

Управление образования администрации Надеждинского муниципального
района

Индивидуальный предприниматель Жураковская Евгения Сергеевна



УТВЕРЖДАЮ:

Е.С. Жураковская

1 августа 2024г.

ПРОГРАММИРОВАНИЕ PYTHON START

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

Возраст обучающихся: 12–13 лет

Срок реализации программы: 1 год

Голубцова Полина Сергеевна, методист
Липчанская Мария Анатольевна, методист

Надеждинский муниципальный район
2024

РАЗДЕЛ № 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1. Пояснительная записка

Программирование уже изменило повседневную жизнь. Оно поменяло наши привычки поиска информации, подходы к заботе о здоровье, прослушиванию музыки, общению с друзьями. Системные занятия программированием формируют комплекс навыков, которые можно использовать в повседневной жизни. Среди них умение непрерывно учиться новому, использование передовых технологий, работа в команде, гибкость, адаптивность, внимательность, самостоятельность и инициативность, способность к рефлексии и высокий уровень критического мышления.

Программа позволяет учащимся познакомиться с языком программирования Python через захватывающее путешествие в разработку игр и приложений. Современный мир предъявляет новые требования к молодому поколению, вступающему в жизнь, так как будущее сегодняшних детей – это цифровое общество.

Развитие навыков программирования является уникальным способом формирования интереса учащихся к техническим направлениям деятельности, а также совершенствования алгоритмического, креативного мышления и навыков проектной деятельности.

Программа составлена с учётом задач национального проекта «Цифровая экономика» (федерального проекта «Кадры для цифровой экономики») и национального проекта «Образование» (федерального проекта «Молодые профессионалы»): создание условий для формирования у учащихся основ цифровых компетенций, необходимых для дальнейшего участия в разработке и поддержке инфраструктур передачи, обработки и хранения данных, а также цифровых продуктов для граждан, бизнеса и власти; разработка, внедрение цифровых технологий и платформенных решений.

Направленность программы – техническая.

Язык реализации программы: государственный язык РФ – русский.

Уровень освоения – стартовый.

Отличительные особенности

Обучение осуществляется в рамках ролевой игры, сюжетная линия которой охватывает весь курс. Курс построен таким образом, что дает возможность обучающимся пройти через различные этапы работы в большой IT-компании. Обучающийся проходит путь от собеседования и начала карьеры стажера-разработчика (intern python developer) до ведущего разработчика (senior python developer). Данные задачи не только способствуют формированию предметных компетенций, но еще и формируют взгляд на мир, раскрывают значение soft skills, учат грамотному использованию компьютера и готовят к жизни в цифровом обществе.

В рамках программы, обучающиеся освоят базовые принципы программирования на языке Python через создание собственных проектов и решение задач разной сложности (от исправления ошибки в чужой программе до запуска beta-версии своего продукта).

Особенность программы проявляется в том, что, изучив необходимый теоретический материал, обучающимся предлагается самостоятельно решить практическую задачу, представленную в виде адаптированного кейса из реальной жизни. Для этого, обучающиеся проходят через следующие этапы: определяют и составляют последовательность действий, ведущую к цели, и записывают её с помощью формального языка. Полученные знания, обучающиеся применяют при создании собственных проектов, которые защищаются перед другими учениками, педагогами и родителями.

Новизна программы заключается в превалировании прикладного аспекта над фундаментальным, что позволяет изучать программирование с нуля без значительной опоры на математику, демонстрировать применение полученных знаний в решении задач из реальной жизни и мотивировать обучающихся с разным уровнем подготовки. В отличие от других курсов программирования Python Start предоставляет обучающимся возможность думать, как программисты, а не просто

решать стандартный набор задач. Более того, проекты, выполняемые на занятиях, охватывают разные области интересов, в том числе школьные предметы. Также обучающиеся развивают умение самопрезентации и презентации своих проектов или идей как в группе, так и перед учителями и родителями.

Авторское воплощение замысла программы заключается в том, что обучающиеся используют инструменты профессиональной разработки для реализации собственных проектов. Для этого сначала обучающийся получает базовые знания (алгоритмы, структурное программирование, использование библиотек в процедурном стиле), затем учится навыкам проектной деятельности, после чего реализует проекты в области своих увлечений. При этом базовая часть также подразумевает возможность выразить свои интересы, благодаря большому количеству задач разного уровня. Проекты выполняются преимущественно индивидуально, чтобы дети сначала научились работать над ними, затем уже – работать в группе. Эволюция компьютеров и программного обеспечения привела к достаточной простоте их освоения для самых неподготовленных пользователей, в том числе детей

12-13 лет.

Адресат программы – обучающихся образовательных организаций Надеждинского муниципального района в возрасте от 12 до 13 лет.

Организация образовательного процесса

Срок освоения программы – 1 год. Объем программы – 72 часа.

Занятия проходят 1 раз в неделю и длятся 2 академических часа. Длительность академического часа 40 минут. Дети занимаются в кабинете с педагогом, самостоятельно работая за компьютером (непрерывно — не более 20 минут), предусмотрен перерыв.

Набор на программу осуществляется на добровольной основе и не требует наличия дополнительных знаний. Зачисление производится на основании заявления родителя/законного представителя. Количество обучающихся в группе от 6 до 12 человек.

Программа реализуется с помощью учебной платформы «Алгоритмика».

1.2. Цель и задачи программы

Цель программы: развитие технического мышления у обучающихся 12-13 лет посредством программирования на языке Python.

Задачи

Воспитательные:

1. Воспитывать обучающегося, обладающего информационной культурой.
2. Воспитывать стремление к познавательной активности, исследовательским и прикладным способностям.
3. Воспитывать у обучающихся самостоятельность, инициативность, целеустремлённость.

Развивающие:

1. Развивать логическое, креативное и пространственное мышление.
2. Совершенствовать навык публичного выступления и презентации.
3. Развивать и расширять специальный словарный запас как на русском, так и на английском языках.
4. Развивать память, внимание, творческое воображение.
5. Развивать навык планирования деятельности и функциональную грамотность.

Обучающие:

1. Знакомить с базовыми понятиями и особенностями синтаксиса языка программирования Python.
2. Обучать принципам объектно-ориентированного программирования.
3. Формировать навык правильного оформления кода.
4. Обучать решению задач и разработке проектов с помощью Python.
5. Обучать разработке игр с помощью языка программирования Python.
6. Формировать навык тестирования и оптимизации созданных проектов.

1.3. Содержание программы

Учебный план

№	Наименование разделов, тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1	Основы языка Python	8	4	4	Проект Опрос
1.1	Введение в Python. Функции ввода-вывода	2	1	1	
1.2	Переменные. Численные типы данных	2	1	1	Сопоставление
1.3	Строки	2	1	1	Заполнение пропусков
1.4	Вложенные конструкции	2	1	1	Презентация
2	Управляющие конструкции	10	5	5	Проект
2.1	Логический тип данных. Условный оператор	2	1	1	Опрос
2.2	Вложенный условный оператор и условный оператор нескольких ветвей	2	1	1	Сопоставление
2.3	Цикл while. Цикл со счетчиком	4	2	2	Сортировка
2.4	Вложенные алгоритмические конструкции (цикл в цикл, условный оператор в цикл и др.)	2	1	1	Презентация
3	Функции и модули	8	4	4	Проект
3.1	Создание функций. Локальная и глобальная области видимости	2	1	1	Классический тест
3.2	Использование одной функции внутри другой	2	1	1	Сопоставление
3.3	Модули. Использование встроенных модулей стандартной библиотеки	2	1	1	Заполнение пропусков
3.4	Создание модулей. Подключение нескольких модулей	2	1	1	Презентация
4	Модуль Turtle. Математика для разработчика	10	5	5	Проект
4.1	Графический исполнитель Turtle. Математика перемещения исполнителя	3	1	2	Сортировка
4.2	Условный оператор. Рисование простых геометрических фигур	2	1	1	Сопоставление
4.3	Цикл. Рисование многоугольников	3	1	2	Заполнение пропусков
4.4	Проектное занятие	2	1	1	Презентация
5	Объектно-ориентированное программирование	12	6	6	Проект
5.1	Объекты, их поля и методы	2	1	1	Классический тест

5.2	Обработка событий мыши и клавиатуры	6	3	3	Сопоставление
5.3	Классы. Конструктор класса	2	1	1	Сортировка
5.4	Наследование (от готового или своего класса)	2	1	1	Проект
6	Основы разработки игр на PyGame	24	12	12	Проект
6.1	Основы разработки игр. Спрайт, игровой цикл	2	1	1	Опрос
6.2	Списки и методы работы с ними. Перебор списка в цикле for	2	1	1	Сопоставление
6.3	Обработка внутриигровых событий	2	1	1	Сортировка
6.4	Игровая физика	8	4	4	Классический тест
6.5	Воркшоп для разработчиков	2	1	1	Презентация проекта
6.6	Хакатон. Test IT	2	1	1	Презентация проекта
6.7	ООП. Проект Simple Paint	2	1	1	Презентация проекта
6.8	Хакатон. Grow Up	2	1	1	Презентация проекта
6.9	Итоговое занятие по программе	2	1	1	Презентация проекта
	Итого:	72	36	36	

Содержание учебного плана

1. Раздел: Основы языка Python

1.1. Тема: Введение в Python. Функции ввода-вывода

Теория. Язык программирования Python. Понятия «язык программирования», «алгоритм», «программа». Синтаксис языка. Функция ввода input(). Функция вывода print(). Необходимость вывода данных.

Практика. Разработка программы, осуществляющей ввод-вывод данных, ввод команд с клавиатуры.

1.2. Тема: Переменные. Численные типы данных

Теория. Данные. Типы данных. Переменная. Числовой тип данных. Оператор присваивания. Приоритет операций. Выражение.

Практика. Разработка программ, осуществляющих обработку переменных числового типа.

1.3. Тема: Строки

Теория. Строковый тип данных. Операции со строками и числами. Изменение типов данных с помощью функций `int()` и `str()`.

Практика. Разработка программ, осуществляющих обработку строковых переменных (сумма строк, повтор строки n-раз, вычисление длины строки, получение символа строки по номеру, замена подстроки на другую подстроку, разбиение строки по разделителю).

1.4. Тема: Вложенные конструкции

Теория. Понятие. Виды. Способы создания.

Практика. Создание вложенных конструкций.

2. Раздел: Управляющие конструкции

2.1. Тема: Логический тип данных. Условный оператор

Теория. Логический тип данных. Утверждение. Условие. Условный оператор. Синтаксис условного оператора. Операторы сравнения.

Практика. Разработка программ, содержащих условный оператор.

2.2. Тема: Вложенный условный оператор и условный оператор нескольких ветвей

Теория. Вложенный условный оператор. Уровни вложенности. Условный оператор нескольких ветвей.

Практика. Разработка программ, содержащих вложенный условный оператор.

2.3. Тема: Цикл while. Цикл со счетчиком

Теория. Цикл. Необходимость применения циклов. Типы циклов. Демонстрация синтаксиса цикла `while()`. Разбор примеров программ с использованием цикла. Цикл со счётчиком. Цикл с накоплением результата вычислений в переменной.

Практика. Разработка программ, содержащих циклы различных типов.

2.4. Тема: Вложенные алгоритмические конструкции (цикл в цикл, условный оператор в цикл и др.)

Теория. Вложенные алгоритмические конструкции. Цикл в цикле. Цикл с вложенными условными операторами. Целочисленное деление.

Практика. Разработка программ, осуществляющих обработку данных с помощью вложенных алгоритмических конструкций.

3. Раздел: Функции и модули

3.1. Тема: Создание функций. Локальная и глобальная области видимости

Теория. Функция. Локальная и глобальная функции. Оператор def. Параметры функции. Значение функции. Вызов функции. Синтаксис создания функций.

Практика. Разработка программ, осуществляющих обработку данных с помощью функций.

3.2. Тема: Использование одной функции внутри другой

Теория. Вложенные (или внутренние) функции, которые находятся внутри других функций. Прямой доступ к переменным и именам, определённым во включающей её функции. Применение вложенных функций. Обеспечение инкапсуляции и скрытие функции от внешнего доступа. Описание вспомогательных функций для облегчения повторного использования кода. Реализация замыкания для сохранения состояний между вызовами. Создание декораторов для добавления поведения к существующим функциям.

Практика. Разработка программ, содержащих функцию внутри другой функции.

3.3. Тема: Модули. Использование встроенных модулей стандартной библиотеки

Теория. Модуль. Встроенные модули стандартной библиотеки. Модуль random.

Практика. Разработка программ, осуществляющих обработку данных с помощью модулей, встроенных в стандартную библиотеку.

3.4. Тема: Создание модулей. Подключение нескольких модулей

Теория. Файл, содержащий код на Python. Переменные, классы и функции. Исполняемый код.

Практика. Разработка программ, осуществляющих обработку данных с помощью дополнительных модулей, не входящих в стандартную библиотеку.

4. Раздел: Модуль Turtle. Математика для разработчика

4.1. Тема: Графический исполнитель Turtle. Математика перемещения исполнителя

Теория. Исполнитель. Графический исполнитель. Пиксель. Координаты. Модуль Turtle: рисование линий, поворот ширина и цвет пера, перемещение пера в указанную точку. Математика перемещения исполнителя.

Практика. Разработка программ, управляющих исполнителем.

4.2. Тема: Условный оператор. Рисование простых геометрических фигур

Теория. Условный оператор в модуле Turtle. Рисование простых геометрических фигур.

Практика. Разработка программ, осуществляющих рисование простых геометрических фигур с использованием условного оператора.

4.3. Тема: Цикл. Рисование многоугольников

Теория. Циклы в модуле Turtle. Первый цикл (внешний цикл), определяющий количество рисуемых многоугольников. Второй цикл (внутренний цикл), отвечающий за прорисовку каждого многоугольника. Вложенная алгоритмическая конструкция или вложенный цикл.

Практика. Разработка программ, осуществляющих рисование многоугольников.

4.4. Тема: Проектное занятие

Теория. Сбор изученных навыков в единый продукт.

Практика. Тестирование продукта.

5. Раздел: Объектно-ориентированное программирование

5.1. Тема: Объекты, их поля и методы

Теория. Объектно-ориентированное программирование в Python – способ компьютерного программирования, использующий идею «объектов» для представления данных и методов. Подход, используемый для создания аккуратного

и многоразового кода вместо избыточного. Объекты. Поля и методы объектов. Создание объектов.

Практика. Разработка программ, осуществляющих обработку данных с помощью передачи объекта в функцию.

5.2. Тема: Обработка событий мыши и клавиатуры

Теория. События. Обработка взаимодействия пользователя и программы. Обработка событий мыши и клавиатуры.

Практика. Разработка программ, осуществляющих обработку событий.

5.3. Тема: Классы. Конструктор класса

Теория. Классы. Конструктор класса. Создание нестандартного объекта и описание его свойств и методов.

Практика. Разработка программ, осуществляющих обработку классов.

5.4. Тема: Наследование (от готового или своего класса)

Теория. Наследование. Класс-наследник. Использование готовых свойств и методов объекта. Добавление новых методов.

Практика. Разработка программ, использующих наследование.

6. Раздел: Основы разработки игр на PyGame

6.1. Тема: Основы разработки игр. Спрайт, игровой цикл

Теория. Спрайт. Игровой цикл: обработка событий, обновление игры, прорисовка, подсчет времени. Библиотека play. Спрайты разных типов и с разными свойствами. Расположение спрайта.

Практика. Разработка программ для создания спрайтов разных типов и управления ими.

6.2. Тема: Списки и методы работы с ними. Перебор списка в цикле for

Теория. Списки. Методы работы со списками. Перебор списков в цикле for.

Практика. Разработка программ, осуществляющих обработку списков.

6.3. Тема: Обработка внутриигровых событий

Теория. Внутриигровые события. Обработка внутриигровых событий.

Практика. Разработка программ, осуществляющих обработку внутриигровых событий.

6.4. Тема: Игровая физика

Теория. Игровая физика. Работа со спрайтами как с физическими объектами. Подключение физики. Применение инструментов при создании игр.

Практика. Разработка игр с использованием встроенной симуляции физики.

6.5. Тема: Воркшоп для разработчиков

Теория. Soft и hard skills. Их отличия. Значение развития обе категории навыков. Структура собеседования разработчика Python. Примеры задач. Общий подход к работе над задачей. Концепция и особенности проведения хакатонов.

Практика. Обсуждение кейсов. Выполнение практического задания на платформе «Навыки разработчика». Выполнение упражнения «Калибровка» для тренировки навыка управления временем.

6.6. Тема: Хакатон. Test IT

Теория. Функции и модули на хакатоне по программированию. Hard и soft skills. Создание функций. Создание модулей. Принципы командной работы. Структура приложения TestIT. План сборки приложения

Практика. Деление разработчиков на команды. Распределение задач между разработчиками. Выполнение первого и второго этапов хакатона. Подготовка презентации проектов. Презентация проектов. Определение проекта-победителя (например, наибольшее число лайков).

6.7. Тема: ООП. Проект Simple Paint

Теория. Важность пользовательского интерфейса и внедрение его в программу. Принципы доработки графического редактора. Планирование работы над проектом. События: подписка и обработка. Необходимость реализации пользовательского интерфейса в программе. Варианты расположения элементов интерфейса. Реализация пользовательского интерфейса на примерах. План доработки функционала проекта.

Практика. Доработка чек-листа работы над проектом и mind map по причине появления новых задач. Работа над проектом Simple Paint. Интерфейс. Подготовка презентации проекта. Публикация проектов в «Зале славы».

6.8. Тема: Хакатон. Grow Up

Теория. Создание классов. Наследование. Плюсы и принципы командной работы. Структура приложения Grow_Up. План доработки и тестирования приложения.

Практика. Деление разработчиков на команды. Распределение задач между разработчиками. Выполнение первого и второго этапов хакатона. Подготовка к презентации проектов. Презентация проектов. Определение проекта-победителя (например, наибольшее количество баллов в турнирной таблице).

6.9. Тема: Итоговое занятие по программе

Теория. Обобщение и систематизация пройденного материала в игровой форме.

Практика. Презентация итогового проекта

1.4. Планируемые результаты

Личностные результаты

Обучающийся будет способен:

- самостоятельно планировать и осуществлять деятельность, связанную с цифровой средой.

У обучающегося будет сформировано:

- понятие цифровой культуры;
- устойчивый интерес к проектной деятельности.

Метапредметные результаты

Обучающийся будет уметь:

- эффективно планировать деятельность;
- применять изученные технологии в повседневной жизни;
- выполнять логические операции: сравнение, анализ, обобщение, установление аналогий.

Обучающийся приобретёт:

- способность самостоятельно анализировать информацию;
- навык представления результатов своей деятельности публично;

- умение использовать цифровую грамотность в учебно-исследовательской и других видах деятельности;

- умение использовать инструменты цифровых сервисов, приложений, программ.

Предметные результаты

Обучающийся будет знать:

- синтаксис языка программирования Python;

- основы программирования в т. ч. алгоритмические конструкции, логические значения и операции.

Обучающийся будет уметь:

- следовать стандарту при оформлении кода;

- тестировать программу и находить в ней ошибки;

- использовать чужой код при разработке: анализировать и изменять.

Обучающийся будет владеть:

- языком Python для решения задач практического характера и задач из смежных дисциплин;

- итерационным подходом.

РАЗДЕЛ № 2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

2.1. Условия реализации программы

Материально-техническое обеспечение:

- учебный кабинет с необходимой мебелью для ведения образовательного процесса (парты/столы, стулья, шкаф для методических материалов);

- персональные компьютеры или ноутбуки по количеству обучающихся;

- мультимедийный проектор;

- доска или флипчарт, мел или маркер;

- бумага А4, ручки/карандаши;

- приложение или доступ на платформу «Алгоритмика» с интеграцией PyGame;

- логины и пароли для доступа на обучающую платформу «Алгоритмика».

2. Учебно-методическое и информационное обеспечение:

- поурочные методические рекомендации к занятиям издательства «Алгоритмика»;
- тематические презентации;
- задания на платформе «Алгоритмика» с автопроверкой;
- справочный теоретический материал, размещенный на платформе «Алгоритмика»;
- бонусные задания на платформе;
- задачи со свободным решением для реализации в среде разработки на базе платформы «Алгоритмика».

2.2. Оценочные материалы и формы аттестации

Контроль сформированности результатов освоения программы осуществляется с помощью нескольких инструментов:

- **на каждом занятии:** выполнение заданий на платформе, взаимоконтроль учеников в парах, самоконтроль ученика;
- **в конце каждого модуля:** проведение презентации финальных проектов модуля и их оценка;
- **после третьего модуля:** промежуточная аттестация в форме опроса обучающихся;
- **после освоения всех модулей:** итоговый контроль в форме презентации проекта.

Используются следующие формы аттестации:

- классический тест - выбор одного или нескольких правильных ответов в шаблоне с вопросами;
- заполнение пропусков – шаблон текста с пропусками для заполнения;
- классификация – шаблон с элементами для группировки объектов в соответствии с их общими признаками;
- сопоставление – шаблон с элементами для сравнения двух элементов для того, чтобы показать их сходства или различия;

- сортировка – шаблон с элементами для последовательного расположения или формирования в группы элементов в зависимости от выбранного критерия;
- ответ в свободной форме, в т. ч. прикрепление файла – шаблон для внесения ответа в свободной форме;
- опрос - шаблон для проверки знаний посредством устных или письменных вопросов;
- презентация – представление информации в форме текста, таблицы, слайдов.
- проект – уникальный индивидуальный или групповой план, направленный на достижение конкретной цели, практического применения полученных знаний.

Возможны следующие **формы деятельности** учащихся: учебная дискуссия, занятие-проект, практическое занятие. При решении одной задачи ученики могут работать индивидуально, затем в парах, а при проверке задачи – в паре с педагогом.

2.3. Методические материалы

Методические материалы предоставляются платформой «Алгоритмика» и состоят из:

- поурочных методических рекомендаций к занятиям;
- тематических презентаций;
- заданий на платформе;
- справочного и теоретического материала, размещенного на платформе;
- бонусных заданий на платформе;
- задач со свободным решением для реализации в среде разработки на базе платформы.

Методическая литература для успешной реализации программы:

1. Лапчик М.П., Семакин И.Г., Хеннер Е.К., Самылкина Н.Н. Теория и методика обучения информатике. М.: Академия, 2008.
2. Асмолов А.Г., Бурменская Г.В., Володарская И.А. и др. Формирование универсальных учебных действий в основной школе: от действия к мысли. Система заданий. / Под ред. А.Г. Асмолова – 2-е изд. М.: Просвещение, 2011.

3. Брайсон Пэйн. Python для детей и родителей. М.: ООО «Эксмо», 2017.

4. Вербицкий А.А., Ларионова О.Г. Личностный и компетентностный подходы в образовании: проблемы интеграции. Монография. М.: Издательство «Логос», 2017.

5. Фридман Л.М. Как научиться решать задачи. М.: Просвещение, 1989.

Дополнительная методическая литература

1. Актуальные проблемы методики обучения информатике в современной школе: материалы Международной научно-практической интернет-конференции, г. Москва, 24–26 апреля 2018 г. / Под ред. Л.Л. Босовой, Н.К. Нателаури; Московский педагогический государственный университет. Кафедра теории и методики обучения информатике [Электронное издание]. М.: МПГУ, 2018.

2. Вордерман К., Вудкок Дж., Макаманус Ш. Программирование для детей. Иллюстрированное руководство для детей; пер. с англ. С. Ломакина. 4-е изд. М.: Манн, Иванов и Фербер, 2018.

3. Вольдерман К., Стили К., Квигли К. и др. Программирование на Python: иллюстрированное руководство для детей. М.: Манн, Иванов и Фербер, 2018.

4. Кларин М.В. Инновационные модели обучения: Исследование мирового опыта. Монография. 2-е издание. М.: Луч, 2018.

2.4. Календарный учебный график

Этапы образовательного процесса	1 год	
Продолжительность учебного года, неделя	32	
Количество учебных дней	36	
Продолжительность учебных периодов	1 полугодие	15.09.2023-30.12.2023
	2 полугодие	10.01.2024-31.05.2024
Возраст детей, лет	12-13	
Продолжительность занятия, академический час	1	
Режим занятия	1 раза/нед	
Годовая учебная нагрузка, час	72	

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Курс: «Python для взрослых и детей». [Электронный ресурс] // URL: <https://stepik.org/course/4401> (Дата обращения: 23.03.2023).

2. Python для начинающих: с чего начать изучение. [Электронный ресурс] // URL: <https://education.yandex.ru/journal/start-python>. (Дата обращения: 10.03.2024).

3. Хакатон python для начинающих. [Электронный ресурс] // URL: <https://pygame.ru/blog/hakaton-python-dlya-nachinayushih.php>. (Дата обращения: 10.03.2024).