Управление образования администрации муниципального образования
«город Северобайкальск»
Муниципальное автономное образовательное учреждение дополнительного образования «Дом детского творчества «Эврика»

Рассмотрена на заседании
педагогического совета
Протокол № 1
от «31» 08 2026 года

УТВЕРЖДАЮ
И.о. директора МАОУ ДО «ДДТ «Эврика»
_____ В.В. Авдюшкина

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА

технической направленности

«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»

(IT-куб)

Возраст обучающихся: 15 – 17 лет
Срок реализации программы: один год

Автор-составитель программы:
Фролова Анастасия Денисовна,
педагог дополнительного образования

г. Северобайкальск, 2026

Пояснительная записка

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» носит практикоориентированный характер и направлена на расширение базовых знаний обучающихся в области алгоритмизации, логики и информационных технологий. Основное внимание в курсе уделяется не просто освоению готовых информационных технологий, а обучению основам их создания. Изучение алгоритмизации позволяет перевести теоретические знания учащихся в плоскость конкретной практической деятельности, развивая системное и критическое мышление. Данный курс является вводным курсом по программированию, дающий представление о базовых понятиях структурного программирования (данных, операциях, переменных, ветвлениях в программе, циклах и функциях).

Выбор Python обусловлен тем, что это язык, обладающий рядом преимуществ перед другими языками для начинающих изучать программирование – ясность кода, быстрота реализации. Python – самый изучаемый и один из самых востребованных на рынке труда. Изучение Python откроет обучающимся возможности дальнейшего развития в области IT и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решении заданий ЕГЭ. Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» имеет техническую направленность.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» разработана на основе следующих нормативно-правовых документов:

– Федеральный закон Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;

– Приказ Министерства просвещения РФ от 27 июля 2022 г. N 629 “Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам”;

– Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 04.09.2014 г. №1726-р);

– Стратегия развития воспитания в Российской Федерации на период до 2025 года (утверждена Распоряжением Правительства Российской Федерации от 29.05.2015 г. №996-р);

– Письмо Минобрнауки России от 18.11.2015 №09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы);

– Методические рекомендации по реализации основных образовательных программ, дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий (под ред. А. А. Бучек [и др.]; БелИРО. – Белгород: БелИРО, 2020).

Актуальность и педагогическая целесообразность программы

Актуальность программы обусловлена широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём на уровне начинающего программиста.

Изучение основных принципов программирование невозможно без регулярной практики написания программ на каком-либо языке программирования. Для обучения в рамках программы выбран язык Python, который является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

Синтаксис языка достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на алгоритмических аспектах программирования.

Отличительная особенность программы, новизна

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Новизна программы заключается в том, что Python дает более широкие возможности в области программирования. На языке Python можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается.

Принципы учебного процесса

- самостоятельность – обучающиеся чувствуют себя полноправными субъектами учебно-воспитательного процесса, то есть участвуют в целеполагании своей деятельности.
- самоорганизация – педагог не учит, а помогает учиться, обуславливая необходимость обучения учащихся умения и навыками рационального учения.
- ответственность – развивает социальную зрелость личности
- психологическое удовлетворение – подразумевает эмоциональное удовлетворение каждого ученика и тем самым мотивации учения.
- развитие – ориентировка на зону ближайшего развития с учетом актуального уровня развития; развитие потребности в преодолении посильных трудностей.

Объём и срок освоения программы.

Объем программы – 72 часа.

Программа рассчитана на 1 год обучения.

Форма обучения.

Очная

Особенности организации образовательного процесса.

Программа «Программирование на языке Python» рассчитана на детей 15–17 лет.

Количество человек в группе: 12 – 15.

Режим занятий.

Занятия проводятся два раза в неделю по 1 академическому часу (40 минут).

Занятия проводятся по группам и индивидуально.

Цели и задачи

Цель программы: способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов,

умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Задачи программы:

Образовательные (предметные):

- познакомить с основными алгоритмическими конструкциями и правилами их записи, с основными способами организации данных;
- научить составлять и записывать алгоритмы с использованием соответствующих алгоритмических конструкций;
- научить распознавать необходимость применения той или иной алгоритмической конструкции при решении задач;
- научить организовывать данные для эффективной алгоритмической обработки;
- научить разрабатывать алгоритмы и реализовывать их на языке программирования Python;
- научить осуществлять отладку и тестирование программы.

Личностные:

- повысить общекультурный уровень обучающихся;
- научить правильным методологическим подходом к познавательной и практической деятельности;
- привить навыки сознательного и рационального использования компьютера в своей учебной, а затем и профессиональной деятельности;
- воспитание трудолюбия, инициативности и настойчивости в преодолении трудностей.

Метапредметные:

- развить умения организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- развить умение планировать, определять последовательность промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разрабатывать последовательность и структуру действий, необходимых для достижения цели при помощи фиксированного набора средств.

Содержание программы

Учебный план

№ п/п	Тема занятий	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
1.	**Раздел 1. Знакомство с языком Python**	**6**	**3**	**3**	Тестирование, решение практических задач
1.1	Общие сведения о языке Практическая работа: Установка программы Python	2	1	1	
1.2.	Режимы работы Практическая работа: Режимы работы с Python	4	2	2	
2.	**Переменные выражения**	**10**	**5**	**5**	Тестирование, решение практических задач
2.1	Переменные Практическая работа: Работа со справочной системой	2	1	1	
2.2	Выражения Практическая работа: Выражения	4	2	2	
2.3	Ввод и вывод	2	1	1	
2.4	Задачи на элементарные действия с числами Практическая работа Задачи на элементарные действия с числами	2	1	1	
3	**Условные предложения**	**10**	**5**	**6**	Тестирование, Решение практических задач
3.1	Логические выражения и операторы. Практическая работа: Логические выражения	4	1	3	
3.2	Условный оператор Практическая работа: «Условный оператор»	2	2	2	
3.3	Множественное ветвление Практическая работа: Множественное ветвление Практическая работа: «Условные операторы»	4	2	2	
4	**Раздел 4. Циклы**	**10**	**5**	**5**	Тестирование, решение практических задач, творческая работа
4.1.	Оператор цикла с условием Практическая работа "Числа Фибоначчи"	2	1	1	
4.2.	Оператор цикла for Практическая работа Решение задачи с циклом for.	4	2	2	
4.3.	Вложенные циклы Практическая работа: Реализация циклических алгоритмов	2	1	1	
4.4.	Случайные числа Практическая работа: Случайные числа	2	1	1	
5	**Раздел 5. Функции**	**10**	**4**	**6**	Тестирование, решение практических задач
5.1.	Создание функций Практическая работа Создание функций	2	1	1	

5.2.	Локальные переменные Практическая работа Локальные переменные	2	1	1	
5.3.	Примеры решения задач с использованием функций Практическая работа Решение задач с использованием функций	2	1	1	
5.4	Рекурсивные функции Практическая работа Рекурсивные функции	4	1	3	
6	**Раздел 6. Строки - последовательности символов**	**8**	**3**	**5**	
6.1.	Строки Практическая работа: Строки	2	1	1	Решение практических задач
6.2.	Срезы строк	2	1	1	
6.3	Примеры решения задач с остроками Практическая работа: Решение задач с остроками.	4	1	3	
7	**Раздел 7. Сложные типы данных**	**12**	**6**	**6**	
7.1.	Списки Практическая работа: Списки. Срезы списков	2	1	1	
7.3	Списки: примеры решения задач Практическая работа: Решение задач со списками	2	1	1	Тестирование, Решение практических задач
7.4	Матрицы	2	1	1	
7.5	Кортежи	2	1	1	
7.6	Введение в словари	2	1	1	
7.7	Множества в языкеPython	2	1	1	
8	**Раздел 8. Стиль программирования и отладка программ**	**6**	**3**	**3**	
8.1	Стиль программирования	2	1	1	Решение практических задач, презентация проекта, рефлексия.
8.2	Отладка программ	2	1	1	
8.3	Зачет по курсу «Программирование на языке Python»	2	1	1	
	ВСЕГО	**72**	**34**	**38**	

Содержание учебного плана

Вводное занятие.

Теория: Знакомство с планом работы объединения, инструктаж по ТБ. Практика: Опрос. Тренинг на командообразование.

Раздел 1. Знакомство с языком Python.

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Первая программа. Структура программы на языке Python. Комментарии.

Практическая работа 1.1. Установка программы Python Практическая работа 1.2. Режимы работы с Python Тест №1. Знакомство с языком Python

Учащиеся должны знать/ понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.

Раздел 2. Переменные и выражения (10 часов)

Теория: Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Ввод и вывод. Ввод данных с клавиатуры. Вывод данных на экран. Пример скрипта, использующего ввод и вывод данных. Задачи на элементарные действия с числами. Решение задач на элементарные действия с числами.

Практическая работа 2.1. Работа со справочной системой Практическая работа 2.2. Переменные

Практическая работа 2.3. Выражения

Практическая работа 2.5. Задачи на элементарные действия с числами

Тест №2. Выражения и операции.

Учащиеся должны знать/ понимать:

- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Раздел 3. Условные предложения.

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практическая работа 3.1. Логические выражения Практическая работа 3.2. "Условный оператор" Практическая работа 3.3. Множественное ветвление Практическая работа 3.4. "Условные операторы"

Самостоятельная работа №1. Решение задач по теме "Условные операторы". Зачетная работа №1. "Составление программ с ветвлением".

Тест №3."Условные операторы".

Учащиеся должны знать/ понимать:

- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- Учащиеся должны уметь:
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов.

Раздел 4. Циклы

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов.

Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практическая работа 4.1. "Числа Фибоначчи" Практическая работа 4.2. Решение задачи с циклом for.

Практическая работа 4.3. Реализация циклических алгоритмов Практическая работа 4.4. Случайные числа

Практическая работа 4.5. Решение задач с циклом. Самостоятельная работа №2 "Составление программ с циклом"

Тест №4. Циклы

Творческая работа № 1."Циклы"

Учащиеся должны знать/ понимать:

- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.

Учащиеся должны уметь:

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использовать цикл с параметром для решения поставленной задачи.

Раздел 5. Функции

Теория: Создание функций. Параметры и аргументы. Локальные и глобальные переменные.

Поток выполнения. Функции, возвращающие результат. Анонимные функции, инструкция lambda. Примеры решения задач с использованием функций.

Рекурсивные функции. Вычисление факториала. Числа Фибоначчи. Практическая работа 5.1. Создание функций

Практическая работа 5.2. Локальные переменные

Практическая работа 5.3. Решение задач с использованием функций. Практическая работа 5.4. Рекурсивные функции

Самостоятельная работа №3 по теме "Функции" Тест №5. Функции

Учащиеся должны знать/ понимать:

- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.

Учащиеся должны уметь:

- создавать и использовать функции;
- использовать механизм параметров для передачи значений.

Раздел 6. Строки- последовательности символов

Теория: Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки.

Срезы строк. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи). Примеры решения задач со строками.

Практическая работа 6.1. Строки

Практическая работа 6.2. Решение задач со строками.

Учащиеся должны знать/ понимать:

- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками.

Учащиеся должны уметь:

- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке.

Раздел 7. Сложные типы данных

Теория: Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Списки: примеры решения задач.

Матрицы. Вложенные списки. Матрицы. Строки и списки. Генераторы списков в Python. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения

Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы.

Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение. Оператор определения принадлежности элемента множеству.

Практическая работа 7.1. Списки.

Практическая работа 7.2. Решение задач со списками. Тест №7. Списки

Учащиеся должны знать/ понимать:

- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.

Учащиеся должны уметь:

- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.

Раздел 8. Стиль программирования и отладка программ

Стиль программирования. Отладка программ.

Зачет по курсу «Программирование на языке Python»

Учащиеся должны знать/ понимать:

- что такое стиль программирования;

- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- определять вид ошибок и находить ошибки в программе.
- выполнять тестирование и отладку программ.

Планируемые результаты

Реализация программы предполагает достижение обучающимися следующих личностных, метапредметных и предметных результатов:

Предметные:

- умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;
- формирование представлений об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ;
- умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;
- умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;
- формирование умения соблюдать нормы информационной этики и права.

Метапредметные:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно- исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;

- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно- исследовательской деятельности.

Личностные:

- формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;
- развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно- исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;
- формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;
- формирование ценности здорового и безопасного образа жизни;
- усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Условия реализации программы

Для организации работы центра «IT-Куб» в распоряжении «Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования «IT-Куб» рекомендуется следующее оборудование лаборатории:

- ноутбук — рабочее место преподавателя;
- рабочее место обучающегося;
- жёсткая, неотключаемая клавиатура: наличие;
- русская раскладка клавиатуры: наличие;
- диагональ экрана: не менее 15,6 дюймов;
- разрешение экрана: не менее 1920/1080 пикселей;
- количество ядер процессора: не менее 4;
- количество потоков: не менее 8;
- базовая тактовая частота процессора: не менее 1 ГГц;
- максимальная тактовая частота процессора: не менее 2,5 ГГц;
- объём установленной оперативной памяти: не менее 8 Гбайт;
- объём накопителя SSD: не менее 240 Гбайт;
- внешний интерфейс USB стандарта не ниже 3.0: не менее трёх свободных;
- внешний интерфейс LAN (использование переходников не предусмотрено): наличие;
- наличие модулей и интерфейсов (использование переходников не предусмотрено): VGA, HDMI;
- беспроводная связь Wi-Fi: наличие с поддержкой стандарта IEEE 802.11n или современнее;
- веб-камера: наличие;
- манипулятор «мышь»: наличие;
- предустановленная операционная система с графическим пользовательским интерфейсом, обеспечивающая работу распространённых образовательных и общеси-

стемных приложений: наличие), МФУ, веб-камера, интерактивный моноблочный дисплей.

ВОСПИТАТЕЛЬНЫЙ МОДУЛЬ

Цель и задачи воспитательной программы

Цель: сформировать у обучающихся культуру ответственного цифрового творчества, алгоритмическое и критическое мышление, навыки командной разработки и осознанное отношение к этическим и правовым аспектам использования программного кода; поддержать профессиональное самоопределение в IT-сфере.

Задачи:

- воспитать трудолюбие, усидчивость, умение доводить программный проект до работоспособного состояния;
- развивать навыки сотрудничества: распределение ролей в команде, взаимная проверка кода (code review), конструктивная обратная связь;
- формировать цифровую и правовую культуру: понимание лицензий на код, этики публикации решений, правил работы с данными;
- тренировать саморегуляцию и стрессоустойчивость при отладке и разборе ошибок (понимание, что ошибка — это часть процесса);
- поддерживать креативность и исследовательский подход: поиск разных способов решения одной задачи, сравнение эффективности;
- содействовать профориентации через знакомство с реальными ролями в разработке (разработчик, тестировщик, аналитик) и практиками индустрии.

Направления воспитания и их интеграция в занятия

Направление	Как реализуется в содержании программы
Трудовое и проектное воспитание	Работа над проектом как продуктом: планирование спринтов, чек-листы задач, контроль качества кода, тестирование, демонстрация результата.
Социально-коммуникативное	Парное программирование, взаимная проверка решений, разбор конфликтов в команде, правила уважительной критики кода.
Духовно-нравственное и правовое	Этические и правовые нормы: лицензии на код, правила публикации, работа с персональными данными, недопустимость плагиата.
Интеллектуальное развитие	Алгоритмическое мышление, анализ сложности, сравнение подходов, рефлексия «почему это решение работает/не работает».
Профориентационное	Знакомство с ролями в команде разработки, гостевые лекции, разбор реальных задач, мини-пробы (тестирование, рефакторинг, документирование).
Личностное развитие и саморегуляция	Техники тайм-менеджмента для задач по программированию, разбор «стресса отладки», управление фрустрацией при ошибках.

Календарный план воспитательной работы

Месяц	Мероприятие	Направление	Форма
Сентябрь	Вводное собрание, инструктаж по ТБ, игра «Безопасный кабинет»	Безопасность, дисциплина	Беседа + игра

Октябрь	«Этика разработчика»: разбор кейсов по лицензиям и публикации кода	Правовое, духовно-нравственное	Кейс-метод, дискуссия
Ноябрь	Парное программирование: мини-хакатон на 2 часа	Социально-коммуникативное, трудовое	Практикум + взаимная оценка
Декабрь	Разбор ошибок: «Почему код не работает и что с этим делать»	Личностное развитие, интеллектуальное	Практикум, разбор типичных ошибок
Январь	Мастер-класс «Тестирование без стресса»: pytest и автотесты	Качество, саморегуляция	Практикум, мини-конкурс на лучший тест
Февраль	Гостевое занятие: встреча с разработчиком/ инженером	Профориентация, мотивация	Интервью
Март	Внутригрупповой code review: «Что можно улучшить в этом коде»	Социально-коммуникативное, профессиональное	Разбор решений, правило «бутерброда»
Апрель	Подготовка к демо: оформление README, скриншоты, описание архитектуры	Проектное, трудовое	Практикум, чек-листы, командная работа
Май	Итоговый демо-день: защита проектов, рефлексия	Личностное развитие, профориентация	Демо, эссе «Мой путь в разработке»

Критерии и способы оценки воспитательных результатов

- **Наблюдение педагога:** дисциплина, соблюдение сроков, участие в обсуждениях, помощь товарищам (чек-листы по каждому этапу).
- **Рефлексия:** короткие письменные ответы «Что было самым сложным?», «Как я справился?», «За что я благодарен команде?».
- **Портфолио проекта:** репозиторий с кодом, тесты, README, скриншоты результатов, описание использованных библиотек и лицензий.
- **Оценка групповой работы:** вклад каждого участника (самооценка, оценка команды, оценка педагога) по критериям: «соблюдал сроки», «помогал другим», «предлагал решения».
- **Публичная защита:** ясность изложения, умение отвечать на вопросы, готовность к доработке.

Электронные ресурсы

Сайт/справочные материалы [Электронный ресурс] − Режим доступа: https://metanit.com/python/, свободный.

Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] − Режим доступа: http://pythontutor.ru/, свободный.

Сайт/Адаптивный тренажер Python [Электронный ресурс] − Режим доступа: https://stepik.org/course/431 , свободный.

Сайт / среда разработки для языка Python [Электронный ресурс] − Режим доступа: https://www.jetbrains.com/pycharm/?fromMenu , свободный.

Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] − Режим доступа: openbookproject.net , свободный.

Официальный сайт программы [Электронный ресурс] − Режим доступа: https://docs.python.org/, свободный.

Сайт, среда разработки для языка Python. [Электронный ресурс] − Режим доступа: https://www.jetbrains.com/pycharm/?fromMenu, свободный.

Литература

Бизли, Дэвид М. Python. Подробный справочник. – М.–СПб.: Символ-Плюс, 2010.

Лутц, Марк Python. Справочник. – М.: Вильямс, 2015.