Общество с ограниченной ответственностью «ЮМАКС»

УТВЕРЖДЕНО

Приказом №01/1-ОП от 15.01.2024

Генеральный директор ООО «Юмакс»

Мягков М.Ю.

М.П.

Дополнительная общеобразовательная программа

«Основы программирования на языке Python MAXIMUM»

Москва 2024

ОГЛАВЛЕНИЕ

1. Пояснительная записка. 3

- **Общие положения** 3
- **Востребованность программы** 3
- **Цели и задачи обучения** 5
- **Целевая аудитория и прием на обучение** 5
- **Форма и режим занятий** 5
- **КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК** 6

2. Учебный тематический план 7

3. Содержание Программы 7

4. Оценка результатов Программы и формы контроля. 12

5. Методическое обеспечение программы 13

5.1. Методические и учебные материалы 13

5.2. Учебно-материальная база 14

6. Литература 14

6.1 Для обучающихся 14

6.2 Для преподавателя 14

1. Пояснительная записка.

- **Общие положения**

Представленная программа является дополнительной общеобразовательной программой «Основы программирования на языке Python» является авторской, предметно-ориентированной. Данная программа разработана на основе оригинальных методик компании «Юмакс» разработанных в 2013 г., апробированых на протяжении 11 лет во многих учебных группах и являющихся результатом нескольких лет работ творческого коллектива компании «Юмакс». Программа «Основы программирования на языке Python» рассчитана на один год обучения. Учебные результаты программы носят социально-педагогический характер, что определяет ее направленность.

Реализация программы позволит обеспечить углубленную подготовку учащихся 7-9 классов общеобразовательной школы в области программирования на языке Python. Такая подготовка является чрезвычайно востребованной учащимися и родителями учащихся 7-9 классов.

Программа включает в себя следующие *тематические и проверочные модули*:

1. *Переменные и типы данных;*
2. *Условные операторы;*
3. *Циклы;*
4. *Списки;*
5. *Строки;*
6. *Пользовательские функции;*
7. *Модули;*
8. *Текстовые файлы;*
9. *Множества;*
10. *Кортежи;*
11. *Словари;*
12. *Вложенные структуры данных;*
13. *Обработка ошибок и исключений;*
14. *Создание Telegram-ботов.*

- **Востребованность программы**

Педагогическая целесообразность программы дополнительного образования «Основы программирования на языке Python» обусловлена, во-первых, тем, что программирование является одним из наиболее востребованных навыков в современной образовательной и профессиональной среде. Язык Python обладает большой практической значимостью, он широко используется в различных областях — от анализа данных до разработки программного обеспечения. Этот курс вносит особенно большой вклад в развитие логического и алгоритмического мышления учащихся, формируя базу для дальнейшего изучения информатики, математических и технических дисциплин. При этом академических часов, уделяемых программированию в общеобразовательных учреждениях, зачастую недостаточно для полного освоения современных технологий и подготовки учащихся к выполнению более сложных проектов, требуемых для успешной академической и профессиональной карьеры.

Во-вторых, педагогическая целесообразность программы обусловлена задачей всесторонней подготовки учащихся к будущей профессиональной деятельности и дальнейшему обучению. Программа позволяет, с одной стороны, актуализировать знания учащихся, полученные на ранних этапах обучения информатике и математике, с другой — углубить их понимание ключевых аспектов программирования. Она направлена на решение практических задач и создание проектов, что способствует закреплению теоретических знаний на практике и достижению высоких результатов в последующих образовательных курсах.

С целью лучшего освоения изучаемых тем, общее движение Программы от более простых заданий к более сложным сочетается с чередованием различных тематических разделов, таких как базовые структуры данных, циклы, условные операторы, функции, работа с файлами и т. д.

Выбор тем-модулей программы обусловлен их важностью для формирования базовых навыков программирования, востребованных как в учебных программах средних образовательных учреждений, так и в профессиональной среде. Модули программы могут корректироваться в зависимости от новых образовательных стандартов и технологий в области программирования. Программа ориентирована на задания, которые помогут учащимся овладеть языком Python на базовом уровне, а также выполнить задачи, характерные для профессионального использования языка. Особое внимание уделяется изучению тем, вызывающих затруднения у начинающих программистов, а также проработке типичных ошибок, выявленных при решении практических задач.

Таким образом, тематические модули, из которых состоит предлагаемая программа дополнительного образования, охватывают важные базовые

аспекты программирования на языке Python. Программа разработана как единый комплекс, взаимосвязанные части которого дополняют друг друга. Освоение отдельных тем программы в отрыве от общего процесса обучения не может обеспечить полного понимания программирования и получения высоких результатов в последующих учебных и профессиональных проектах.

Актуальность Программы обусловлена чрезвычайной практической значимостью изучения технологий написания программ, с целью получения практических навыков, требуемых для успешного понимания концепций программирования, чтобы тем самым получить представление о быстро развивающемся мире программирования и впоследствии более осознанно выбрать сферу для дальнейшего профессионального развития.

- **Цели и задачи обучения**

Цель Программы - *освоение обучающимися основ программирования на языке Python.*

Для достижения этой цели в процессе реализации программы решаются следующие **задачи**:

- формирование у учащихся умения использовать термины программирования, понимать различия между употреблением этих терминов;
- освоение синтаксиса языка программирования Python;
- формирование у учащихся умения составлять алгоритмы управления и записывать их на языке программирования Python;
- формирование у учащихся умения использовать логические значения, операции и выражения с ними;
- формирование у учащихся умения создавать и выполнять программы для решения алгоритмических задач;
- формирование у учащихся умения использовать готовые прикладные компьютерные программы и сервисы при реализации проектов.

- **Целевая аудитория и прием на обучение**

Возраст учащихся: Программа рассчитана на детей в возрасте от 13 до 15 лет, обучающихся в 7-9 классах общеобразовательной школы. В процессе обучения учитываются возрастные особенности детей – сформированность процессов восприятия, внимания, памяти, мышления и воображения. Опираясь на эти процессы, преподаватель программы формирует у обучающихся компетенции, необходимые для успешного освоения программы.

Сроки реализации программы: Программа «Основы программирования на языке Python MAXIMUM» рассчитана на 100 учебных часов. С учетом возможностей учеников заниматься дополнительно к основному образованию, она может быть реализована за один год.

- **Форма и режим занятий**

Расписание занятий формируется с учетом занятости учащихся в общеобразовательной школе, проходят в послеобеденное время в рабочие дни и в утренние или дневные часы в выходные. Обучение представлено в формате дистанционного электронного обучения (Образовательная платформа MAXIMUM). Занятия в формате электронного обучения предполагают изучение теоретического материала и отработку практических заданий. Продолжительность онлайн-консультаций 20 минут без перерыва.

Наполняемость групп – 6 человек.

Самостоятельные занятия учеников (домашние работы) включают в себя изучение теории, решение проверочных задач на знание теории, и решение задач по алгоритмам, изученным на классных занятиях.

Режим занятий:

1. Количество часов в неделю – не более 3 учебных часов;
2. Количество занятий в неделю – 1 раз в неделю;
3. Периодичность занятий – еженедельно один раз в неделю.
4. Начало занятий в рабочие дни с 16:00-19:00, в выходные с 11:00-14:00 и с 15:00-18:00.

- **КАЛЕНД**АРНЫЙ УЧЕБНЫЙ ГРАФИК

месяцы	1 месяц				2 месяц				3 месяц				4 месяц				5 месяц				6 месяц				7 месяц				8 месяц			
недели	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
Занятия 1 раз в неделю	2	2	2	2	2	2	2	3	2	2	2	2	2	2	2	3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	3

9 месяц				10 месяц				11 месяц				12 месяц				итого
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	48
2	2	2	2	2	2	2	3	2	2	2	2	2	2	2	2	100

2. Учебный тематический план

№ п/п	Тема	Всего часов	В том числе			Форма контроля
			лекции	практические занятия	самостоятельная работа	
1	Переменные и типы данных	4	-	2	2	
2	Условные операторы	4	-	2	2	
3	Циклы	9		4	4	Контрольная работа 1 ч.
4	Строки	6	-	3	3	
5	Списки	11	-	5	5	Контрольная работа 1 ч.
6	Пользовательские функции	4	-	2	2	
7	Модули	4	-	2	2	
8	Текстовые файлы	9	-	4	4	Контрольная работа 1 ч.
9	Множества	8	-	4	4	
10	Кортежи	4	-	2	2	
11	Вложенные структуры данных	11	-	5	5	Контрольная работа 1 ч.
12	Словари	10	-	5	5	
13	Обработка ошибок и исключений	2	-	1	1	
14	Создание Telegram-ботов	14	-	7	7	
	Итого	100	0	48	48	4

3. Содержание Программы

3.1.Переменные и типы данных

Раздел изучается в течение 4 часов, в процессе освоения которых учащиеся учатся создавать переменные и работать с базовыми типами данных языка Python.

В процессе подготовки ученики актуализируют или заново осваивают следующие темы:

- Понятие переменной и присваивание значений;
- Основные типы данных: целые числа (int), числа с плавающей точкой (float), строки (str), булевы значения (bool);
- Преобразование типов данных;

- Операции с числами: арифметические операторы (+, -, *, /, //, %, **);
- Логические операторы и операции сравнения (>, <, ==, !=, <=, >=);
- Ввод и вывод данных: функции input() и print();
- Функции работы с типами данных: type(), int(), float(), str(), bool().

3.2. Условные операторы

Раздел изучается в течение 4 часов, в процессе освоения которого учащиеся учатся использовать условные операторы на языке Python для создания программ, которые выполняют различные действия в зависимости от условий. Условные операторы играют ключевую роль в программировании, поскольку они позволяют программам принимать решения на основе заданных условий. Этот раздел помогает учащимся понять, как управлять потоком выполнения программы и использовать логику для решения задач с ветвлениями.

В процессе подготовки ученики осваивают следующие темы:

- Понятие условных операторов и их применение в программах;
- Условный оператор if, ветвления с elif и else;
- Вложенные условные операторы;
- Логические выражения для построения условий (использование операторов and, or, not);
- Правила составления и отладки условий;
- Практические задачи с использованием условных операторов.

3.3. Циклы

Раздел изучается в течение 9 часов, в процессе освоения которого учащиеся учатся использовать циклы на языке Python для автоматизации повторяющихся действий. Циклы позволяют выполнять однотипные действия многократно, что делает программы более эффективными и компактными.

В процессе подготовки ученики осваивают следующие темы:

- Понятие циклов и их значение в программировании;
- Цикл for: синтаксис и использование для многократного выполнения блоков кода;
- Цикл while: синтаксис и работа с условными повторениями;
- Функция range(): генерация числовых последовательностей для циклов;
- Вложенные циклы;
- Операторы управления циклами: break, continue;

- Алгоритмы поиска минимума и максимума с использованием циклов;
- Практические задачи по созданию программ с циклами для решения вычислительных задач.

В конце раздела ученики пишут контрольную работу, проверяющую знания по переменными и типам данных, условным оператора и циклам.

3.4.Строки

Раздел изучается в течение 6 часов, в процессе освоения которого учащиеся осваивают работу со строками на языке Python. Строки являются одним из базовых типов данных в Python, представляя собой последовательности символов. Работа со строками является важной частью программирования, так как строки часто используются для хранения и обработки текстовой информации.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие строки как последовательности символов;
- Индексация и срезы строк;
- Конкатенация и повторение строк;
- Основные методы и функции для работы со строками;
- Форматирование строк с использованием f-строк;
- Работа с вводом и выводом строк с использованием функций input() и print().

3.5.Списки

Раздел изучается в течение 11 часов, в процессе освоения которого учащиеся изучают работу со списками на языке Python. Списки являются одним из наиболее важных и универсальных типов данных в Python, представляя собой упорядоченные коллекции объектов. Работа со списками необходима для хранения и обработки наборов данных, что делает этот тип данных ключевым для решения многих задач программирования.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие списка как упорядоченной коллекции объектов;
- Индексация и срезы списков;
- Проход по элементам списка с использованием циклов;
- Добавление, удаление и изменение элементов списка;
- Основные методы и функции для работы со списками;
- Списочные выражения и их использование;
- Работа с вводом и выводом списков.

В конце раздела ученики пишут контрольную работу, проверяющую знания по спискам и строкам.

3.6.Пользовательские функции

Раздел изучается в течение 4 часов, в процессе освоения которого учащиеся знакомятся с созданием и использованием пользовательских функций на языке Python. Пользовательские функции помогают организовать код в компактные и повторно используемые блоки, что упрощает написание и поддержку программ.

В процессе подготовки учащиеся осваивают следующие темы:

- Основы функции как блока кода, выполняющего определённую задачу;
- Определение функций с помощью ключевого слова def и базовый синтаксис объявления функций;
- Параметры функций и возвращаемые значения с использованием ключевого слова return;
- Простое использование функций, включая вызов и передачу аргументов;
- Основные принципы работы с функциями и их значимость для организации кода;
- Область видимости переменных: различие между глобальными и локальными переменными.

3.7.Модули

Раздел изучается в течение 4 часов, в процессе освоения которого учащиеся знакомятся с модулями на языке Python. Модули позволяют организовывать код в отдельные, логически связанные файлы, что способствует лучшему структурированию и повторному использованию кода.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие модуля как отдельного файла с кодом, который можно импортировать в другие программы;
- Основы импорта модулей с использованием ключевого слова import и различных способов импорта (например, import module, from module import ...);
- Работа с встроенными модулями Python, такими как math, random и datetime;
- Использование функции dir() и help() для изучения содержимого модулей.

3.8.Текстовые файлы

Раздел изучается в течение 9 часов, в процессе освоения которого учащиеся знакомятся с работой с текстовыми файлами на языке Python. Работа с текстовыми файлами является важной частью программирования, так как она позволяет сохранять и обрабатывать данные, а также взаимодействовать с внешними источниками информации.

В процессе подготовки учащиеся осваивают следующие темы:

- Основы работы с текстовыми файлами: открытие, чтение, запись и закрытие файлов с использованием встроенных функций open(), read(), write() и close();
- Режимы открытия файлов (чтение, запись, добавление) и их влияние на работу с файлами;
- Чтение и запись данных в текстовом формате построчно с использованием специальных методов;
- Обработка больших файлов и работа с большими объемами данных с использованием построчной обработки;
- Сохранение данных в текстовом формате и их дальнейшее использование в программах.

В конце раздела ученики выполняют мини-проект, используя пользовательские функции, модули и текстовые файлы.

3.9.Множества

Раздел изучается в течение 8 часов, в процессе освоения которого учащиеся знакомятся с множествами на языке Python. Множества представляют собой неупорядоченные коллекции уникальных элементов, которые являются важным инструментом для выполнения операций над данными и решения множества задач программирования:

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие множества как коллекции уникальных элементов и его основное использование в Python;
- Создание множеств с помощью литералов {} и функции set(), а также преобразование других коллекций в множества;
- Основные операции над множествами: объединение (union() и |), пересечение (intersection() и &), разность (difference() и -), симметрическая разность (symmetric_difference() и ^);
- Добавление и удаление элементов из множества с использованием методов add(), remove(), discard() и pop();
- Проверка принадлежности элемента множеству и работа с подмножествами с помощью методов in, issubset(), issuperset(), и isdisjoint();

- Примеры применения множеств для решения практических задач, таких как удаление дубликатов из коллекций и выполнение операций над наборами данных;

3.10. Кортежи

Раздел изучается в течение 4 часов, в процессе освоения которого учащиеся знакомятся с кортежами на языке Python. Кортежи представляют собой упорядоченные коллекции элементов, которые являются неизменяемыми, что делает их полезными для хранения данных, которые не должны изменяться в процессе выполнения программы.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие кортежа как неизменяемой коллекции элементов и основные особенности кортежей в Python;
- Создание кортежей с помощью литералов () и функции tuple(), включая создание кортежей с одним элементом и пустых кортежей;
- Доступ к элементам кортежа с использованием индексации и срезов;
- Основные операции с кортежами: конкатенация, повторение и проверка на принадлежность элементов;
- Преимущества кортежей по сравнению с другими коллекциями, такими как списки;
- Распаковка кортежей и присвоение их элементов переменным;
- Основные методы кортежей, такие как count() и index(), и их применение в программировании;
- Примеры использования кортежей в практических задачах, таких как возврат нескольких значений из функции.

3.11. Вложенные структуры данных

Раздел изучается в течение 11 часов, в процессе освоения которого учащиеся подробно знакомятся с вложенными структурами данных на языке Python. Вложенные структуры данных позволяют создавать более сложные и многослойные коллекции информации, такие как списки, словари и множества, содержащие другие коллекции.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие вложенных структур данных и их использование для организации и хранения информации;
- Создание и работа со списками, содержащими другие списки (двумерные и многомерные списки);

- Создание и работа со словарями, содержащими другие словари или списки, включая вложенные структуры данных;
- Применение множеств внутри других коллекций для решения задач, связанных с уникальными элементами и операциями над наборами данных;
- Индексация, срезы и доступ к элементам вложенных структур данных, включая использование вложенных циклов для обработки многомерных коллекций;
- Основные операции с вложенными структурами: добавление, удаление и изменение элементов на различных уровнях вложенности;
- Примеры использования вложенных структур данных для представления сложных объектов, таких как таблицы данных, графы и иерархии;

В конце раздела ученики пишут контрольную работу, проверяющую знания по пользовательским функциям, модулям, текстовым файлам, множествам и кортежам.

3.12. Словари

Раздел изучается в течение 10 часов, в процессе освоения которого учащиеся знакомятся со словарями на языке Python. Словари представляют собой неупорядоченные коллекции пар 'ключ-значение', что делает их мощным инструментом для хранения и обработки данных, которые ассоциируются друг с другом.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие словаря как коллекции пар 'ключ-значение' и основные особенности словарей в Python;
- Создание словарей с помощью литералов {} и функции dict(), включая создание пустых словарей и словарей с начальными значениями;
- Доступ к значениям словаря по ключам, добавление, изменение и удаление пар 'ключ-значение' с использованием методов get(), update(), pop() и del;
- Проверка наличия ключей в словаре с помощью операторов in и методов keys(), values(), и items();
- Итерация по ключам, значениям и парам 'ключ-значение' в словаре с использованием циклов и методов items(), keys(), и values();
- Работа со вложенными словарями и создание сложных структур данных с использованием словарей в качестве значений для других словарей или коллекций;

- Примеры применения словарей для решения практических задач, таких как подсчёт частоты элементов, создание адресных книг и управление конфигурациями;

В конце раздела ученики пишут контрольную работу, проверяющую знания по вложенным структурам данных и словарям. Также в конце раздела ученики выполняют мини-проект, используя вложенные структуры данных и словари.

3.13. Обработка ошибок и исключений

Раздел изучается в течение 2 часов, в процессе освоения которого учащиеся знакомятся с обработкой ошибок и исключений на языке Python. Обработка ошибок и исключений позволяет управлять непредвиденными ситуациями, которые могут возникнуть во время выполнения программы, и обеспечивает более надёжную и устойчивую работу кода.

В процессе подготовки учащиеся осваивают следующие темы:

- Понятие исключений как механизмов для обработки ошибок и непредвиденных ситуаций в программе;
- Основные виды исключений и ошибки, которые могут возникнуть во время выполнения программы (например, ZeroDivisionError, FileNotFoundError, TypeError);
- Использование конструкции try и except для перехвата и обработки исключений, включая синтаксис и базовые принципы работы;
- Обработка нескольких исключений с использованием нескольких блоков except;
- Использование блока else для выполнения кода, если исключений не произошло, и блока finally для выполнения кода, который должен выполниться в любом случае (например, закрытие файлов);

3.14. Создание Telegram-ботов

Раздел изучается в течение 14 часов, в процессе освоения которого учащиеся знакомятся с созданием Telegram-ботов на языке Python с использованием библиотеки pyTelegramBotAPI. Знания, полученных в рамках данного раздела, позволяют разрабатывать приложения, которые могут взаимодействовать с пользователями через популярную платформу обмена сообщениями, а также автоматизировать различные задачи и интегрировать с другими сервисами.

В процессе подготовки учащиеся осваивают следующие темы:

- Основы работы с Telegram-ботами: создание бота через BotFather, получение токена и настройка бота для работы с API Telegram;
- Установка и настройка библиотеки pyTelegramBotAPI, включая установку через pip и базовые настройки;
- Основные принципы работы с библиотекой pyTelegramBotAPI: создание экземпляра бота и настройка обработки сообщений и команд;
- Реализация простых команд бота с использованием декораторов @bot.message_handler и функций-обработчиков для команд и текстовых сообщений;
- Работа с типами сообщений и методами Telegram API: отправка текстовых сообщений, фото, видео и других медиафайлов;
- Обработка пользовательского ввода и реализация интерактивных функций: кнопки, инлайн-кнопки и обработка пользовательских запросов;
- Работа с клавиатурами и меню: создание пользовательских клавиатур и инлайн-кнопок для улучшения взаимодействия с пользователем;
- Обработка ошибок и исключений при работе с библиотекой pyTelegramBotAPI для создания более устойчивых и надёжных приложений;
- Примеры использования ботов для различных задач, таких как автоматизация задач, игры, опросы и уведомления.

В конце раздела ученики создают финальный проект, проверяющий знания по всему курсу. Финальный проект может содержать в своей основе как Telegram-бот, так и консольное приложение на языке Python.

4. Оценка результатов Программы и формы контроля.

Предполагается, что учащиеся, успешно завершившие дополнительную общеобразовательную программу «Основы программирования на языке Python», смогут значительно улучшить свои навыки программирования, уверенно решать задачи по созданию программных решений и более эффективно применять знания на практике. Они будут лучше подготовлены к будущему изучению более сложных аспектов программирования и смогут применить приобретенные знания в различных учебных и практических ситуациях.

В рамках программы учащиеся выполняют домашние задания и проектные работы, которые вносятся в образовательную платформу для мониторинга их выполнения и получения обратной связи. Преподаватель получает информацию о выполнении заданий и качестве работы учащихся на

протяжении всей программы, что позволяет оперативно отслеживать прогресс и корректировать учебный процесс.

Оценивание осуществляется преподавателем, который учитывает как результаты домашних заданий и тестов, так и общие достижения учащихся в рамках курса. Особое внимание уделяется выполнению итогового проекта, который служит основной формой оценки результатов программы. Итоговый проект предполагает разработку полноценного программного приложения на языке Python, демонстрирующего освоенные учащимися навыки и умения.

Итоговые результаты программы оцениваются на основе выполнения итогового проекта. Преподаватель проводит оценку проекта, предоставляя подробную обратную связь, которая включает как положительные аспекты работы, так и рекомендации по улучшению. Результаты проекта обсуждаются с учащимися, что позволяет скорректировать их подход к программированию и подготовке к дальнейшему изучению языка Python.

5. Методическое обеспечение программы

5.1. Методические и учебные материалы

Каждое занятие Программы описано в методическом пособии для преподавателей. Общий алгоритм проведения занятий состоит из нескольких этапов:

- постановка целей урока, описание того, что учащиеся должны достигнуть в результате урока;

- указание на место урока и его функции в общей программе обучения программированию на Python, включая связь с последующими темами курса и их применение в реальных задачах;

- демонстрационное решение типовых задач и примеров по теме урока, с акцентом на разъяснение алгоритма решения и основных концепций, таких как работа со списками, функциями, модулями и т.д.;

- закрепление знаний учащихся путем выполнения практических заданий, которые направлены на применение изученного алгоритма на практике, включая задачи на создание функций, работу с файлами, использование библиотек и другие аспекты программирования на Python;

- привязка освоенных теоретических и практических навыков к требованиям курса программирования в общеобразовательных учреждениях, с целью подготовки учащихся к выполнению более сложных программных проектов и эффективного использования языка Python в различных контекстах.

В процессе урока учащимися применяется учебное пособие, разработанное ООО «Юмакс».

5.2. Учебно-материальная база

Для реализации программы необходимо то же оборудование, что и для ведения учебного процесса в среднем общем образовании в формате вебинаров – стационарный компьютер или ноутбук, веб-камера, микрофон и подключение к сети Интернет. Программа реализуется в формате видеоконференции, в группах до 6 человек.

В качестве учебных пособий используются материалы, разработанные компанией ООО «Юмакс». Это методические материалы для преподавателей и наборы домашних заданий для учеников.

6. Литература

6.1 Для преподавателя

1. Методические материалы к урокам Программы по Программированию на языке Python. М.: издание ООО «Юмакс», 2024 – 2025 гг.