

Министерство образования и науки Хабаровского края
Краевое государственное автономное
нетиповое образовательное учреждение
«Краевой центр образования»

РАССМОТРЕНО
на заседании Педагогического совета
КГАНОУ «Краевой центр образования»
Протокол № 1
«27» августа 2024 г.

УТВЕРЖДАЮ
Генеральный директор КГАНОУ
«Краевой центр образования»
/П.С.Черемухин
«27» августа 2024 г.



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА «ПРОГРАММИРОВАНИЕ НА
ЯЗЫКЕ PYTHON»**

Направленность: техническая
Уровень освоения: стартовый
Возраст учащихся: 13-17 лет
Срок реализации: 1 год

Составитель:
Деркач Анатолий Александрович,
ПДО

Хабаровск
2024 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Программирование на языке Python» направлена на развитие технических и творческих способностей, обучающихся через обучение и применение технологий в программировании.

Нормативно-правовые документы:

- Федеральный закон от 29.12.2012 г. № 273-ФЗ «Об образовании в Российской Федерации»;
- Приказ Министерства просвещения Российской Федерации от 27 июля 2022 г. № 629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Распоряжение Правительства РФ от 31.03.2022 г. № 678-р «Концепция развития дополнительного образования детей до 2030 г.»;
- Постановление Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28 Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Концепция персонифицированного дополнительного образования детей в Хабаровском крае от 05.08.2019 г.;
- Правила ПФ Министерства образования и науки Хабаровского края от 26.09.2019 г. №1321;
- Приказ КГАОУ ДО РМЦ №383П от 26.09.2019 г. об утверждении Положения о дополнительной общеобразовательной программе в Хабаровском крае;
- Устав краевого государственного автономного нетипового образовательного учреждения «Краевой центр образования».

Актуальность

- **Востребованность на рынке труда:** Python - один из самых популярных языков программирования, применяющийся в различных областях, от веб-разработки до анализа данных и машинного обучения.
- **Универсальность:** Python легко изучить, он обладает понятным синтаксисом и богатой экосистемой библиотек.
- **Развитие цифровых компетенций:** Знание Python позволяет учащимся развить логическое мышление, алгоритмизацию, креативность и навыки решения проблем, которые необходимы в современном мире.

Педагогическая целесообразность

- **Повышение мотивации к изучению информатики:** Занятия программированием на Python делают изучение информатики более интересным и практическим, что повышает мотивацию учащихся.
- **Развитие навыков самостоятельной работы:** Python способствует развитию навыков самостоятельного решения задач, анализа и структурирования информации, что пригодится в будущей профессиональной деятельности.
- **Интеграция с другими дисциплинами:** Python может быть использован для решения задач в различных предметах, например, математике, физике, химии, что делает обучение более комплексным и интересным.

Цель программы: развитие алгоритмического мышления и усвоение навыков программирования.

- **Предметные:**
 - Овладеть базовыми знаниями и навыками программирования на языке Python.
 - Научиться создавать простые программы для решения различных задач.
 - Изучить основные алгоритмы и структуры данных.
 - Познакомиться с различными модулями и библиотеками Python.
- **Метапредметные:**
 - Развить навыки логического мышления, алгоритмизации, анализа и синтеза информации.
 - Повысить уровень самостоятельной работы и креативности.
 - Научиться работать в команде и эффективно решать проблемы.
- **Личностные:**
 - Повысить интерес к изучению информатики и программирования.

- Развить навыки самообучения и самостоятельного поиска информации.
- Формировать ответственное отношение к профессиональной деятельности.

Адресат программы: дети 12-16 лет.

Наполняемость группы: 8-12 человек.

Форма реализации программы: очная, с применением дистанционных образовательных технологий.

Объем, режим и сроки реализации программы.

Период	Продолжительность занятия	Кол-во занятий в неделю	Кол-во часов за неделю	Кол-во недель	Кол-во часов всего
1 год	2 ч.	2	4 ч.	36ч.	144 ч.

УЧЕБНЫЙ ПЛАН

№ п/п	Разделы	Количество часов			Форма контроля
		Всего	Теория	Практика	
1.	Введение в программирование	12	11	1	Тест 1 (приложение 2)
2.	Основы языка Python	36	6	30	Тест 2 (Приложение 3)
3.	Алгоритмы и структуры данных	42	8	34	Тест 3 (приложение 4)
4.	Объектно-ориентированное программирование	24	4	20	Тест 4 (приложение 5)
5.	Практика программирования	22	2	20	Мини-проект
6.	Проектная деятельность	8	2	6	Презентация проекта
	ВСЕГО:	144	33	111	

СОДЕРЖАНИЕ

1. Введение в программирование (12ч)

Знакомство с программой, ее содержанием, результатами. Знакомство с группой. Техника безопасности. Расписание занятий.

Теория: Знакомство с понятием Программирование.

Практика: Установка языка и программ.

2. Основы языка Python(36ч)

Теория: Изучение типов данных в языке Python. Знакомство со всеми типами операторов. Работа с циклами for и while. Работа с функциями и основы модульного программирования. Работа со строками, ввод и вывод данных.

Практика: Реализация программ использующие разные типы данных. Решение задач с использованием операторов и циклов. Создание собственных функций и работа с ними

Практическая работа № 1. Создание меню программы.

3. Алгоритмы и структуры данных (42ч).

Теория: Введение в алгоритмы. Основные алгоритмы сортировки и поиска. Изучение структуры данных.

Практика: Реализация алгоритмов сортировки на языке Python. Работа с различными структурами данных, создание собственных структур. Решение задач по программированию, где необходимо использовать алгоритмы и структуры данных.

4. Объектно-ориентированное программирование (24ч).

Теория: Введение в ООП: классы, объекты, атрибуты и методы. Наследование классов. Полиморфизм. Абстрактные классы.

Практика: Создание классов и объектов, использование атрибутов и методов. Наследование классов, создание подклассов. Решение задач с использованием ООП.

5. Практика программирования (22ч).

Теория: Работа с библиотеками стандартной библиотеки Python (`math`, `random`, `datetime`). Работа со сторонними библиотеками и фреймворками.

Работа с модулями и создание своих полностью модульных программ.

Практика:

6. Проектная деятельность (20ч).

Теория: Этапы разработки программного проекта. Планирование и управление проектом. Документирование кода. Тестирование программ.

Практика: Выбор темы для группового проекта. Разработка и реализация проекта. Презентация проекта.

Подготовка и участие в конкурсах. Примерный список конкурсов: «Краевой фестиваль научно-технического творчества», «Взгляд в будущее», «Хакатон» и др.

Планируемые результаты.

Предметные:

- Учащиеся смогут описать основные концепции программирования (алгоритмы, переменные, типы данных, операторы, циклы, функции).
- Учащиеся смогут писать простые программы на языке Python, используя базовые синтаксические конструкции.
- Учащиеся смогут решать задачи по программированию, используя изученные алгоритмы и структуры данных.
- Учащиеся смогут использовать библиотеки стандартной библиотеки Python для решения задач.
- Учащиеся смогут объяснять основы объектно-ориентированного программирования.
- Учащиеся смогут создавать простые программы с использованием объектно-ориентированного подхода.

Метапредметные:

- **Познавательные:**
 - Развитие навыков алгоритмического мышления.
 - Формирование умения анализировать задачи и находить решения.
 - Развитие навыков поиска и обработки информации.
 - Формирование навыков самостоятельной работы.
- **Коммуникативные:**
 - Развитие навыков устной и письменной речи при описании алгоритмов и программ.
 - Формирование умения работать в команде при выполнении групповых проектов.
 - Развитие навыков критического мышления и аргументации.
- **Регулятивные:**
 - Развитие умения планировать свою работу и контролировать ее выполнение.
 - Формирование умения оценивать свою работу и работу других.
 - Развитие умения корректировать свою работу.

Личностные:

- Развитие интереса к информационным технологиям.
- Повышение уровня информационной культуры.
- Формирование навыков самообучения.
- Развитие творческого потенциала.
- Формирование умения работать в команде.
- Развитие ответственности за свою работу.

**Комплекс организационно-педагогических условий
Информационно-методическое обеспечение:**

- "Python для детей: программирование игр и анимации" by Jason R. Briggs (<https://www.amazon.com/Python-Kids-Programming-Games-Animations/dp/1783558184>): Отличный учебник, который подходит для начинающих и делает обучение программированию веселым.
- "Изучаем Python" by Mark Lutz, David Ascher (<https://www.amazon.com/Learning-Python-5th-Mark-Lutz/dp/1449355730>): Более глубокий учебник для продвинутых изучающих Python.
- "Think Python: How to Think Like a Computer Scientist" by Allen Downey (<https://greenteapress.com/thinkpython/>): Бесплатный учебник, который фокусируется на принципах программирования и решении задач.
- "Learn Python 3" (<https://realpython.com/python-beginner-tutorial/>): Популярный курс, который охватывает все основы Python.
- "Python for Beginners" (<https://www.udemy.com/course/complete-python-bootcamp/>): Полный курс, который включает практические проекты и упражнения.
- "Codecademy Python" (<https://www.codecademy.com/learn/learn-python-3>): Интерактивный курс, который позволяет получить практические навыки программирования.
- "Python документация" (<https://docs.python.org/3/>): Официальный ресурс с описанием языка, библиотек и примеров кода.
- "Python Tutor" (<http://pythontutor.com/visualize.html#mode=edit>): Визуализатор кода Python, который помогает понять, как работают программы.
- "Python для начинающих" (https://www.youtube.com/playlist?list=PLj8W7-M9Y89Kq6fF7m5uW02H0K3e_c84f): Плейлист с уроками по Python от "PyDev" на YouTube.

- "Программирование на Python"
(<https://www.youtube.com/playlist?list=PL2O3akYVOu6lX-KJ4X753bF2fZ1V0R1Y8>): Плейлист с уроками от "CodeBasics" на YouTube.
- "Coding for Beginners with Python"
(https://www.youtube.com/playlist?list=PLZPZq0r_IX0U-6p_e5wL-J1R1lJ9F9v7c): Плейлист с интерактивными уроками от "Socratica" на YouTube.

Методы обучения:

- **Лекции:** Изучение теоретических основ программирования на Python.
- **Практические занятия:** Решение задач, выполнение лабораторных работ, создание простых программ.
- **Самостоятельная работа:** Выполнение домашних заданий, изучение материалов, поиск информации в интернете.
- **Проектная работа:** Разработка небольших проектов, демонстрирующих полученные знания и навыки.

Материально-техническое обеспечение:

- Проектор
- Компьютеры 12 шт
- Устройства ввода (компьютерная мышь)

Информационно-методическое обеспечение:

- ПО Python 3.12
- ПО PyCharm Community Edition 2024
- Microsoft EDGE
- ОС Windows 10

Формы контроля и оценки результатов:

- **Текущий контроль:** Проверка выполнения заданий на практических занятиях, домашние работы.
- **Промежуточная аттестация:** Контрольные работы по пройденному материалу.

- **Итоговая аттестация:** Сдача и защита итогового проекта.

КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

Дата	Тема	Кол-во часов
	Введение. Техника безопасности.	2
	Что такое программирование.	2
	Языки программирования. Алгоритмы.	2
	Типы данных, операторы.	2
	Установка Python и его первичная настройка.	2
	Знакомство с IDE PyCharm и его первичная настройка.	2
	Базовый синтаксис.	2
	Ввод-вывод данных. (input(),print())	2
	Переменные, константы и их типы данных.	2
	Операторы.	2
	Арифметические операторы.	2
	Оператор сравнения.	2
	Логический оператор.	2
	Самостоятельная работа: Решение задач на ввод и вывод данных.	2
	Условные операторы. Логические выражения и условия.	2
	Оператор if, else, elif.	2
	Цикл for.	2
	Цикл while.	2
	Вложенные условные операторы.	2
	Вложенные циклы.	2
	Самостоятельная работа: Решение задач с использованием условным операторов и циклов.	2
	Списки.	2
	Определение списков, доступ к элементам.	2
	Методы работы со списками.	2
	Одномерные массивы.	2
	Самостоятельная работа: Решение задач с использованием списков.	2
	Строки.	2

	Определение строк и доступ к элементам.	2
	Методы работы со строками.	2
	Форматирование строк.	2
	Самостоятельная работа: Решение задач с использованием строк.	2
	Функции.	2
	Определение и вызов функций.	2
	Передача аргументов в функции.	2
	Возвращение значений из функции.	2
	Самостоятельная работа: Решение задач с использованием функций.	2
	Модули и библиотеки.	2
	Импорт модулей.	2
	Использование стандартных библиотек Python: math, random, datetime, os.	2
	Использование библиотек Python: matplotlib, pandas.	2
	Самостоятельная работа: Решение задач с использованием библиотек и модулей.	2
	Работа с файлами.	2
	Открытие и закрытие файлов.	2
	Чтение и запись данных в файл.	2
	Самостоятельная работа: Решение задач с использованием файлов.	2
	Введение в ООП.	2
	Понятие класса и объекта. Их определение.	2
	Атрибуты методов класса.	2
	Конструктор и деструктор.	2
	Самостоятельная работа: Решение задач с использованием классов.	2
	Наследование и полиморфизм.	2
	Наследование классов.	2
	Полиморфизм.	2
	Самостоятельная работа: Решение задач с использованием наследования классов и полиморфизма.	2
	Работа с модулями.	2
	Создание собственных модулей. Импорт модулей.	2
	Самостоятельная работа: Решение задач с	2

	использованием собственных модулей.	
	Понятие исключения.	2
	Обработка исключений: try, except, else, finally.	2
	Самостоятельная работа: Решение задач с использованием исключений.	2
	Основы проектной деятельности.	2
	Работа над творческим проектом	22
	Защита творческого проекта	2
	Итого	144

Список источников

Учебники:

- "Python для чайников" Марка Лутца
- "Изучаем Python" Эрика Мэттиса

Онлайн-ресурсы:

- Python.org
- Codecademy
- freeCodeCamp

Библиотеки:

- Tkinter
- NumPy
- Pandas
- Matplotlib

Диагностическая карта оценки результатов усвоения программы

Каждый критерий оценивается в баллах от 1 до 10.

Высокий уровень усвоения – 80% и более

Средний – 60-80%

Низкий - до 60 %

ФИО ребенка						
Критерий оценки						
Умение работать в PyCharm						
Умение работать с языком Python						
Умеют находить в интернете доступные методы решения задач в программировании						
Знание, что такое проект и его структурные элементы						
Умение работать в команде: слышать и слушать друг друга принимать чужое мнение распределять обязанности в группе, нести ответственность за общий результат, находить компромиссы в конфликтных ситуациях.						
Оценки итогового продукта						

Тест 1 к разделу 2 (Введение в программирование)

1. Что из перечисленного является правильным именем переменной в Python? а) 123abc б) abc_123 в) abc-123 д) abc 123
2. Какой знак используется для присваивания значения переменной? а) = б) == в) + д) -
3. Какой оператор используется для проверки равенства? а) = б) == в) + д) -
4. Какой оператор используется для проверки неравенства? а) != б) == в) < д) >
5. Какой цикл используется для повторения блока кода определенное количество раз? а) while б) for в) if д) else
6. Какой цикл используется для повторения блока кода, пока условие не станет ложным? а) while б) for в) if д) else
7. Какой оператор используется для выполнения блока кода только в том случае, если условие истинно? а) while б) for в) if д) else
8. Какой оператор используется для выполнения блока кода только в том случае, если условие ложно? а) while б) for в) if д) else
9. Какое из выражений является правильным синтаксисом для вывода текста в консоль? а) print("Привет, мир!") б) write("Привет, мир!") в) display("Привет, мир!") д) show("Привет, мир!")
10. Какое из выражений является правильным синтаксисом для получения ввода от пользователя? а) input() б) getinput() в) readInput() д) receive_input()

Тест 2 к разделу 3 (Основы языка Python)

1. Что такое функция? а) Блок кода, который выполняется только один раз. б) Блок кода, который выполняется только при определенных условиях. в) Блок кода, который можно вызывать многократно. г) Блок кода, который не выполняется.
2. Как называется блок кода, который определяет функцию? а) `main` б) `function` в) `def` г) `loop`
3. Как вызывается функция? а) Вводя ее имя в консоль. б) Используя оператор `call`. в) Используя оператор `execute`. г) Вводя ее имя и скобки.
4. Как передать данные в функцию? а) Используя оператор `give`. б) Используя оператор `send`. в) Используя оператор `pass`. г) Используя аргументы в скобках.
5. Как функция возвращает результат? а) Используя оператор `return`. б) Используя оператор `send`. в) Используя оператор `give`. г) Используя оператор `result`.
6. Что такое аргумент функции? а) Значение, которое функция возвращает. б) Значение, которое функция получает. в) Имя функции. г) Тип данных, который функция обрабатывает.
7. Что такое значение, которое функция возвращает? а) Аргумент функции. б) Результат работы функции. в) Имя функции. г) Тип данных, который функция обрабатывает.
8. Как создать функцию, которая не возвращает результат? а) Используя оператор `void`. б) Не используя оператор `return`. в) Используя оператор `none`. г) Используя оператор `empty`.
9. Какое из выражений является правильным синтаксисом для определения функции? а) `function myfunction():` б) `def myfunction():` в) `main myfunction():` г) `loop myfunction():`
10. Какое из выражений является правильным синтаксисом для вызова функции? а) `myfunction()` б) `call myfunction()` в) `execute myfunction()` г) `run myfunction()`

Тест 3 к разделу 4 (Алгоритмы и структуры данных)

1. Какой тип данных используется для хранения целых чисел? а) int б) float в) str д) bool
2. Какой тип данных используется для хранения десятичных чисел? а) int б) float в) str д) bool
3. Какой тип данных используется для хранения текста? а) int б) float в) str д) bool
4. Какой тип данных используется для хранения логических значений (True/False)? а) int б) float в) str д) bool
5. Как проверить, является ли переменная строкой? а) `isinstance(variable, str)` б) `type(variable) == str` в) `variable.is_string()` д) All of the above
6. Как преобразовать строку в целое число? а) `int(string)` б) `str(int)` в) `string.toInt()` г) `int.fromstring(string)`
7. Как преобразовать целое число в строку? а) `str(int)` б) `int(str)` в) `int.toString()` г) `str.fromint(int)`
8. Как создать список? а) `list()` б) `array()` в) `[]` г) All of the above
9. Как добавить элемент в список? а) `append()` б) `insert()` в) `add()` г) Both a and b
10. Как удалить элемент из списка? а) `remove()` б) `delete()` в) `pop()` г) Both a and c

Тест 4 к разделу 5 (Объектно-ориентированное программирование)

1. Что такое класс? а) Шаблон для создания объектов. б) Объект, который можно использовать в программе. в) Блок кода, который выполняется только один раз. г) Блок кода, который выполняется только при определенных условиях.
2. Что такое объект? а) Шаблон для создания классов. б) Экземпляр класса. в) Блок кода, который выполняется только один раз. г) Блок кода, который выполняется только при определенных условиях.
3. Как создать объект? а) Используя оператор "create". б) Используя оператор "new". в) Используя оператор "object". г) Используя оператор "instance".
4. Что такое атрибут класса? а) Переменная, которая хранит данные об объекте. б) Функция, которая выполняет действия с объектом. в) Имя класса. г) Тип данных, который класс обрабатывает.
5. Что такое метод класса? а) Переменная, которая хранит данные об объекте. б) Функция, которая выполняет действия с объектом. в) Имя класса. г) Тип данных, который класс обрабатывает.
6. Как обратиться к атрибуту объекта? а) Используя оператор "get". б) Используя оператор "access". в) Используя оператор ".". г) Используя оператор "[]".
7. Как вызвать метод объекта? а) Используя оператор "call". б) Используя оператор "execute". в) Используя оператор ".". г) Используя оператор "[]".
8. Что такое наследование в ООП? а) Возможность создавать новые классы на основе существующих. б) Возможность изменять существующие классы. в) Возможность удалять существующие классы. г) Возможность создавать новые объекты.
9. Что такое полиморфизм в ООП? а) Возможность использовать один и тот же код для объектов разных классов. б) Возможность изменять поведение объектов разных классов. в) Возможность создавать новые классы. г) Возможность создавать новые объекты.
10. Что такое инкапсуляция в ООП? а) Возможность скрыть данные и методы объекта от внешнего доступа. б) Возможность изменять данные и методы объекта. в) Возможность создавать новые классы. г) Возможность создавать новые объекты.

Ключи ответов:

Тема 1:	Тема 2:	Тема 3:	Тема 4:
1. б)	1. с)	1. а)	1. а)
2. а)	2. с)	2. б)	2. б)
3. б)	3. д)	3. б)	3. с)
4. а)	4. д)	4. а)	4. д)
5. б)	5. а)	5. б)	5. д)
6. а)	6. б)	6. с)	6. а)
7. с)	7. б)	7. с)	7. а)
8. д)	8. б)	8. а)	8. д)
9. а)	9. б)	9. а)	9. д)
10. а)	10. а)	10. а)	10. д)