Информатика

**8класс**



Общеобразовательная общеразвивающая программа технической направленности

*Программа школьного курса*

*«Информатика» для 8 класса*

Длительность 34 ак.ч Для педагогов и учащихся

ФГАУ «Фонд новых форм развития образования»

Москва, 2020

УДК ББК

**Программа школьного курса «Информатика» для 8 класса**

Авторы:

Целевая аудитория: Срок реализации:

# Оглавление

I.

Пояснительная записка 4 II.

Общая характеристика учебного предмета 5 III.

Место курса в учебном плане 6 IV.

Личностные, метапредметные и предметные результаты освоение конкретного учебного предмета, курса 7

V.

Содержание курса 10 VI.

Поурочное планирование 11 VII.

Планируемые результаты обучения 34

ISBN

(с) ФНФРО 2020

В пособии использованы материалы из открытых источников сети Интернет. Поскольку источни- ки, размещающие у себя информацию, далеко не всегда являются обладателями авторских прав, просим авторов использованных нами материа- лов откликнуться, и мы разместим указание на их авторство.

Сборник предназначен исключительно для не- коммерческого использования.

# Пояснительная записка

Программа данного курса посвящена обучению школьников различным аспектам программирования на современном язы- ке Python. Занятия курса направлены на развитие мышления, логики, творческого потенциала учеников. Программа ори- ентирована на использование получаемых знаний для разра- ботки реальных проектов. Курс содержит большое количество творческих заданий (именуемых Кейсами).

## Цель и задачи обучения

Целью изучения предмета «Информатика» является получе- ние теоретических и практических знаний, умений и навыков в области современной информатики; формирование целост- ного мировоззрения, соответствующего современному уров- ню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие со- временного мира.

Для достижения поставленной цели необходимо решение следующих задач:

* создание условий для развития интеллектуальных и твор- ческих способностей учащихся, необходимых для успешной социализации и самореализации личности;
* формирование информационной и алгоритмической культу- ры;
* 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 дан- ных в соответствии с поставленной задачей;
* овладение важнейшими общеучебными умениями и уни- версальными учебными действиями (формулировать цели

деятельности, планировать ее, находить и обрабатывать не- обходимую информацию из различных источников, включая Интернет и др.);

# Общая характеристика учебного предмета

Программа по предмету «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
* Продвинутые библиотеки языка Python. Pygame

Важная задача изучения этих содержательных линий в курсе – добиться систематических знаний, необходимых для самосто- ятельного решения задач, в том числе и тех, которые в самом курсе не рассматривались. На протяжении всего курса учащи- еся изучают различные аспекты программирования на совре- менном языке Python.

## Технологии, используемые в образовательном процессе:

* Технологии традиционного обучения для освоения миниму- ма содержания образования в соответствии с требованиями стандартов; технологии, построенные на основе объясни- тельно-иллюстративного способа обучения. В основе – ин- формирование, просвещение обучающихся и организация их репродуктивных действий с целью выработки у школьни- ков общеучебных умений и навыков.
* Технологии компьютерных практикумов.
* Технологии реализации межпредметных связей в образова- тельном процессе.
* Технологии дифференцированного обучения для освоения учебного материала обучающимися, различающимися по уровню обучаемости, повышения познавательного интереса.
* Технология проблемного обучения с целью развития твор- ческих способностей обучающихся, их интеллектуально- го потенциала, познавательных возможностей. Обучение ориентировано на самостоятельный поиск результата, са- мостоятельное добывание знаний, творческое, интеллекту- ально-познавательное усвоение учениками заданного пред- метного 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 индивиду- альных способностей.
* Информационно-коммуникационные технологии.
* Технология коллективных методов обучения (работа в парах постоянного и сменного состава)

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

# Место курса в учебном плане

Данная программа предусматривает на реализацию програм- мы по информатике в 8 классе 34 часов. Рабочая программа рассчитана на 34 учебные недели, 1 часа в неделю, общее коли- чество часов — 34. Рабочая программа может реализовываться с использованием электронного обучения (ЭО) и дистанцион- ных образовательных технологий (ДОТ).

# Личностные, метапредметные и предмет- ные результаты освоения учебного предмета

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

* формирование ответственного отношения к учению, готов- ности и способности обучающихся к саморазвитию и само- образованию на основе мотивации к обучению и познанию, осознанному выбору и построению дальнейшей индиви- дуальной траектории образования на базе ориентировки в мире профессий и профессиональных предпочтений, с учё- том устойчивых познавательных интересов;
* формирование целостного мировоззрения, соответствую- щего современному уровню развития науки и общественной практики, учитывающего социальное, культурное, языковое, духовное многообразие современного мира;
* формирование коммуникативной компетентности в обще- нии и сотрудничестве со сверстниками, детьми старшего и младшего возраста, взрослыми в процессе образовательной, общественно полезной, учебно-исследовательской, творче- ской и других видов деятельности.

## Метапредметные результаты изучения предмета «Информа- тика»:

* 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 це- лей, в том числе альтернативные, осознанно выбирать наи- более эффективные способы решения учебных и познава- тельных задач;
* умение соотносить свои действия с планируемыми результа- тами, осуществлять контроль своей деятельности в процес-

се достижения результата, определять способы действий в рамках предложенных условий и требований, корректиро- вать свои действия в соответствии с изменяющейся ситуа- цией;

* умение оценивать правильность выполнения учебной зада- чи, собственные возможности её решения;
* владение основами самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;
* умение определять понятия, создавать обобщения, уста- навливать аналогии, классифицировать, самостоятельно вы- бирать основания и критерии для классификации, устанав- ливать 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 познава- тельных задач;
* умение организовывать учебное сотрудничество и совмест- ную деятельность с учителем и сверстниками; работать ин- дивидуально и в группе: находить общее решение и разре- шать конфликты на основе согласования позиций и учёта интересов; формулировать, аргументировать и отстаивать своё мнение;
* формирование и развитие компетентности в области ис- пользования информационно-коммуникационных техноло- гий.

## Предметные результаты изучения предмета «Информатика»:

* формирование представления об основных изучаемых по- нятиях курса;
* формирование информационной и алгоритмической куль- туры; формирование представления о компьютере как уни- версальном 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 дан- ных в соответствии с поставленной задачей, с использова- нием соответствующих программных средств обработки данных;
* знакомство с базовыми конструкциями языка Python; фор- мирование умения придумывать алгоритмы и их реализовы- вать на языке Python;
* формирование умений работы с дополнительными библио- теками языка Python (tkinter, pygame, etc);
* формирования представления о том, что такое криптогра- фия, каковы были классические алгоритмы шифрования дан- ных 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 про- грамм по выбранной специализации;
* развитие умений применять изученные понятия, результаты, методы для решения задач практического характера и задач из смежных дисциплин с использованием при необходимо- сти справочных материалов, компьютера;
* формирование навыков и умений безопасного и целесоо- бразного поведения при работе с компьютерными програм- мами и в Интернете, умения соблюдать нормы информаци- онной этики и права.

# Содержание курса

## Основы языка Python (6 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 (8 часов)

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

## Криптография (13 часов)

История криптографии. Знаменитые шифры (атбаш, сцитала, 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. Продвинутые возможности Python: слова- ри. Дополнительные библиотеки языка Python для работы с датами и временем. Разработка игрового приложения “Мемо- ри”.

## Искусственный интеллект (2 часа)

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

**Продвинутые библиотеки языка Python. Pygame (5 часов)** Обзор дополнительных библиотек для работы с графическим интерфейсом. Библиотека Pygame. Шаблон программы. Геоме- трические примитивы в Pygame. Простая анимация в Pygame.

События клавиатуры. События мыши. Дополнительные по- верхности. Работа с текстом. Музыка.

# Поурочное планирование

## Модуль 1. Основы языка Python.

**Урок 1. Вводное занятие.**

Техника безопасности. Знакомство с программой курса. Крат- кая 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 интерпретируе- мых языков программирования.

Демонстрация установки интерпретатора Python. Используем Python как калькулятор: интерактивный режим работы с ин- терпретатором Python, вычисляем сумму чисел, вычисляем 2 в степени 100. Запуск встроенной среды разработки IDLE, рабо- та в командной строке Shell – аналог интерактивного режима. Тест по технике безопасности и простым фактам о языке Python.

## 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: считы- ваем число и выводим его же. Простые операции со строками, конкатенация строк. Создание простой программы, спрашива- ющей имя пользователя и затем приветствующей его.

Создание простой программы, спрашивающей у пользователя имя, количество лет, а затем выводящей имя столько раз, сколь- ко пользователю лет.

**Урок 3. Основы языка Python. Операции с числами и строками.** Типы данных int и str. Преобразование типов с помощью опе- раторов 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 пе- ревода строки ‘\n’. Тест на понимание работы функций print, input, str, int.

Простые программы на взаимодействие с системой. Два воз- можных 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)

**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 реали- зовать 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!’)

## 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 опе- ратор elif. Показать несколько простых примеров с использо- ванием elif и без использования elif. Создание простых про- грамм, где удобно использовать elif, например,

## Программа 2. Погода

weather = input(‘Какая сегодня погода? дождь/снег/солнце’) if weather == ‘дождь’:

print(‘Захватите зонтик!’) elif weather == ‘снег’:

print(‘Не забудьте варежки!’) else:

print(‘Ура! Берем солнечные очки!’)

Что такое модули и как их подключать? Модуль random и не- которые его функции. Примеры использования модуля random.

## Программа 3. Чье число больше?

import random

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.

## Урок 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 ко- личества операций: три способа записи for i in range(n); for i in range(a, b); for i in range(a, b, d)). Простые программы, например:

* Как напечатать слово привет столько раз, какое число вве- дет 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 крат- ные трем, и т.п.
* Пользователь задает число n, программа выводит n строк, в первой строке одна звездочка, во второй две и т.д.
1. . Бриггс. Python для детей: Самоучитель по программиро- ванию. Издательство: Манн, Иванов и Фербер, 2018 г.
2. Б. Пэйн. Python для детей и родителей. Издательство: Экс- мо, 2017 г.
3. П. Томашевский. Привет, Python! Моя первая книга по про- граммированию. Издательство: Наука и Техника, 2018 г.
4. https://pythontutor.ru/
5. https://ru.wikiversity.org/wiki/%D0%9A%D1%83%D1%80% D1%81\_%D0%BF%D0%BE\_%D0%B1%D0%B8%D0%B1% D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA% D0%B5\_Tkinter\_%D1%8F%D0%B7%D1%8B%D0%BA%D0

%B0\_Python

## Что получит ученик по окончании модуля:

* Обучающийся познакомится с основами программирования на языке Python.
* Изучит основные конструкции языка Python (условная ин- струкция, циклы, функции, списки, строки) на практических Напишет несложное консольное приложение.

## Модуль 2. Создание приложений с помощью tkinter.

## Урок 7. События в tkinter.

Как научить виджет реагировать на события – различные спо- собы. Методы bind и unbind.

## Кейс 2. Создайте интеллект-карту на тему “События в tkinter”. Урок 8. Создание простых приложений.

Продолжаем знакомиться с tkinter на простых примерах. Создание приложения, состоящего из кнопки, подсчитываю- щей количество нажатий, и исчезающей после определенного количества нажатий. Создание простого приложения, состоя- щего 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 за- крывается.

## Урок 9. Создание простых приложений.

Приложение “Радуга”: простое приложение, состоящее из кно- пок, окрашенных в основные цвета радуги и текстового поля. При нажатии на кнопку, в текстовом поле появляется название цвета нажатой кнопки.

## Урок 10. Создание простых приложений.

Упаковщики виджетов: pack(), grid(), place(). Создание простого калькулятора из нескольких кнопок и текстового поля для вы- вода результата вычислений.

## Урок 11. Продвинутое рисование в tkinter.

Рисование на холсте canvas в tkinter. Создание различных гео- метрических фигур.

## Урок 12. Движение нарисованных объектов.

Движение нарисованных объектов с помощью клавиатуры.

**Урок 13. 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.

**Урок 14. Завершение работы над приложением “пинг-понг”.** Работа в парах: ребята делятся на пары программист-тестер, и тестируют созданные приложения.

## Полезные ресурсы:

1. К. Вордерман и др. Программирование на Python: Иллю- стрированное руководство для детей. Издательство: Манн, Иванов и Фербер, 2018 г.
2. Программирование для детей на языке Python. Издатель- ство: АСТ, 2017 г.
3. Д. Бриггс. Python для детей: Самоучитель по программиро- ванию. Издательство: Манн, Иванов и Фербер, 2018 г.
4. Б. Пэйн. Python для детей и родителей. Издательство: Экс- мо, 2017 г.
5. П. Томашевский. Привет, Python! Моя первая книга по про- граммированию. Издательство: Наука и Техника, 2018 г.
6. https://pythontutor.ru/
7. https://ru.wikiversity.org/wiki/%D0%9A%D1%83%D1%80% D1%81\_%D0%BF%D0%BE\_%D0%B1%D0%B8%D0%B1% D0%BB%D0%B8%D0%BE%D1%82%D0%B5%D0%BA% D0%B5\_Tkinter\_%D1%8F%D0%B7%D1%8B%D0%BA%D0

%B0\_Python

## Что получит ученик по окончании модуля:

* Обучающийся познакомится с встроенной библиотекой компонентов графического интерфейса Tkinter.
* Научится работать с виджетами библиотеки Tkinter, позна- комится с событиями и научится описывать реакцию видже-

тов на события. Выполнит много несложных практических заданий, создавая простые GUI-приложения.

* С помощью Tkinter создаст приложения Сапер и Пинг-Понг, разберется с интеллектуальной начинкой этих игр и с тем, как ее реализовать на языке Python.

## Модуль 3. Криптография

**Урок 15. История криптографии. Старинные шифры**

Что такое криптография и чем она занимается? Как шифровали сообщения в древности? Старинные шифры атбаш и сцитала (шифр Древней Спарты).

## Урок 16. История криптографии. Старинные шифры

Чем отличается стеганография от криптографии. Диск и ли- нейка Энея. Шифр Цезаря.

**Урок 17. История криптографии. Старинные шифры** Шифрование с использованием таблиц. Квадрат Полибия. Ре- шетка Кардано.

## Урок 18. Игровое занятие. Разгадываем шифры.

Создание и расшифровка “таинственных посланий” с исполь- зованием вспомогательных материалов – карандаша и полоски бумаги для шифра считала, диска для шифра Цезаря и т.д.

## Урок 19. 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 расшифро- вать сообщение, зашифрованное шифром Цезаря с произволь- ным ключом. Идея перебора всех возможных ключей.

## Урок 20. Модификация и развитие шифра Цезаря.

Шифры, которые практически невозможно разгадать. Шифр Виженера.

## Урок 21. Современные алгоритмы шифрования

Современные алгоритмы шифрования и основные принципы их работы. Кто и зачем придумал RSA? Идея открытого и за- крытого ключа.

## Урок 22. Игровое соревновательное занятие “Взламываем шифр с открытым ключом”.Придумывание школьниками своих шифров для одноклассников и последующее разгадывание сообщений.

## Урок 23. Что такое электронная подпись?

Протокол аутентификации сообщений. Протокол электрон- но-цифровой подписи. Алгоритмы проверки электронной подписи. Алгоритм генерации электронной подписи.

## Урок 24. Кодирование текста.

Кодирование информации в компьютере. Сколько информа- ции 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.

**Полезные ресурсы:**

1. К. Вордерман и др. Программирование на Python: Иллю- стрированное руководство для детей. Издательство: Манн, Иванов и Фербер, 2018 г.
2. Программирование для детей на языке Python. Издатель- ство: АСТ, 2017 г.
3. Введение в криптографию. Под редакцией В.В.Ященко Из- дание четвертое, дополненное, Москва, МЦНМО, 2012.
4. https://ru.wikipedia.org/wiki/%D0%98%D1%81%D1%82%D 0%BE%D1%80%D0%B8%D1%8F\_%D0%BA%D1%80%D0

%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B 0%D1%84%D0%B8%D0%B8

1. https://habr.com/ru/company/yandex/blog/324866/
2. https://tproger.ru/translations/understanding-cryptography/

## Что получит ученик по окончании модуля:

* Обучающийся познакомится с историей криптографии и са- мыми знаменитыми шифрами.
* Потренируется решать задачи, в которых нужно разгадать за- шифрованное сообщение.
* Научится создавать криптографические приложения с ис- пользованием библиотеки Tkinter.
* Получит представление о современных методах шифрова- ния.
* Получит представление о кодировках текста и почему для русского текста существует так много кодировок.
* Научится работать с файлами в Python и создаст приложе- ние, позволяющее сохранять информацию в файле.
* Узнает, что такое словарь, и как с ним работать в языке Python.
* Познакомится с библиотеками для работы с датой и време- нем и создаст с их помощью несложное приложение.
* Напишет игру «Мемори».

## Модуль 4. Искусственный интеллект

**Урок 25. Искусственный интеллект. История.**

Что такое искусственный интеллект и что такое интеллект во- обще? Есть ли IQ у компьютерных программ? Как можно срав- нить человеческий и компьютерный интеллекты? Алан Тью- ринг и его работы. Немного про вычислительную сложность.

**Урок 26. Искусственный интеллект. Идея двоичного поиска.** Обсуждение задачи угадывания числа: учитель загадал нату- ральное 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 от- гадать загаданное число? Если перебирать числа последова- тельно (такой способ в программировании называется линей- ный поиск), за 99 вопросов точно можно угадать, но наверняка можно и за меньшее количество. Обсуждение со школьниками, школьники задают вопросы, учитель фиксирует на доске во- просы и отвечает “наихудшим” возможным образом. Обсуж- дение того, какие вопросы были хорошими, и почему. Обсуж- дение 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.

## Что получит ученик по окончании модуля:

* Обучающийся познакомится с понятием искусственного ин- теллекта и с историческими фактами, касающимися искус- ственного интеллекта.
* Разберется с идеей двоичного поиска.
* Напишет несколько простых консольных приложений.

## Модуль 5. Продвинутые библиотеки языка Python. Pygame.

**Урок 27. Знакомство с продвинутыми фреймворками для раз- работки GUI-приложений.**

Обзор дополнительных библиотек, позволяющих создавать приложения с графическим интерфейсом (PyQt, wxPython,Pygame). Примеры приложений. Установка библиотеки Pygame.

## Урок 28. Шаблон программы на Pygame

Как устроен шаблон программы на Pygame. Подключение мо- дулей, инициализация, главный цикл, корректное завершение работы приложения. Функции init(), set\_mode(), update(), quit(). События.

## Урок 29. Геометрические примитивы в Pygame

Цветовые модели. Рисуем и раскрашиваем линии, прямоуголь- ники, многоугольники, круги, овалы.

## Урок 30. Кейс 11. Простая анимация в Pygame.

Создание анимации движения геометрической фигуры от ле- вой границы главного окна вправо. Как изменить программу так, чтобы фигура не исчезала, а отражалась от стен.

## Урок 31. События клавиатуры.

Модуль pygame.event.

## Кейс 12: создание приложения, реализующего перемещение геометрической фигуры с помощью стрелок вправо/влево и вверх/вниз.

**Урок 32. 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.

**Урок 33. Дополнительные поверхности в Pygame.**

Surface и blit(). Простые примеры.

## Урок 34. Кейс 14. Работа в команде. Создание приложения “Поймай звезду”.

Создание игры, в которой в произвольном месте экрана появ- ляются звездочки, падающие вниз. При щелчке мыши по звез- дочке она исчезает, а в соответствующем текстовом поле идет подсчет пойманных звезд. Классы и объекты. Реализация по- явления звезд. Ребята разбиваются на группы, распределяют задачи внутри группы.

## 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 допол- нительных библиотеках языка Python, позволяющих разра- батывать приложения с GUI (PyQt, wxPython, Pygame).
* Обучающийся познакомится с принципами работы основ- ных элементов библиотеки Pygame и научится понимать код, использующий Pygame.
* В процессе обучения ученик разработает и напишет игровое приложение с использованием Pygame.

# 7. Планируемые результаты обучения.

## Важнейшими умениями/знаниями являются следующие:

* умение пользоваться персональным компьютером и его пе- риферийным оборудованием;
* умение следовать требованиям техники безопасности, гиги- ены, эргономики и ресурсосбережения при работе со сред- ствами информационных и коммуникационных технологий;
* умение осуществлять взаимодействие посредством элек- тронной почты, чата, форума;
* умение искать информацию с применением правил поиска (построения запросов), в компьютерных сетях, некомпью- терных источниках информации (справочниках и словарях, каталогах, библиотеках) при выполнении заданий и проек- тов по различным учебным дисциплинам;
* знакомство с основными конструкциями языка Python (ус- ловная инструкция, циклы, функции, списки, строки) на прак- тических примерах;
* умение работать со встроенной библиотекой компонентов графического интерфейса tkinter.
* формирование представления о некоторых дополнитель- ных библиотеках языка Python, позволяющих разрабатывать приложения с GUI (PyQt, wxPython, Pygame);
* формирование представления о современных методах шиф- рования;
* знакомство с понятием искусственного интеллекта и с исто- рическими фактами, касающимися искусственного интел- лекта;
* формирование умений разрабатывать несложные консоль- ные приложения и приложения с графическим интерфейсом;
* умение выбирать способ представления своего проекта с использованием соответствующих программных средств.

34

[**www.roskvantorium.ru/fond**](http://www.roskvantorium.ru/fond)

**Календарно – тематическое планирование.**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Тема урока** | **Кол-во часов** | **Дата проведения** |
| **факт** | **По плану** |
|  | **Основы языка Python.** | **6ч.** |  |  |
| 1. | Вводное занятие. | 1 |  |  |
| 2. | Основы языка Python. Ввод-вывод данных, числа | 1 |  |  |
| 3. | Основы языка Python. Операции с числами и строками. | 1 |  |  |
| 4. | Основы языка Python. Условная инструкция. | 1 |  |  |
| 5. | Основы языка Python. Логические операции. | 1 |  |  |
| 6. | Цикл for в Python. | 1 |  |  |
|  | **Создание приложений с помощью tkinter.** | **8ч.** |  |  |
| 7. | События в tkinter**.** | 1 |  |  |
| 8. | Создание простых приложений. | 1 |  |  |
| 9. | Создание простых приложений. | 1 |  |  |
| 10. | Создание простых приложений. | 1 |  |  |
| 11. | Продвинутое рисование в tkinter**.** | 1 |  |  |
| 12. | Движение нарисованных объектов. | 1 |  |  |
| 13. | Создание игрового приложения «пинг – понг» | 1 |  |  |
| 14. | Завершение над приложением «пинг – понг» | 1 |  |  |
|  | **Криптография**  | **13 ч.** |  |  |
| 15. | История криптографии. Старинные шрифты. | 1 |  |  |
| 16. | История криптографии. Старинные шрифты. | 1 |  |  |
| 17. | История криптографии. Старинные шрифты. | 1 |  |  |
| 18. | Игровое занятие. Разгадываем шифры. | 1 |  |  |
| 19. | Модификация и развитие шифра Цезаря. | 1 |  |  |
| 20. | Современные алгоритмы шифрования. | 1 |  |  |
| 21. | Современные алгоритмы шифрования. | 1 |  |  |
| 22. | Взламываем шифр | 1 |  |  |
| 23. | Что такое электронная подпись? | 1 |  |  |
| 24. | Кодирование текста. | 1 |  |  |
|  | **Искусственный интеллект.** | **2ч.** |  |  |
| 25. | Искусственный интеллект. История. | 1 |  |  |
| 26. | Искусственный интеллект. Идея двоичного поиска. | 1 |  |  |
|  | **Продвинутые библиотеки языка Python. Pygame.** | **8ч.** |  |  |
| 27. | Знакомство с продвинутыми фреймворками. | 1 |  |  |
| 28. | Шаблон программы на Pygame. | 1 |  |  |
| 29. | Геометрические примитивы на Pygame. | 1 |  |  |
| 30. | Простая анимация на Pygame. | 1 |  |  |
| 31. | События клавиатуры. | 1 |  |  |
| 32. | События мыши. | 1 |  |  |
| 33. | Дополнительные поверхности в Pygame. | 1 |  |  |
| 34. | Работа в команде. Создание приложения «Поймай звезду» | 1 |  |  |