

Министерство образования и занятости населения Приморского края
Краевое государственное автономное профессиональное образовательное
учреждение
«Владивостокский судостроительный колледж»

«УТВЕРЖДАЮ»
Директор КГА ПОУ «ВСК»
_____ Глушкова И.В.

«15» марта 2024 г.

Углублённое программирование на Python

Дополнительная общеобразовательная общеразвивающая программа
технической направленности

Возраст учащихся: 14-17 лет
Срок реализации программы: 1 год

_____,
Преподаватель

г.Владивосток
2024 г.

Раздел № 1. ОСНОВНЫЕ ХАРАКТЕРИСТИКИ ПРОГРАММЫ

1.1 Пояснительная записка

Актуальность программы

Программа «Углубленное программирование на Python» позволяет посредством формирования начальных навыков программирования подготовить платформу для изучения в дальнейшем более сложных языков программирования. Данный учебный курс позволит обучающемуся самостоятельно моделировать алгоритмические конструкции. Одним из средств достижения запланированных результатов является язык программирования «Python»

Особенности языка Python: язык очень легок в изучении; язык является объектно-ориентированным, поэтому его реализация является самой понятной; на Python можно написать любую программу; широкий спектр применения; Python работает во всех операционных системах, так как это интерпретируемый язык; Python распространяется свободно на условиях лицензии GNU 2.0. Данная лицензия разрешает бессрочно использовать Python на любом количестве компьютеров в любых целях без оформления каких-либо дополнительных документов. Поэтому целесообразно использовать этот язык при изучении программирования школьников. В связи с этим становится очевидным актуальность предлагаемой программы.

Программа предназначена для тех, кто обладает базовыми знаниями в области разработки, поэтому она подойдет тем, кто хочет расширить свои знания в области программирования.

Направленность программы: техническая.

Язык реализации программы: русский.

Уровень освоения: стартовый (ознакомительный).

Отличительные особенности: состоят в использовании следующих педагогических технологий:

- технология индивидуального обучения – проведение учебно-воспитательной работы с группой, ориентируясь на личностные особенности каждого обучающегося;
- технология коллективного взаимообучения – в процессе обучения

во время коллективных работ обучающиеся обмениваются опытом и знаниями, дополняя таким образом знания друг друга;

– технология модульного обучения – каждый модуль является самостоятельной, отдельной единицей, благодаря чему отслеживание усвоения программы отдельными учащимися значительно упрощается.

Адресат программы: школьники города Владивостока 14-17 лет.

Особенности организации образовательного процесса:

Для обучения на дополнительной общеобразовательной программе “Программирование на Python” не требуется начальных знаний, так как уровень освоения программы – “Стартовый (ознакомительный)”. Дети могут обучаться по данной программе без каких-либо знаний в этой области.

Поступление на дополнительную общеобразовательную программу “Программирование на Python” осуществляется на основании сертификата на право получения дополнительного образования.

- Режим занятий: занятия проводятся в группах от 8 до 12 человек, продолжительность одного занятия — 2 академических часа;

- Продолжительность образовательного процесса: общая продолжительность программы – 1 год, 72 часов, 2 часа в неделю.

1.2 Цель и задачи программы

Цель программы: формирование базовых навыков программирования у школьников города Владивостока 14-17 лет с помощью языка “Python”.

Задачи программы:

Воспитательные:

1. Сформировать способности к саморазвитию и самообразованию средствами информационных технологий на основе иллюстрированной среды программирования, мотивации к обучению и познанию;
2. Сформировать умения работать индивидуально и в группе для решения поставленной задачи;
3. Сформировать целостное мировоззрение, соответствующее современному уровню развития информационных технологий;
4. Сформировать осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности;
5. Обеспечить усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой;
6. Сформировать культуру начального программирования.

Развивающие:

1. Развить умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
2. Развить умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
3. Развить умение критически оценивать правильность решения учебно-исследовательской задачи;
4. Сформировать владение основами самоконтроля, способность к принятию решений;
5. Формировать мотивацию к профессиональному самоопределению учащихся.

Обучающие:

1. Знание основных алгоритмических структур и умение применять его для построения алгоритмов решения задач по их математическим моделям, умения и навыки составления простых программ по построенному алгоритму на Python;
2. Умения и навыки эффективного использования основных типов прикладных программ (приложений) общего назначения и информационных систем для решения с их помощью практических задач;
3. Умение грамотно интерпретировать результаты решения практических задач с помощью информационных технологий, применять полученные результаты в практической деятельности.

1.3 Содержание программы

Учебный план программы “Программирование на Python”

№ п/п	Наименование раздела	Количество часов			Форма аттестации
		Теория	Практика	Всего	
1	Среда программирования Python. Переменные	2	2	4	Беседа
2	Первые программы на языке Python, основные операторы	1	1	2	Устный опрос
3	Условный оператор if	2	2	4	Устный опрос
4	Циклы в языке Python	2	2	4	Устный опрос
5	Решение задач по темам: “Условный оператор и циклы”	0	4	4	Зачёт
6	Коллекции в языке Python	2	2	4	Устный опрос

7	Функции в языке Python	3	3	6	Устный опрос
8	Решение задач по пройденным темам	0	4	4	Зачёт
9	Введение в ООП. Основные понятия ООП	3	3	6	Устный опрос
10	HTTP запросы в Python	2	2	4	Устный опрос
11	Решение задач по пройденным темам	0	4	4	Зачёт
12	Django. Понятия и первое приложение	1	1	2	Устный опрос
13	Django. Шаблонизация и определение страниц	1	1	2	Устный опрос
14	Django. Модели объектов и миграции в БД	2	1	3	Устный опрос
15	Django. Связные модели	2	1	3	Устный опрос, тестирование
16	Django. Формы отправки	1	2	3	Устный опрос
17	Django. Администрирование приложения	2	1	3	Устный опрос

18	Django. Авторизация и разделение прав	1	1	2	Устный опрос
19	Подготовка проектов	0	6	6	Устный опрос, тестирование
20	Программирование на Python	0	2	2	Контрольная работа
Итого		27	45	72	

Содержание учебного плана

1. Тема: Среда программирования Python. Переменные.

Теория. Среда программирования в языке Python, основные инструменты среды, понятие «переменная».

Практика. Технические действия в среде программирования.

2. Тема: Первые программы на языке Python, основные операторы.

Практика. Простые программы на Python. Операторы присваивания, ввода/вывода данных, разработка программ.

3. Тема: Условный оператор if.

Теория. Условный оператор if в Python.

Практика. Разработка программ, реализующих условные алгоритмы.

4. Тема: Циклы в языке Python.

Теория. Операторы цикла for, while в Python.

Практика. Разработка программ, циклические алгоритмы.

5. Тема: Решение задач по темам: “Условный оператор и циклы”.

Практика. Решение дополнительных задач по темам “Условный оператор”, “Циклы”.

6. Тема: Коллекции в языке Python.

Теория. Понятие «коллекция» в языке Python. Виды коллекций

Практика. Разработка программ с использованием коллекций.

7. Тема: Функции в языке Python.

Теория. «Функция» в Python, описание функции, основные приёмы структурного программирования.

Практика. Обращение к функции в тексте программы, приёмы написания программ с использованием вспомогательных алгоритмов

8. Тема: Решение задач по изученным темам.

Практика. Решение дополнительных задач по темам «Списки в языке Python», «Функции в языке Python».

9. Тема: Введение в ООП. Основные понятия ООП.

Теория. Понятия объектно-ориентированного программирования, суть классов и объектов. Как создать взаимодействие между объектами. Главные парадигмы ООП.

Практика. Написание собственных классов и создание объектов

10. Тема: HTTP запросы в Python.

Теория. Изучение понятий HTTP Request, API, методов GET и POST. Разбор протоколов передачи данных, как передаются файлы, сообщения и потоки данных.

Практика. Написание различных запросов на API сервисы

11. Тема: Решение задач по изученным темам.

Практика. Решение дополнительных задач по темам «Введение в ООП. Основные понятия ООП», «HTTP запросы в Python».

12. Тема занятия: Django. Понятия и первое приложение.

Теория. Понятия фреймворка, веб-приложений и HTTP серверов. Как работает Django, какие инструменты он предоставляет.

Практика. Создание и настройка своего веб-приложения.

13. Тема: Django. Шаблонизация и определение страниц.

Теория. Изучение шаблонизатора Jinja2 и определение переходов по страницам. Способы отображения ответа при переходе по ссылке.

Практика. Подключение шаблонизатора к приложению, определение функций при переходах по страницам.

14. Тема: Django. Модели объектов и миграции в БД.

Теория. Понятие модели объекта, миграции свойств и параметров в базу данных. Какие существуют поля у модели и в каких случаях их нужно использовать.

Практика. Создание собственных моделей объектов и миграции их в базу данных.

15. Тема: Django. Связные модели.

Теория. Определение связной(реляционной) модели и какие способы их построения есть. Как модели связываются друг с другом. Зачем нужны промежуточные модели.

Практика. Создание связной модели с выводом её из базы на страницу.

16. Тема: Django. Формы отправки данных.

Теория. Что такое формы в веб интерфейсе, какие данные можно отправить и как их проверять. Как форму связать с моделью в базе данных.

Практика. Размещение формы на странице, проверка и отправка данных в БД.

17. Тема: Django.Администрирование приложения.

Теория. Рассмотрение приложения администрирования сайта. Что можно сделать в панели администрирования. Как её настраивать под себя.

Практика. Подключение и настройка приложения администрирования.

18. Тема: Django. Авторизация и разделение прав.

Теория. Определение понятия авторизации в системе. Что такое сессии в браузере, что такое токен авторизации и что такое права пользователя.

Практика. Создание и подключение формы для авторизации пользователя, разделение пользователей по правам доступа.

19. Тема: Подготовка проектов.

Практика. Разработка индивидуального или группового проекта на языке программирования Python.

20. Тема: Программирование на Python.

Практика. Защита индивидуальных или групповых проектов.

1.4 Планируемые результаты

Личностные результаты:

Обучающийся будет готов и способен к саморазвитию и личностному самоопределению;

У обучающегося будет сформирована мотивация к обучению и целенаправленной познавательной деятельности.

Метапредметные результаты:

Обучающийся будет знать, как ставить и формулировать для себя цели действий, прогнозировать результаты, анализировать их (причём как положительные, так и отрицательные), делать выводы в процессе работы и по её окончании, корректировать намеченный план, ставить новые цели; умение соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности.

Обучающийся приобретёт умение оценивать правильность выполнения учебной задачи, корректировать свои действия в соответствии с изменяющейся ситуацией, определять способы действий в рамках предложенных условий.

Предметные результаты:

1. Обучающийся будет владеть теоретической базой и практическими навыками программирования на языке Python;
2. Обучающийся будет владеть навыками разработки, тестирования и отладки несложных программ;
3. Обучающийся будет знать навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
4. Обучающийся будет уметь развивать логические, аналитические и творческие способности.

РАЗДЕЛ № 2. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

1.1 Условия реализации программы

1. Материально-техническое обеспечение:

Характеристика помещения для занятий по программе - учебное помещение образовательного учреждения, оснащенное наглядными

пособиями, учебным оборудованием, мебелью и техническими средствами обучения, в котором проводится учебная, индивидуальная и внеклассная работа с обучающимися в соответствии с действующими государственными образовательными стандартами, учебным планом и образовательными программами.

Для реализации данного курса требуется следующее оборудование:

- Проектор и экран для демонстрации учебного материала
- Доска
- Персональные компьютеры для обучающихся
- Раздаточные материалы
- Наушники с микрофоном

Требуемое программное обеспечение:

- Пакет офисных приложений;
- Программное обеспечение PyCharm, Python;
- Браузер Google Chrome, Mozilla Firefox или Яндекс Браузер.

2. Учебно-методическое и информационное обеспечение:

Ссылки на печатные источники:

1. Лутц М. Изучаем Python. СПб.: Символ Плюс, 2019;
2. Буйначев С. К. Основы программирования на языке Python: учебное пособие. Екатеринбург: Изд-во Урал. ун-та, 2020;
3. Шуманн Ханс-Георг. Python для детей. М.: ДМК-Пресс, 2022;
4. Пэйн, Брайсон. Python для детей и родителей. М.: Издательство «Э», 2019.

1.2 Оценочные материалы и формы аттестации

В процессе изучения программы для оценки текущей работы используются методы устного, письменного, практического, машинного контроля и самоконтроля. Педагогический контроль осуществляется в несколько этапов и включает в себя несколько уровней.

Виды и формы контроля знаний и навыков учащихся:

- Текущий контроль осуществляется регулярно, посредством проведения

лабораторных занятий, заключается в ответе учащихся на контрольные вопросы, фронтальных опросах учителем, а также в демонстрации полученных скриптов в среде Python (опрос, лабораторные работы).

- Тематический контроль—определение результатов обучения за определенный раздел программы, проводится посредством выполнения учащимися контрольных работ (контрольные работы, тестовые занятия).
- Итоговый контроль — проводится по окончании обучения по программе, он предполагает комплексную проверку образовательных результатов по всем ключевым целям и направления (разработка учащимися индивидуальных/групповых проектов и их защита).

Формы контроля результатов (промежуточная аттестация):

- Беседа - это диалогический метод обучения, при котором учитель путем постановки тщательно продуманной системы вопросов подводит учеников к пониманию нового материала или проверяет усвоение ими уже изученного. (применяется во время занятия “Тема 1: Среда программирования Python. Переменные);

- Устный опрос - позволяет выявить правильность ответа по содержанию, его последовательность, самостоятельность суждений и выводов, степень развития логического мышления, культуру речи учащихся. Эта форма применяется для текущего и тематического учета, а также для отработки и развития экспериментальных умений учащихся. Причем устную проверку считают эффективной, если она направлена на выявление осмысленности восприятия знаний и осознанности их использования, если она стимулирует самостоятельность и творческую активность учащихся. (применяется во время занятий: “Тема: 2-4”, “Тема: 6-7”, “Тема 9-10”, “Тема 12-19”);

- Зачёт - проводится для определения достижения конечных результатов обучения по определенной теме каждым учащимся. Перед

началом изучения материала, учащиеся знакомятся с перечнем вопросов и обязательных задач по теме, а также дополнительными вопросами и задачами. Иногда целесообразны закрытые зачеты, когда учащиеся получают вопросы и задания непосредственно во время проведения зачета. Его достоинство заключается в том, что он предполагает комплексную проверку всех знаний и умений учащихся. (применяется во время занятий “Тема: 5”, “Тема: 8”, “Тема 11”);

- Тестирование - представляет собой кратковременное технически сравнительно просто составленное испытание, проводимое в равных для всех испытуемых условиях и имеющее вид такого задания, решение которого поддается качественному учету и служит показателем степени развития к данному моменту известной функции у данного испытуемого. (применяется во время занятия “Тема 15.”, “Тема 19”);

- Контрольная работа - проводится с целью определения конечного результата в обучении по данной теме или разделу, контролировать знания одного и того же материала неоднократно. Целесообразно проводить контрольные работы различного вида. С помощью промежуточной контрольной работы Преподаватель проверяет усвоение учащимися материала в период изучения темы. Итоговая контрольная работа проводится с целью проверки знаний и умений учащихся по отдельной теме, курсу. (применяется во время занятия “Тема 20: Программирование на Python”).

Формы фиксации результатов:

- Ведомости результатов аттестации учащихся (применяются после завершения курса для фиксации итоговых результатов);

- Бланки тестовых заданий и контрольных работ по темам программы (применяются на протяжении всего периода обучения. Тема 1 – Тема 20);

- Разработанные, в ходе проведения занятий, программы в виде файлов (применяются на протяжении всего периода обучения. Тема 1 – Тема 20);

- Журнал посещаемости (применяется на протяжении всего периода обучения. Тема 1 – Тема 20);
- Отзывы детей и родителей (собираются после завершения курса для оперативной обратной связи по поводу предложений и замечаний);
- Свидетельство (сертификат). Выдается участнику после успешного завершения учебной программы.

1.3 Методические материалы

Дидактические и методические материалы: раздел «Среда программирования Python. Переменные» дидактических материалов пособия.

На рисунке 1 изображен фрагмент практической работы по теме “Среда программирования Python. Переменные”.

Практическая часть

Цель работы: ознакомление со средой программирования Python и запуском первых программ.

Ход работы

1. Проверить, установлена ли на компьютере/ноутбуке среда IDE PyCharm.
2. Ознакомиться с помощью учителя со справочным материалом, посвящённым установке языка Python, установке и первичной настройке IDE PyCharm.
3. Создать первую программу на языке Python по приведённому ниже описанию.

Хорошей традицией при изучении языка программирования является написание первой вашей программы с простым приветствием на экране «Привет, я изучаю Python!».

На жёстком диске создать каталог (папку) с именем *PythonProject*. Запустить среду разработки PyCharm. Выбрать пункт *Открыть (Open)* и указать вашу папку *PythonProject*. Откроется рабочее окно вашего проекта (рис. 29).

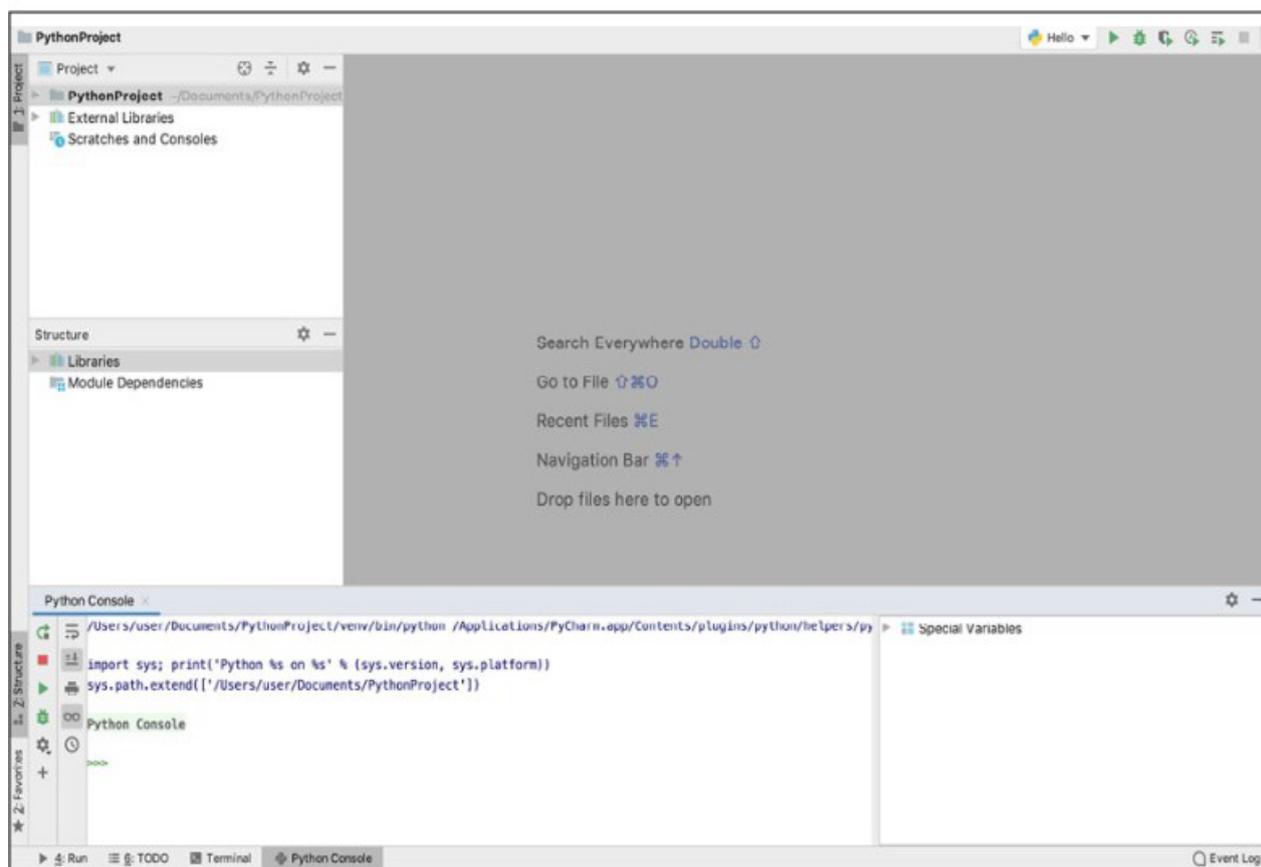


Рисунок 1 – Фрагмент дидактических материалов по Теме №1

Дидактические и методические материалы: раздел «Первые программы на языке Python, основные операторы» дидактических материалов пособия.

На рисунке 2 изображен фрагмент практической работа по теме “Первые программы на языке Python, основные операторы”.

Лабораторная работа 2.1. Первые программы на языке Python, основные операторы

Теоретическая часть

В первой лабораторной работе были показаны основные аспекты работы со средой для языка Python, вы писали простые команды (операторы).

Далее предлагается рассмотреть структуру программы на языке Python, а также основные правила написания программ.

Любую программу на языке Python можно представить как набор лексем (допустимых символов), записанных в определённом порядке и по определённым правилам. Лексема может представлять собой: литералы, знаки пунктуации, переменные, специальные ключевые слова, комментарии.

Программа на языке Python может содержать достаточное количество комментариев, каждый комментарий начинается с символа # «решётка».

Литералы представляют собой значения, заданные в коде программы, например числа (25) или строки ("Привет"). В языке Python используется динамическая типизация (типы данных определяются автоматически, и их не требуется объявлять в программном коде), но при этом Python является языком со строгой типизацией (вы сможете выполнять над объектом только те операции, которые применимы к его типу).

Если говорить об использовании знаков пунктуации, то стоит отметить, что каждая строка в программе на языке Python не должна заканчиваться точкой с запятой, как, например, в C++, но если есть необходимость записать несколько операторов в одну строчку, то их можно разделять точкой с запятой.

Переменные используются для хранения данных.

Важно!

В языке Python нет специального раздела описания переменных, в котором указывается тип переменной перед её первым использованием. Тип переменной определяется по тому значению, которое ей присваивается.

Есть определённые правила для задания имён переменных (идентификаторов): это последовательность букв и цифр, которая не может начинаться с цифры, но может содержать символ подчёркивания (_). Имена переменных чувствительны к регистру. Имена переменных не могут совпадать с ключевыми словами.

Ключевые слова в языке Python имеют специальное назначение и представляют собой управляющие конструкции языка. Примеры: `and`, `break`, `for` и т. д.

Рисунок 2 – Фрагмент дидактических материалов по Теме №2

Дидактические и методические материалы: раздел «Первые программы на языке Python, основные операторы» дидактических материалов пособия.

На рисунке 3 изображен фрагмент практической работы по теме «Первые программы на языке Python, основные операторы».

Теоретическая часть

Воспользоваться материалами из лабораторной работы 2.1.

Практическая часть

Цель работы: ознакомление с основами написания программ на языке программирования Python, работа с операторами присваивания, ввода/вывода данных.

Ход работы

1. Открыть среду разработки PyCharm.
2. Составить программу вычисления значения функции $y(x) = x^2 - 7x + 8$ для заданного с клавиатуры значения аргумента x .

Указание. Примерный вид программы:

```
x=float(input("x= "))
y=x**2-7*x+8
print("y(x)= ", y)
```

3. Составить программу вычисления расстояния между двумя точками, заданными на плоскости своими координатами.

Указание. Примерный вид программы:

```
x1=float(input("x1= "))
y1=float(input("y1= "))
x2=float(input("x2= "))
y2=float(input("y2= "))
r=((x1-x2)**2+(y1-y2)**2)**(1/2)
print("расстояние= ", r)
```

Рисунок 3 – Фрагмент дидактических материалов по Теме №2

Пример шаблона формы для фиксации результатов индивидуальных проектов:

Лист оценивания проекта

Критерии оценивания	1-я группа	2-я группа	...
Актуальность темы			
Соответствие содержания проекта заявленной теме			
Техническая сложность			
Оригинальность			
Дизайн			
Наличие соответствующего <u>музыкального</u> сопровождения с указанием в титрах <u>авторов</u> музыки			
Уровень проработанности проекта			
Возможность применения проекта в школе			
Итоговое количество баллов			

1.4 Календарный учебный график

Вариант календарного учебного графика, когда программа размещается на сайте:

Этапы образовательного процесса	1 год	
Продолжительность учебного года, неделя	72	
Количество учебных дней	72	
Продолжительность учебных периодов	1 полугодие	01.10.2024- 31.12.2024
	2 полугодие	10.01.2025- 31.05.2025
Возраст детей, лет	14-17	
Продолжительность занятия, час	2	
Режим занятия	1 раз /нед	
Годовая учебная нагрузка, час	72	

1.5 Календарный план воспитательной работы

Для выстраивания адресной образовательной деятельности необходимо хорошо понимать логистическую и содержательную структуру работы, включающей обучение и воспитание учащихся.

Рабочая программа воспитательной работы Центра едина для всех творческих объединений и создает единую воспитательную среду учреждения, которая объединяет всех участников образовательного процесса. Практическая реализация цели и задач воспитания учащихся осуществляется в рамках направлений воспитательной работы образовательной организации, представленных в соответствующем модуле.

При составлении календарного плана воспитательной работы творческого объединения было адаптировано содержание модулей календарного плана программы для работы с учащимися творческого объединения по приоритетным направлениям различного уровня с учетом

конкретных условий и особенностей деятельности.

Вызовы времени заставили снова активно заговорить о воспитании личности. С 1 сентября 2020 года вступил в силу Федеральный закон от 31 июля 2020 г. № 304-ФЗ «О внесении изменений в Федеральный закон «Об образовании в Российской Федерации» по вопросам воспитания учащихся». Политика государства в сфере образования определяет воспитание как первостепенный приоритет в образовании, а в качестве важнейших задач выдвигает формирование гражданской ответственности, правового самосознания, духовности и культуры, инициативности, самостоятельности, толерантности, способности к успешной социализации в обществе.

В Федеральном законе акцентировано внимание на том, что система образования не только учит, но и воспитывает, формирует личность, передает ценности и традиции, на которых основано общество, что смысл предлагаемых поправок — «укрепить, акцентировать воспитательную составляющую отечественной образовательной системы».

В соответствии с Федеральным законом № 304-ФЗ вводится механизм организации воспитательной работы (программа воспитания), который является частью общеобразовательной (общеразвивающей) программы педагога.

СПИСОК ЛИТЕРАТУРЫ

Ссылки на печатные источники:

1. Лутц М. Изучаем Python. СПб.: Символ Плюс, 2019;
2. Буйначев С. К. Основы программирования на языке Python: учебное пособие. Екатеринбург: Изд-во Урал. ун-та, 2020;
3. Шуманн Ханс-Георг. Python для детей. М.: ДМК-Пресс, 2022;

4. Пэйн, Брайсон. Python для детей и родителей. М.:Издательсов «Э»,
2019.