой 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	Игровое соревновательное занятие “Взламываем шифр с открытым ключом”.		Придумывание школьниками своих шифров для одноклассников и последующее разгадывание сообщений
40	Что такое электронная подпись?		Протокол аутентификации сообщений. Протокол электронно-цифровой подписи. Алгоритмы проверки электронной подписи. Алгоритм генерации электронной подписи.
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	Работа с файлами в		Чтение данных из файла. Методы readline(), readlines() и read(). Удаление концевых символов строки с помощью метода rstrip(). Вывод данных

			файл. Создание программы, считывающей данные из файла посимвольно
43	Кейс 7. Разработка приложения, сохраняющего данные в файле		Подготовка к созданию приложения, позволяющего открывать текстовые файлы в текстовом поле в tkinter или создавать новые, а также редактировать их и сохранять. Многострочное текстовое поле Text в tkinter. Конфигурация виджета Text (размеры, шрифт, цвет).
44	Разработка приложения, сохраняющего данные в файле		Методы insert(), get() и delete() виджета Text. Создание простого приложения, содержащего многострочное текстовое поле и две кнопки, позволяющие добавлять текст и удалять его
45	Разработка приложения, сохраняющего данные в файле.		Модуль filedialog в tkinter (диалоговые окна открытия и сохранения файлов). Создание приложения, позволяющего открывать текстовые файлы в текстовом поле в tkinter или создавать новые, редактировать их и сохранять. Использование полос прокрутки Scrollbar
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	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	Завершение работы над программой		Устраняем неполадки
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.
50	Кейс 8. Создание приложения		Обсуждение проекта.

	Календарь дней рождения моей семьи		
51	Завершение работы над проектом.		Доработка проекта
52	Кейс 9. Разработка и создание игры “Мемори”.		Подготовка игрового поля. Использование символов Unicode в качестве картинок, например, <code>u'\u2702'</code> это символ с изображением ножниц. Создание сетки из кнопок с закрытым текстом с помощью упаковщика <code>grid</code> . Конфигурация цвета поля и размера текста. Создание функции, открывающей текст-изображение при нажатии на кнопку. Обсуждение дальнейшей реализации.
53	Разработка и создание игры “Мемори”.		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.
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
55	Модуль 4. Искусственный интеллект Урок 55. Искусственный интеллект. История.		Что такое искусственный интеллект и что такое интеллект вообще? Есть ли IQ у компьютерных программ? Как можно сравнить человеческий и компьютерный интеллекты? Алан Тьюринг и его работы. Немного про вычислительную сложность
56	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?
57	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).
58	Кейс 10: приложение, угадывающее возраст.		Создание приложения на языке Python, угадывающее возраст пользователя. Идея: допустим, человек родился 16 июля. Умножим день рождения на 2, получим 32 ($16 \cdot 2$). Прибавим к результату 5, получим 37 ($16 \cdot 2 + 5$). Умножим результат на 50, получим 1850 ($(16 \cdot 2 + 5) \cdot 50 = 16 \cdot 100 + 250$). К результату прибавим номер месяца рождения, получим

			1857. Результат вычислений попросим сообщить нам
59	Модуль 5. Продвинутое библиотеки языка Python. Pygame. Урок 59. Знакомство с продвинутыми фреймворками для разработки GUI-приложений		Обзор дополнительных библиотек, позволяющих создавать приложения с графическим интерфейсом
60	Шаблон программы на		Как устроен шаблон программы на Pygame. Подключение модулей, инициализация, главный цикл, корректное завершение работы приложения. Функции init(), set_mode(), update(), quit(). События.
61	Геометрические примитивы в Pygam		Цветовые модели. Рисуем и раскрашиваем линии, прямоугольники, многоугольники, круги, овалы.
62	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.
63	События клавиатуры. Кейс 12: создание приложения, реализующего перемещение геометрической фигуры с помощью стрелок вправо/влево и вверх/вниз.		Модуль pygame.event.
64	События мыши.		Обработка событий нажатия кнопки мыши, отпускания кнопки мыши, движения мыши. Координаты мыши
65	Дополнительные поверхности в Pygame		Surface и blit(). Простые примеры.
66	Работа в команде. Создание приложения "Поймай звезду"		Создание игры, в которой в произвольном месте экрана появляются звездочки, падающие вниз. При щелчке мыши по звездочке она исчезает, а в соответствующем текстовом поле идет подсчет пойманных звезд. Классы и объекты. Реализация появления звезд. Ребята разбиваются на группы, распределяют задачи внутри группы.
67	Продолжение работы над игрой. Текст в		Реализация реакции звезды на событие щелчка мыши. Работа с текстом в Pygame. Модуль
68	Завершение работы над проектом. Музыка в Pygame		Добавление фоновой музыки и звуковых эффектов в игру. Модули pygame.mixer и pygame.mixer.music. Представление проекта.
69			

Учебно-методическое обеспечение образовательного процесса

1. К. Вордерман и др. Программирование на Python: Иллюстрированное руководство для детей. Издательство: Манн, Иванов и Фербер, 2018 г.
2. Программирование для детей на языке Python. Издательство: АСТ, 2017 г.
3. Д. Бриггс. Python для детей: Самоучитель по программированию. Издательство: Манн, Иванов и Фербер, 2018 г.
4. Б. Пэйн. Python для детей и родителей. Издательство: Эксмо, 2017 г.
5. П. Томашевский. Привет, Python! Моя первая книга по программированию. Издательство: Наука и Техника, 2018 г.

[h](#)

[h](#)

[1](#)1.Наталья Свешникова: Английский язык. Компьютерные системы, информационные

[p](#)

[e](#)

[k](#)

[n](#)

[d](#)

[n](#)

[d](#)

[e](#)

[n](#)

[d](#)

[e](#)

[n](#)

[y](#)

[nom](#)

[p](#)

[o](#)

[г](#)

[p](#)

[a](#)

[м](#)

[м](#)

[и](#)

[p](#)

[о](#)

[в](#)

[а](#)

[н](#)

[и](#)

[е](#)

[h](#)

[t](#)

[t](#)

[p](#)

[s](#)

[f](#)

[i](#)

[c](#)