Департамент образования администрации Города Томска

Муниципальное автономное образовательное учреждение дополнительного образования Центр творческого развития и гуманитарного образования «Томский Хобби – центр»

Структурное подразделение IT-cube.Томск

Принята на заседании
Педагогического совета
От «28» августа 2022 г.
Протокол № 2

Утверждаю:
Директор Томского Хобби-центра
_____ Л.В.Дубровина
«29» августа 2022 г.

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ

«Основы программирования на языке Python»

Возраст учащихся: 14-16 лет

Срок реализации: 2 года

Автор составитель:
Гураль Илья Сергеевич,
педагог дополнительного образования
Редакция:
Филатова Анастасия Владимировна,
методист

ДОКУМЕНТ ПОДПИСАН ЭЛЕКТРОННОЙ ПОДПИСЬЮ

Сертификат: 0080DD707807906EC989B8DCED4410C612
Владелец: Дубровина Людмила Викторовна
Действителен: с 16.06.2022 до 09.09.2023

г.Томск, 2022.

ХАРАКТЕРИСТИКА ПРОГРАММЫ

***Название программы*:** Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python»

Направленность: техническая
Возраст учащихся: 14-16 лет
Срок обучения: 2 года
Особенности состава учащихся: постоянный
Форма обучения: очная
По степени авторства: модифицированная
По уровню содержания: ознакомительная
По срокам реализации: долгосрочная

В ее основе лежат нормативные документы:

- Конституция РФ;
- Конвенция ООН о правах ребенка;
- Федеральный закон от 29 декабря 2012 года №273-ФЗ «Об образовании в Российской Федерации»;
- Федеральный проект «Успех каждого ребенка» в рамках национального проекта «Образование». Министерство Просвещения Российской Федерации 2018 г.;
- Федеральный проект «Цифровая образовательная среда» в рамках национального проекта «Образование» государственной программы «Развитие образования». Министерство Просвещения Российской Федерации 2018 г.;
- Распоряжение Минпросвещения России от 12.01.2021 N Р-5 "Об утверждении методических рекомендаций по созданию и функционированию центров цифрового образования "IT-куб";
- Приказ Министерства образования и науки Российской Федерации (Минбрнауки России) от 9 ноября 2018 г. №196 г. Москва «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Постановление Главного государственного санитарного врача Российской Федерации 28.09.2020 № 28 г. Москва «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Письмо Минобрнауки России от 18.11.15 года № 09-3242 «Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые)»;
- Устав МАОУ «Томский Хобби-центр»;

Данная программа составлена на основе программы АНО ДПО «Школа анализа данных» «Основы программирования на языке Python (базовый уровень)» Поволоцкого А.В. и с учётом рекомендаций современной методической литературы Л.Н.Буйловой, Е.А.Ворониной, Е.Б.Евладовой, С.В.Кочневой.

Пояснительная записка

В настоящее время мы переживаем большие изменения в развитии современного общества. В современную жизнь человека всё больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причём зачастую не на пользовательском уровне, а на уровне начинающего программиста.

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество

часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Нами был выбран язык Python. Его синтаксис достаточно прост и интуитивно понятен, что понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса.

Актуальность обусловлена тем, что язык программирования Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

Программа позволяет реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Новизна данной программы заключается в формировании профессиональных знаний и умений в области программирования на языке Python с детьми среднего и старшего школьного возраста для последующего применения в учебной, познавательной деятельности и повседневной жизни.

Программа курса:

- обеспечивает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне;
- имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика;
- охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний;
- ориентирована на существующий парк вычислительной техники и дополнительные ограничения;
- допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального);
- предусматривает возможность индивидуальной работы с учащимися.

Педагогическая целесообразность программы выражена в подборе интерактивных и практикоориентированных форм занятий, чтобы, научившись программировать на языке Python, учащиеся получили мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ОГЭ и ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Практическая значимость

В рамках предлагаемого курса «Программирования на языке Python» изучение основ программирования на языке Python − это не столько средство подготовки к будущей

профессиональной деятельности, сколько формирование новых общеинтеллектуальных умений и навыков: разделение задачи на этапы решения, построение алгоритма и др. Программирование способствует формированию мыслительных и психических процессов учащихся (внимание, память, логика), освоению приёмов умственных действий, самостоятельному нахождению и составлению алгоритмов решения задач, умению строить модели, чётко и лаконично реализовывать этапы решения задач, что активизирует процесс индивидуально-личностного становления учащегося. Освоение программы может стать началом профессионального самоопределения подростка.

Цель программы: создание условий для изучения методов программирования на языке программирования Python; рассмотрение различных парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная); подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Для достижения поставленной цели необходимо решить следующие **задачи**:

Развивающие задачи:

1. сформировать и развить навыки алгоритмического и логического мышления, грамотной разработки программ;
2. развить у обучающихся интерес к программированию;
3. приобрести навыки поиска информации в сети Интернет, анализ выбранной информации на соответствии запросу, использование информации при решении задач.

Образовательные задачи:

1. расширить кругозор учащихся в области программирования
2. систематизировать знания по математическим основам информатики.
3. познакомить с принципами и методами функционального программирования;
4. познакомить с принципами и методами объектно-ориентированного программирования;
5. приобрести навыки работы в интегрированной среде разработки на языке Python;
6. изучить конструкции языка программирования Python;
7. познакомить с основными структурами данных и типовыми методами обработки этих структур;
8. приобрести навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

Воспитывающие задачи:

1. воспитать умение выстраивать конструктивный диалог при работе в команде;
2. воспитать интерес к информационной и коммуникативной деятельности, этическим нормам работы с информацией;
3. воспитать упорство и стремление в достижении результата;
4. воспитать бережное отношение к техническим устройствам;
5. воспитать самостоятельность в решении задач с использованием средств вычислительной техники.

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

Сроки реализации данной программы составляют 2 года. Программа рассчитана на 144 учебных часа, по 72 часа за каждый учебный год. Один академический час

составляет 45 мин, где 30 минут уделяется непосредственно работе за компьютером, 15 минут теоретической подготовке, проверке знаний. Занятие имеет следующую структуру:

15 минут – постановка учебной цели и задачей, проверка знаний, объяснение нового материала.

30 минут – проверка и отработка умений, освоение нового материала на практике непосредственно за компьютером.

10 минут – двигательная активность, дидактическая игра по теме, проверка нового знания и его закрепление.

30 минут - работа с новым материалом по замыслу педагога, учащегося. Работа над собственным проектом непосредственно за компьютером.

5 минут – рефлексия, выявление проблемных мест при работе по новой теме.

Возраст учащихся: средний и старший школьный возраст 14-16 лет. Программа рассчитана на детей 14–16 лет, проявляющих интерес к иформационно-коммуникационным технологиям. Группы комплектуются на основании заявлений законных представителей учащихся (самих учащихся с 14 лет) из школьников разного возраста по результатам предварительной аттестации, получившим по итогам тестирования в рейтинговой таблице наивысший балл.

Психолого-педагогические особенности возрастной категории учащихся

Программа составлена с учётом индивидуальных и возрастных особенностей учащихся. Подростковый возраст обычно характеризуют как переломный, переходный, критический, но чаще как возраст полового созревания.

Л. С. Выготский [29:288] подробно рассматривал проблему интересов в переходном возрасте, называя ее «ключом ко всей проблеме психологического развития подростка». Он писал, что все психологические функции человека на каждой ступени развития, в том числе и в подростковом возрасте, действуют не бессистемно, не автоматически и не случайно, а в определенной системе, направляемые конкретными, отложившимися в личности стремлениями, влечениями и интересами. В подростковом возрасте, подчеркивал Л. С. Выготский, имеет место период разрушения и отмирания старых интересов, и период созревания новой биологической основы, на которой впоследствии развиваются новые интересы.

Поэтому так важно заинтересовать учащегося в новой для него сфере-программирования и технического творчества в целом.

Психолого-педагогические особенности учащихся определяют и методы индивидуальной работы педагога с каждым из них, темпы прохождения образовательного маршрута.

Количество учащихся в объединении, их возрастные категории.

Данная программа ориентирована как на мальчиков, так и на девочек. Занятия групповые, в каждой группе от 8 до 12 человек. Набор в группы ведется с июля по сентябрь и с декабря по январь. В программе задействованы учащиеся среднего и старшего школьного возраста.

Особенности набора

Набор в группы ведется ***на основании результатов предварительной аттестации партнера программы «Яндекс-лицей».*** Прием на обучение в данную программу проводится на условиях, определенных локальным нормативным актом организации МАОУ «Томский Хобби-центр», в соответствии с законодательством РФ (ч. 5 ст. 55 Федерального закона № 273-ФЗ) и партнера программы «Яндекс-лицей».

Форма занятий, режим и продолжительность занятий

Форма занятий *очная*. Занятия проходят 2 раза в неделю по 2 академических часа (1 академический час равен 45 минутам, что соответствует СанПиНу 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»).

Количество занятий и учебных часов в неделю

Количество занятий в неделю – 2, количество академических часов -4. Каждое занятие по 2 академических часа. Общий объем часов по данной программе - 144 часа за 2 года.

Ожидаемые результаты

- сформировано ответственное отношения к учению, способность довести до конца начатое дело аналогично завершённым творческим учебным проектам;
- сформированы способности к саморазвитию и самообразованию средствами информационных технологий на основе, приобретённой благодаря иллюстрированной среде программирования, мотивации к обучению и познанию;
- имеется опыт участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;
- сформировано умение выстраивать конструктивный диалог со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;
- сформировано целостное мировоззрение об уровне развития информационных технологий;
- сформировано осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности;
- сформированы ценности здорового и безопасного образа жизни; правила индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.
- умеет 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;
- сформированы представления об основных предметных понятиях («информация», «алгоритм», «исполнитель», «модель») и их свойствах;
- развиты логические способности и алгоритмическое мышление, умение составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами — линейной, условной и циклической;
- умеет выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;
- имеет 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;
- умеет 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;
- сформированы представления о математических объектах информатики и об их свойствах, умение оперировать с числами в различных системах счисления, измерять количество информации, представлять информацию в структурированном

виде;

- сформированы умения соблюдать нормы информационной этики и права.
- умеет самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умеет самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умеет критически оценивать правильность решения учебно- исследовательской задачи;
- умеет корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владеет основами самоконтроля, способность к принятию решений;
- умеет создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- сформированы и развиты навыки в области использования информационно-коммуникационных технологий;
- умеет организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно- исследовательской деятельности.

Формы подведения итогов реализации программы

Освоение данной программы сопровождается процедурой промежуточной аттестации учащихся 3 раза в год: в сентябре, декабре и мае, проводимой в формах, определенных программой и учебно-тематическим планом, как составной частью образовательной программы, и в порядке, установленном приказами и Уставом МАОУ «Томский Хобби-центр» (ч. 1 ст. 58, ч. 2 ст. 30 Федерального закона № 273-ФЗ) и условиями партнера программы «Яндекс-лицей».

Текущий контроль за реализацией программы проводится в разных формах:

- тестирование;
- устный опрос;
- самостоятельные и контрольные работы;
- участие в проектной деятельности.

Учебно-тематический план

Учебно-тематический составлен из расчета 2-х часов в неделю, но рекомендовано выделять дополнительное время для отработки материала или консультаций.

1 год обучения

№ п/п	Наименование тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
Инструктаж по технике безопасности в компьютерном классе		**2**	**2**		Опрос
1.	**Раздел 1. Знакомство с языком Python**	**14**	**6**	**8**	Тестиро

1.1	Занятие 1. Знакомство с программой и с LMS Практическая работа: Установка и настройка среды программирования	2	1	1	вание, решение практических задач
1.2.	Занятие 2. PEP8. Ввод-вывод. Переменные Практическая работа: Работа со справочной системой	2	1	1	
1.3	Занятие 3. Введение в математическую логику. Условный оператор. Отступы. Операции над строками Практическая работа: Задачи на элементарные действия с числами	2	1	1	
1.4	Занятие 4. Сложные условия. Вложенные структуры. Логические операции и их свойства (таблицы истинности). Практическая работа: Выражения	2	1	1	
1.5	Занятие 5. Типы данных. Операции над числами Практическая работа: Переменные	2	1	1	
1.6	Занятие 6. Приоритет операций. Простейшие функции	2	1	1	
1.7	*Самостоятельная работа №1. Линейные алгоритмы, преобразование типов, простейшие функции*	2		2	
2.	**Раздел 2. Условные предложения**	**16**	**5**	**11**	Тестирование, решение практических задач
2.1	Занятие 7. Цикл с предусловием. Введение в системы счисления	2	1	1	
2.2	Занятие 8. Решение задач на цикл с предусловием. Алгоритм Евклида	2		2	
2.3	Занятие 9. Отладка программ	2	1	1	
2.4	Занятие 10. Цикл for. Диапазоны	2	1	1	
2.5	*Самостоятельная работа №2. Циклы*	2		2	
2.6	Занятие 11. Булевы переменные. Прерывания и продолжения циклов	2	1	1	
2.7	Занятие 12. Вложенные циклы	2	1	1	
2.8	*Контрольная работа №1. Циклы и линейные алгоритмы*	2		2	
3	**Раздел 3. Элементы теории множеств**	**18**	**8**	**10**	Тестирование, решение
3.1.	Занятие 13. Множества в Python. Практическая работа: Представление	2	1	1	

	логических операций через диаграммы Эйлера-Венна				практических задач
3.2.	Занятие 14. Строки. Индексация	2	1	1	
3.3.	Занятие 15. Строки. Срезы. Строки. Практическая работа: Системы счисления с основаниями, являющимися степенью числа 2	2	1	1	
3.4.	*Самостоятельная работа №3.Строки и множества*	2		2	
3.5.	Занятие 16. Списки	2	1	1	
3.6	Занятие 17. Кортежи	2	1	1	
3.7	Занятие 18. Списочные выражения. Методы split() и join()	2	1	1	
3.8	Занятие 19. Другие методы списков и строк	2	1	1	
3.9	*Самостоятельная работа №4.Строки и множества*	2	1	1	
4	**Раздел 4. Вложенные списки**	**24**	**6**	**14**	
4.1.	Занятие 20. Вложенные списки. Арифметические операции в различных системах счисления Практическая работа: Арифметические операции в различных системах счисления	2	1	1	Тестирование, решение практических задач
4.2.	Занятие 21. Словари	2	1	1	
4.3.	Занятие 22. Решение задач на словари	2		2	
4.4	*Контрольная работа №2. Циклы и линейные алгоритмы*	2		2	
4.5	Занятие 23. Разработка собственного проекта	4		4	
	ВСЕГО	**144**	**25**	**119**	

Учебно-тематический план

2 год обучения

№ п/п	Наименование тем	Количество часов			Формы аттестации/ контроля
		Всего	Теория	Практика	
5.	**Раздел 5. Решение прикладных задач в Python**	**48**	**18**	**30**	Тестиро

5.1	Занятие 1. Повторение пройденного ранее. Решение задач.	4	2	2	вание,
5.2.	Занятие 2. PEP8. Функции. Введение Практическая работа: Создание функций	4	2	2	решение практических задач
5.3	Занятие 3. Функции. Возвращение значений Практическая работа: Решение задач с использованием функций	4	2	2	
5.4	Занятие 4. Области видимости переменных	4	2	2	
5.5	Занятие 5. Функции. Передача параметров	4	2	2	
5.6	*Самостоятельная работа №1. Функции*	4		2	
5.7	Занятие 6. Функции с переменным числом аргументов	4	2	2	
5.8	Занятие 7. Лямбда-функции	4		2	
5.9	Занятие 8. Функциональная обработка коллекций	4	2	2	
5.9.1	Занятие 9. Библиотеки Python (стандартная библиотека)	4	2	2	
5.9.2	Занятие 10. Примеры решения задач с использованием функций	4	2	2	
5.9.3	*Самостоятельная работа №2. Функции и стандартная библиотека*	4		2	
6	**Раздел 6. Библиотеки Python**	**22**	**8**	**16**	
6.1.	Занятие 11. Библиотеки Python (графика)	4	2	2	
6.2.	Занятие 12. Библиотеки Python (графика + звук)	4	2	2	
6.3.	Занятие 13. Библиотеки Python (документы). Электронные таблицы и работа с ними	6	2	4	
6.4	Занятие 14. Примеры решения задач с использованием электронные таблицы и библиотек	4	2	2	
6.5	*Контрольная работа №1. Функции и библиотеки*	4		4	
7	**Раздел 7. Файловая система**	**16**	**5**	**11**	Тестирование,
7.1.	Занятие 13. Файловая система. Файлы в Python	6	3	3	
7.2.	Занятие 14. Хранение данных. Работа с csv-документами.	6	2	4	решение практических задач
7.4	*Самостоятельная работа №2. Файлы*	4		4	
8	**Раздел 8. Введение в ООП**	**58**	**16**	**42**	
8.1	Занятие 15. Введение в ООП	4	2	2	Тестирова

8.2	Занятие 16. Введение в ООП. Полиморфизм	4	2	2	ние, решение практических задач
8.3	Занятие 17. Определение операторов	4	2	2	
8.4	*Самостоятельная работа №3. ОПП*	4		4	
8.5	Занятие 18. Введение в ООП. Наследование	4	2	2	
8.6	Занятие 19. Введение в ООП. Наследование. Часть №2	4	2	2	
8.7	*Контрольная работа №1. ОПП*	4		2	
8.8	Занятие 20. Методы проектирования программ	12	2	10	
8.9	Занятие 21. Работа над проектом	10		10	
9	Занятие 22. Стиль программирования	4	2	3	
10	Занятие 23. Отладка программ	4	2	3	
	ВСЕГО	144			

Содержание учебно-тематического плана

Раздел 1 – 2. Введение в программирование.

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.

Знакомство с электронной образовательной системой: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой автоматизированной проверки задач.

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки, интерпретатор, код программы и редактор кода.

Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова.

Выражения. Операции. Порядок выполнения операций. МаРазделтические функции. Композиция.

Учащиеся должны знать / понимать:

- понятие программы;
- структура программы на Python;
- режимы работы с Python.
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе.
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;

Практическая работа: Установка программы Python, Режимы работы с Python Работа со справочной системой, Переменные, Выражения, Задачи на элементарные действия с числами.

Самостоятельная работа № 1. Линейные алгоритмы, преобразование типов, простейшие функции

Раздел 2. «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.:

Ввод-вывод в программе, условный оператор, оператор цикла с предусловием. Простейшие программы с использованием условного оператора, оператора циклов while и операторов ввода-вывода. Технология разработки программы.

Практическая работа: на данном этапе обучающиеся разрабатывают первые алгоритмы и программы, а также анализируют, на какие функциональные блоки может быть разбита программа, и определяют работоспособность разработанной программы.

В течение модуля ученики решают большое количество задач: от самых простых до сложных.

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.

Контрольная работа №1. Циклы и линейные алгоритмы

Раздел 3 – 4. Базовые конструкции языка Python.

Раздел 3. «Элементы теории множеств»

Теория: Понятие о языке высокого уровня Python. Структура программы, переменные и константы, работа с числовыми переменными, арифметические операторы в Python. Основные управляющие конструкции алгоритмов с ветвлением в Python.

Устройство циклов for. Основные управляющие конструкции циклического алгоритма в Python.

Практическая работа: простейшие циклы и циклы с переменными.

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 4. «Коллекции. Вложенные списки»

Работа со списками, строками, множествами и кортежами в Python.

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. Понятие итератора.

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.

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

Понятие ассоциативного массива. Словари в Python. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы. Решение задач.

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.

Контрольная работа №1. Циклы и линейные алгоритмы

Разработка собственного проекта на заданную тематику. Проработка освоенного материала в рамках творческого задания, выбранного учеником.

Раздел 5 – 6. Решение прикладных задач в Python.

Раздел 5. «Решение прикладных задач в Python»

Теория: Понятие подпрограммы, процедуры, функции. Функции в Python. оздание 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.

Раздел 6. Библиотеки Python

Модули в Python. Подключение и использование модулей стандартной библиотеки. Модульный принцип компоновки программы. Работа с документацией в стандартной библиотеке. Понятие репозитория различных пакетов Python. Работа с внешними библиотеками Python и утилитой pip.

Практическая работа: решение задач.

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.

Самостоятельная работа № 3 по теме "Функции"

Раздел 7 – 8. Введение в ООП на языке Python.

Понятие класса и объекта. Создание классов на языке Python. Атрибуты и экземпляры объектов. Методы и перегрузка операторов. «Магические» методы в Python. Наследование и множественное наследование. Проектирование классов.

Самостоятельная работа: работа с конспектом, который описывает способы решения задач, разработка проекта по индивидуальному заданию, составление отчёта о выполнении индивидуальной или совместной работы.

Тематика самостоятельных и контрольных работ:

- задачи, в которых по заданному алгоритму необходимо написать программу;
- задачи, в которых необходимо составить алгоритм решения и написать по нему программу;
- задачи, для решения которых необходимо найти некоторую информацию в Интернете;
- выполненные самостоятельные работы загружаются в автоматизированную тестирующую систему для дальнейшей оценки.

Комплекс организационно-педагогических условий

Календарный учебный график

Учебный период	**Количество учебных недель**	**Дата начала учебного периода**	**Каникулы**	
			Продолжительность	**Организация деятельности по отдельному расписанию и плану**
1 полугодие	16 недель	01 сентября	С 01 по 29.12 декабря	С 01. 10.2021 по 05.11 2021г. – работа лагеря с дневным пребыванием детей. С 21.12.2021г. по 09.01.2022г. участие в организации новогодних мероприятий
2 полугодие	20 недель	03 января	С 01 июня по 01 сентября.	30.05.2022 – 17.06.2022 – работа лагеря с дневным пребыванием детей с Работа загородных детских оздоровительно-образовательных лагерей «Лукоморье» и «Солнечная республика». Подготовка и участие в турнирах, соревнованиях.

Продолжительность учебного года – с 01.09.2021 по 25.05.2022 – 36 учебных недель

Условия реализации программы

Материально-техническое обеспечение: для эффективности реализации программы необходимо оборудование и материалы:

- хорошо проветриваемое светлое помещение с хорошим естественным и искусственным освещением;
- столы для компьютера;
- компьютерные стулья;
- шкафы для дидактических материалов, пособий;
- канцтовары.

Информационное обеспечение:

- персональный компьютер (на каждого участника);
- мультимедийный проектор;
- видеоматериалы разной тематики по программе;
- оргтехника;
- выход в сеть Internet;
- специальная и научно-популярная литература для педагога и учащихся.

Аппаратное обеспечение:

- Процессор не ниже Core2 Duo;
- Объем оперативной памяти не ниже 4 ГбDDR3;
- Дисковое пространство на менее 128 Гб;
- Монитор диагональю на мене 19.

Программное обеспечение:

- Операционная система Windows 7 Профессиональная или выше;
- Интерпретатор Python версии 3.7 и выше;
- IDE JatBrains PyCharm;
- Foxit Reader или другой просмоторщик PDF файлов;
- WinRAR;
- Пакет офисных программ;
- Adobe Photoshop или другой растровый графический редактор;
- Любой браузер для интернет серфинга.

Кадровое обеспечение: один педагог со специальным педагогическим образованием, студенты технических СУЗов и ВУЗов.

Методы организации учебного процесса

При организации занятий по программе «Основы программирования на языке Python (базовый уровень)» для достижения поставленных целей и решения поставленных задач используются формы проведения занятий с активными методами обучения:

- занятие в форме проблемно-поисковой деятельности;
- занятие с использованием межпредметных связей;
- занятие в форме мозгового штурма;
- занятие в форме частично-поисковой деятельности.

Методы работы

1. Объяснительно-иллюстративный – предъявление информации различными способами (объяснение, рассказ, беседа, демонстрация и др).
2. Проблемный – постановка задачи и самостоятельный поиск ее решения обучающимися.

3. Программированный – набор операций, которые необходимо выполнить в ходе выполнения практических работ.
4. Эвристический – метод творческой деятельности.
5. Многократный повтор способов работы, подходя к изучению последовательно, от простого к сложному, чередуя медленные темпы с быстрыми.

Формы организации учебных занятий

Основной тип занятий — комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий программы выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок занятий Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом календарно-тематического плана. С учётом регулярного повторения ранее изученных тем, темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема программы начинается с постановки задачи — характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью педагог проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики – отработки умений самостоятельно решать поставленные задачи, соответствующих минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в группе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом ученики не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

В каждую тему интегрированы вопросы, связанные с теоретическими основами изучаемого раздела. Это позволяет учащимся получить более глубокое представление о принципах работы с информацией и способах автоматизации этой деятельности.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены дополнительные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

Формы и методы контроля

Для отслеживания результативности и динамики освоения ребенком образовательной программы педагогу необходимо понимать, какой уровень развития тех

или иных навыков имеет учащийся при зачислении на обучение. Поэтому проводится **входящая диагностика** по диагностической карте промежуточной аттестации, чтобы определить уровень развития ребенка по определенным параметрам, заложенным картой.

Также используются такие формы и методы контроля, как:

- наблюдение;
- устный опрос;
- устный анализ самостоятельных работ;
- тестирование, самостоятельные и контрольные работы;
- участие в проектной деятельности;
- промежуточная аттестация - диагностика по параметрам – в сентябре, декабре и мае;
- обучение заканчивается написанием программы для решения одной из задач;

Наблюдение – необходимый педагогу метод для осуществления промежуточной и текущей аттестации, применяется педагогом постоянно.

Устный опрос – метод при котором педагог может оценить теоретически знания учащихся. Проводится в конце пройденной темы.

Устный анализ самостоятельных работ – дает возможность учащимся научится логически мыслить и уметь высказать собственное суждение, поможет педагогу оценить логическое мышление учащихся. Проводится в конце пройденной темы.

Тестирование, самостоятельные и контрольные работы – помогут учащимся проверить свои знания по пройденным темам и оценить приобретенные умения, проводятся в конце каждого раздела.

Участие в проектной деятельности – это условие, необходимое для становления самостоятельности учащихся, которые сформируют установку на самостоятельность, индивидуальные цели и возможности их реализации;

Промежуточная аттестация проводится три раза за учебный год: *в начале учебного года – в сентябре, в середине учебного года – в декабре и в конце учебного года – в мае.* Задачей промежуточной аттестации является определение эффективности педагогического воздействия и полноты освоения образовательной программы.

Параметры и критерии освоения программы в области специальных знаний языка программирования Pyhton определены партнером «Яндекс-лицей» и проводятся в виде промежуточной аттестации три раза в год. Педагог проводит промежуточную аттестацию по параметрам, не относящимся к специальным знаниям в области языка программирования Pyhton.

Диагностическая карта промежуточной аттестации

№	Ф.И. ребенка	Год рожд.	Параметры						Итого
			Работа в команде	Самостоятельность в выполнении задач	Участие в конкурсах	Специальные знания (***по результатам Яндекс-лицея - партнера программы***)	Алгоритмическое, логическое мышление	Поиск и обработка информации в сети Интернет	

1 балл – низкий уровень

2 балла – средний уровень

3 балла – высокий уровень

При подсчете баллов по каждому учащемуся можно определить уровень освоения им программы, также по каждой группе и по объединению в целом.

Определение уровня каждого учащегося выводится после подсчета баллов по всем параметрам:

от 1 до 8 баллов – низкий уровень освоения программы учащимся;

от 9 до 21 баллов – средний уровень освоения программы учащимся;

от 22 до 27 баллов – высокий уровень освоения программы учащимся.

После этого подсчитывается процентное соотношение уровней освоения по группам и по объединению и результат заносится в сводную таблицу.

Сводная таблица

	Высокий уровень %	Средний уровень %	Низкий уровень %
На начало года реализации образовательной программы			
На середину года реализации образовательной программы			
На конец года реализации образовательной программы			

По этой таблице можно провести анализ результативности данной программы.

Список литературы

Литература, рекомендуемая для учащихся

1. *М. Лутц Изучаем Python. СПб.: Символ-Плюс, 2011*
2. Информатика и ИКТ. Задачник-практикум в 2 частях. Под ред. *И.Г. Семакина* и *Е.К. Хеннера*. М.: БИНОМ, Лаборатория знаний, 2014

Литература, используемая педагогом

1. *К.Ю. Поляков, Е.А. Еремин* Информатика. Углубленный уровень. Учебник для 8, 9, 10 классов. М.: БИНОМ. Лаборатория знаний, 2014 или более поздние редакции.
2. *М. Лутц* Изучаем Python. СПб.: Символ-Плюс, 2011
3. Задачи по программированию. Под ред. *С.М. Окулова*, М.: БИНОМ. Лаборатория знаний, 2006.
4. *С.М. Окулов* Основы программирования. М.: БИНОМ, Лаборатория знаний, 2012.

Ресурсы в интернете

1. Материалы и презентации к урокам в LMS Яндекс.Лицей [Электронный ресурс]. – Режим доступа: https://lyceum.yandex.ru/
2. «Python 3 для начинающих» [Электронный ресурс]. – Режим доступа: www.pythonworld.ru
3. «Питонтьютор» [Электронный ресурс]. – Режим доступа: www.pythontutor.ru