

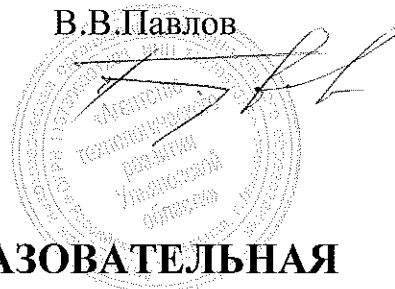
**Автономная некоммерческая организация
дополнительного образования
«Агентство технологического развития Ульяновской области»**

РЕКОМЕНДОВАНА

педагогическим советом
Протокол №4 от 03.09.2021г.

УТВЕРЖДАЮ

Генеральный директор
В.В.Павлов



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА**

«Программирование на Python»

Возраст обучающихся: 14-17 лет
Срок реализации: 2 года

Составители программы:

Кашичкин Максим Олегович,
Старший методист, IT-куб,
Ульяновск.

Ульяновск
2021г.

ОГЛАВЛЕНИЕ

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА	3
УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН	12
СОДЕРЖАНИЕ ПРОГРАММЫ	16
МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ПРОГРАММЫ	28
СПИСОК ЭЛЕКТРОННЫХ ИСТОЧНИКОВ ИНФОРМАЦИИ	33

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

В настоящее время мы переживаем большие изменения в развитии общества. В современную жизнь человека все больше и больше внедряются компьютеры и информационные технологии. Всё большее значение приобретает умение человека грамотно обращаться с компьютером, причем 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. В данной программе выбран язык программирования Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. При этом 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.

Направленность программы

Программа имеет инженерно-техническую направленность, в связи с этим рассматриваются следующие аспекты изучения:

1. Технологический. Содержание программы рассматривается как средство формирования образовательного потенциала, позволяющего развивать наиболее передовые на сегодняшний день технологии — информационные,

интегрирующие в себе науку, технологию, инженерное дело.

2. **Общеразвивающий.** Обучение по данной программе создает благоприятные условия для интеллектуального и духовного воспитания личности ребенка, социально-культурного и профессионального самоопределения, развития познавательной активности и творческой самореализации учащихся.

3. **Общеобразовательный.** Содержание программы рассматривается как средство развития основных познавательных процессов, умения анализировать, выявлять сущности и отношения, описывать планы действий и делать логические выводы, опираясь на такие дисциплины, как теория управления, программирование, теория информации.

Актуальность программы

Актуальность данной программы состоит в том, что она составлена с учётом современных потребностей рынка в специалистах в области информационных технологий, особенно в области программирования.

Научившись программировать на языке Python, учащиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с легкостью выучить любой другой язык программирования.

Знания и умения, приобретенные в результате освоения курса, могут быть использованы обучающимися при сдаче 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.

Данная программа дает возможность детям творчески мыслить, находить самостоятельные индивидуальные решения, а полученные умения и навыки применять в жизни. Развитие творческих способностей помогает также в профессиональной ориентации подростков.

Новизна программы

Новизна программы состоит в том, что она учитывает новые технологические уклады, которые требуют новый способ мышления и тесного взаимодействия при постоянном повышении уровня междисциплинарности проектов, а также использует новые формы диагностики и подведения итогов реализации программы, выполняемые в формате защиты проектов.

В основу программы «Программирование на Python» заложены принципы модульности практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на:

- детальное изучение алгоритмизации;
- реализацию межпредметных связей;
- организацию проектной и исследовательской деятельности.

Цели программы:

- создание условий для изучения методов программирования на языке 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.

Задачи:

Образовательные:

- познакомить с принципами и методами фундаментального программирования;
- познакомить с принципами и методами объектно-ориентированного программирования;
- привить навыки работы в интегрированной среде разработки на языке Python;
- изучить конструкции языка программирования Python;
- привить навыки разработки эффективных алгоритмов и программ на

основе изучения языка программирования Python;

- дать представление о значении информационных технологий в развитии общества и в изменении характера труда человека;

- формировать и развивать навыки публичного выступления.

Воспитательные:

- воспитать мотивацию учащихся к изобретательству, созданию собственных программных реализаций;

- привить стремление к получению качественного законченного результата в проектной деятельности;

- привить информационную культуру: ответственное отношение к информации с учетом правовых и этических аспектов её распространения, избирательного отношения к полученной информации;

- формировать правильное восприятие системы ценностей, принципов, правил информационного общества;

- формировать потребность в самостоятельном приобретении и применении знаний, потребность к постоянному саморазвитию;

- воспитывать социально-значимые качества личности человека: ответственность, коммуникабельность, добросовестность, взаимопомощь, доброжелательность.

Развивающие:

- способствовать развитию навыков алгоритмического и логического мышления, грамотной разработки программ;

- способствовать приобретению навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;

- развивать познавательные способности ребенка, память, внимание, пространственное мышление, аккуратность и изобретательность;

- формировать творческий подход к поставленной задаче;

- развивать навыки инженерного мышления, умения работать как по

предложенным инструкциям, так и находить свои собственные пути решения поставленных задач;

- развивать навыки эффективной деятельности в проекте;
- развивать стрессоустойчивость;
- развивать способности к самоанализу, самопознанию;
- формировать навыки рефлексивной деятельности.

Отличительные особенности программы

Программа предназначена для учащихся, проявляющих повышенный интерес к программированию. Программа имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика.

Ключевым элементом обучения является проектная деятельность, которая ориентирована на использование знаний, умений и навыков, полученных в ходе обучения, для постановки и решения практических задач, которые носят прикладной характер. Она позволяет учащимся участвовать в создании конкретного результата и научиться работать в условиях ограниченного времени, под руководством заказчика, презентовать проект, а также обрести навыки профессиональной коммуникации с контрагентами.

Категория обучающихся

Программа предназначена для детей, проявляющих интерес к программированию, стремящимся к саморазвитию, профессиональному самоопределению, имеющим начальные представления о языках программирования.

Возраст обучающихся: 14 — 17 лет.

Наполняемость группы: 15 человек.

Состав группы: разновозрастной.

Условия приема детей

На курсы программы зачисляются учащиеся прошедшие отбор в два этапа: первый – дистанционное онлайн тестирование, второй – очное собеседование с преподавателем.

Срок реализации программы: 2 года.

Структура программы:

Программа состоит из двух разделов:

1. Основы программирования на языке Python.
2. Основы промышленного программирования.

Форма реализации программы — очная с использованием электронного обучения.

Под электронным образованием понимается реализация образовательных программ с использованием информационно - образовательных ресурсов, информационно-коммуникационных технологий, технических средств, а также информационно-телекоммуникационных сетей, обеспечивающих передачу информационно-образовательных ресурсов и взаимодействие участников образовательного пространства.

Формы организации деятельности обучающихся

При изучении тем программа предусматривает использование фронтальной, индивидуальной и групповой формы учебной работы обучающихся.

При организации занятий по курсу «Программирование на языке Python» для достижения поставленных целей и 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 обучения

Основным методом обучения является метод проектов.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на 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.

В ходе обучения проводится промежуточные 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.

Типы занятий: теоретические, практические, комбинированные, контрольные.

Режим занятий

Четыре часа один раз в неделю или два часа по два раза в неделю с разницей между занятиями не менее 2 дней.

Ожидаемые результаты

Основным результатом обучения является достижение высокой информационно-коммуникационной компетентности учащегося в области программирования на языке Python.

В результате освоения программы обучающийся должен приобрести следующие знания, умения и навыки:

знать:

- ✓ правила работы с компьютером и технику безопасности;
- ✓ основные предметные понятия («информация», «алгоритм», «исполнитель», «модель») и их свойства;
- ✓ основы современных языков программирования;
- ✓ особенности работы с интегрированной средой разработки;
- ✓ базовые и сложные конструкции, способы организации процедур и

функций в языке программирования Python;

✓ технику ведения проектной деятельности и принципов тайм-менеджмента.

уметь:

✓ работать с информацией: находить с применением правил поиска в компьютерных сетях, оценивать и использовать информацию из различных источников при выполнении заданий и проектов по различным темам;

✓ определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе программы анализа данных, читать и понимать программы, написанные на языке программирования высокого уровня Python;

✓ выполнять пошагово алгоритмы управления исполнителями и анализа числовых и текстовых данных;

✓ 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;

✓ самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;

✓ самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;

✓ критически оценивать правильность решения учебно-исследовательской задачи;

✓ корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;

✓ организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской

деятельности;

обладать навыками:

✓ исследовательской, проектной и социальной деятельности, строить логическое доказательство;

✓ использования, создания и преобразования различных символьных записей, схем и моделей для решения познавательных и учебных задач в различных предметных областях, исследовательской и проектной деятельности;

✓ разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

✓ использования специальных средств и библиотек языка Python

✓ работы в интегрированной среде разработки на языке программирования Python;

✓ алгоритмического и логического мышления, грамотной разработки программ;

✓ проектирования, разработки, документирования и представления собственных проектов;

✓ самообразования - периодической оценкой своих успехов и собственной работы самими обучающимися.

Способы определения результативности

Педагогическое наблюдение, педагогический анализ результатов решения задач с использованием автоматизированной системы контроля знаний, результаты участия в интеллектуальных конкурсах всероссийского уровня.

Виды контроля:

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

Формы подведения итогов реализации программы

По окончании обучения проводится итоговая аттестация в форме

публичной защиты проектов. Документальной формой подтверждения итогов промежуточной аттестации является документ об образовании установленного Центром «Поиск» образца и сертификат от компании-партнера Samsung.

УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

1. РАЗДЕЛ «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON»

№	Наименование кейса, темы	Количество часов		
		Теория	Практика	Всего
	Модуль 1. Введение в программирование.	14	16	30
1	Тема 1.1. Знакомство с системой Яндекс.Контест. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода. Простейшие программы с выводом на экран.	2	1	3
2	Тема 1.2. Условный оператор. Переменные и арифметика.	2	1	3
3	Тема 1.3. Знакомство с циклом while.	2	2	4
4	Тема 1.4. Отладчик.	2	2	4
5	Тема 1.5. Знакомство с циклом for.	2	2	4
6	Тема 1.6. Погружение в условия. True, False, Break, Continue.	2	2	4
7	Тема 1.7. Знакомство со списками.	2	2	4
8	Тема 1.8. Решение задач по теме модуля 1.		4	4
	Модуль 2. Базовые конструкции в Python.	20	22	42
9	Тема 2.1. Дополнительные возможности цикла for. Кортежи. Сортировки.	2	2	4
10	Тема 2.2. Знакомство со срезами и диапазонами. Равенство и совпадение объектов.	2	2	4
11	Тема 2.3. Списочные выражения. Методы split и join.	2	2	4
12	Тема 2.4. Другие методы списков и строк.	2	2	4

13	Тема 2.5. Функции.	4	2	6
14	Тема 2.6. Области видимости переменных.	4	2	6
15	Тема 2.7. Контрольная работа по темам 1 и 2 модулей.		2	2
16	Тема 2.8. Разбор контрольной работы.		2	2
17	Тема 2.9. Введение в компьютерную графику. Модуль TKInter.	4	2	6
18	Тема 2.10. Решение задач по теме модуля.		4	4
	Модуль 3. Решение прикладных задач в Python.	35	37	72
19	Тема 3.1. Повторение материала 1 и 2 модулей.	2	2	4
20	Тема 3.2. Функции. Детальное изучение. Функциональная парадигма программирования.	10	6	16
21	Тема 3.3. Словари и множества.	8	4	12
22	Тема 3.4. Обзор стандартной библиотеки Python.	2	4	6
23	Тема 3.5. Дополнительные библиотеки Python.	8	4	12
24	Тема 3.6. Введение в ООП	5	5	10
25	Тема 3.7. Решение задач по теме модуля.		8	8
26	Тема 3.8. Итоговая контрольная работа.		2	2
27	Тема 3.9. Разбор контрольной работы.		2	2
	Итого	69	75	104 + 16

2. РАЗДЕЛ ОСНОВЫ ПРОМЫШЛЕННОГО ПРОГРАММИРОВАНИЯ

№	Наименование кейса, темы	Количество часов		
		Теория	Практика	Всего
	Модуль 1. Программирование на Python.	18	24	42

1	Тема 1.1. Повторение. Решение задач на основные конструкции и структуры данных. Решение задач на классы и библиотеки.	2	4	6
2	Тема 1.2. ООП. Наследование.	4	2	6
3	Тема 1.3. Обработки исключительных ситуаций в Python.	4	2	6
4	Тема 1.4. Рекурсия.	2	1	3
5	Тема 1.5. Файлы в Python. Работа с различными типами файлов. Использование библиотек.	6	3	9
6	Тема 1.6. Решение задач по теме модуля 1.		10	10
7	Тема 1.7. Самостоятельная работа.		2	2
	Модуль 2. Базовые конструкции в Python.	12	18	30
8	Тема 2.1. Подходы к тестированию. Создание простых тестов. Использование библиотек unittest и py.test	4	2	6
9	Тема 2.2. Работа с командной строкой. Библиотека argparse.	4	2	6
10	Тема 2.3. Репозитории. Совместная работа над проектами.	4	2	6
11	Тема 2.4. Самостоятельная работа		2	2
12	Тема 2.5. Решение задач по теме модуля 2.		8	8
13	Тема 2.6. Контрольная работа по темам 1 и 2 модулей.		2	2
	Модуль 3. Проекты GUI, Игры и API.	18	54	72
14	Тема 3.1. Изучение виджетов.	6	4	10
15	Тема 3.2. Решение задач на тему «GUI».		4	4
16	Тема 3.3. Самостоятельная работа.		2	2
17	Тема 3.4. Работа над проектом по теме «GUI» и его защита.		8	8

18	Тема 3.5. Изучение игровых технологий и средств языка Python для их реализации.	6	8	14
19	Тема 3.6. Самостоятельная работа.		2	2
20	Тема 3.7. Работа над проектом по теме «Игры» и его защита.		8	8
	Тема 3.8. Изучение некоторых API и работа с ними с помощью языка Python.	6	8	14
21	Тема 3.9. Самостоятельная работа.		2	2
22	Тема 3.10. Работа над проектом по теме «API» и его защита.		8	8
	Итого	48	96	144

СОДЕРЖАНИЕ КУРСА «ПРОГРАММИРОВАНИЕ НА PYTHON»

1. РАЗДЕЛ «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON»

Модуль 1. Введение в программирование.

Знакомство с виртуальной средой взаимодействия: 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.

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

На этом 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:

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы сортировки списков;
- алгоритмы обработки списков (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов).

Учащиеся должны уметь:

- устанавливать необходимый инструментарий для разработки приложений;
- составлять программный код с использованием разных типов данных и операций над ними;

- составлять, отлаживать и исполнять на персональном компьютере программы решения учебных задач по программированию;
- применять условные конструкции;
- решать задачи с использованием разных видов циклов;
- составлять программы обработки списков;
- искать информацию в свободных источниках и структурировать ее;
- объективно оценивать результаты своей работы.

Формы занятий, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа.

Тема 1.1. Знакомство с системой Яндекс.Контест.

Теория. Понятия кода, интерпретатора, программы. Интегрированные среды, исполнение кода.

Практика. Простейшие программы с выводом на экран.

Тема 1.2. Условный оператор. Переменные и арифметика.

Теория. Алфавит и словарь языка Python. Объявление переменных. Типы данных. Арифметические операции. Операции с присваиванием. Линейные программы. Условный оператор.

Практика. Обзор типов данных. Вычисление математических выражений с помощью стандартных арифметических функций. Решение задач по теме.

Тема 1.3. Знакомство с циклом while.

Теория. Определение цикла. Оператор цикла с предусловием.

Практика. Решение задач по теме «Цикл while».

Тема 1.4. Отладчик.

Теория. Определение. Установка и запуск отладчика программного кода.

Практика. Решение задач.

Форма подведения итогов: самостоятельная работа на решение задач по теме 1.4.

Тема 1.5. Знакомство с циклом for.

Теория. Оператор цикла с постусловием. Оператор с известным числом повторов.

Практика. Решение задач по теме «Цикл for».

Тема 1.6. Погружение в условия. True, False, Break, Continue.

Теория. Принцип работы операторов True, False, Break, Continue.

Практика. Решение задач на тему «Погружение в условия».

Тема 1.7. Знакомство со списками.

Теория. Определение понятия «списки». Объявление списка. Действия над списками, над элементами списка.

Практика. Решение задач по теме «Списки»

Форма подведения итогов: самостоятельная работа по теме модуля.

Модуль 2. Базовые конструкции языка Python.

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. Простейшие циклы и циклы с переменными.

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

Понятие итератора.

Понятие подпрограммы, процедуры, функции. Функции в Python. Решение задач.

Учащиеся должны знать:

- базовые конструкции языка программирования Python (операции присваивания, ветвления, цикла, ввод\вывод, запись констант и выражений);
- основные методы сортировки списков;
- алгоритмы обработки списков (поиск элемента с заданными свойствами, удаление элементов, вставка элементов, замена элементов);
- способы организации процедур и функций.

Учащиеся должны уметь:

- записывать конструкции языка программирования Python;
- тестировать и осуществлять отладку программ;
- объективно оценивать результаты своей работы.

Формы занятий, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа.

Тема 2.1. Повторение. Дополнительные возможности цикла for. Кортежи. Сортировки.

Теория. Структура программы. Стандартные функции и арифметические выражения. Ветвления. Логические операции. Оператор цикла с известным количеством повторов, постусловием, предусловием. Оператор безусловного перехода.

Практика. Решение задач на тему «Цикл for».

Тема 2.2. Знакомство со срезами и диапазонами.

Теория. Понятие срезов и диапазонов. Равенство и совпадение объектов.

Практика. Решение задач по теме «Срезы и диапазоны»

Тема 2.3. Списочные выражения. Методы split и join.

Теория. Работа со списками. Применение методов split и join.

Практика. Решение задач на тему «Методы split и join».

Тема 2.4. Другие методы списков и строк.

Теория. Генераторы списков. Методы find и rfind. Метод replace и count.

Практика. Решение задач на методы списков и строк.

Тема 2.5. Знакомство с функциями.

Теория. Определение функции. Правила составления и вызова функций в программном коде. Локальные и глобальные переменные.

Практика. Решение задач на тему «Функции».

Тема 2.6. Области видимости переменных.

Теория. Локальная область видимости. Глобальная область видимости.

Нелокальная область видимости

Практика. Решение задач на тему «Области видимости переменных».

Тема 2.7. Введение в компьютерную графику. Модуль TKInter.

Теория. Компьютерная графика и её виды. Кроссплатформенная библиотека для разработки графического интерфейса TKInter.

Практика. Решение задач на тему «Модуль TKInter».

Форма подведения итогов: контрольная работа по теме модуля.

Модуль 3. Решение прикладных задач в Python.

Понятие ассоциативного массива. Словари в 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.

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 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 систему проверки задач Яндекс.Контест для дальнейшей оценки.

Зачётное занятие: выполнение итогового индивидуального задания по предложенной теме.

Учащиеся должны знать:

- основные понятия объектно-ориентированного проектирования;
- понятия ассоциативного массива, репозитория;
- модульный принцип компоновки программ;
- алгоритмы решения основных видов прикладных задач.

Учащиеся должны уметь:

- генерировать идеи;
- подключать и использовать модули стандартной библиотеки;
- работать с документацией в стандартной библиотеке;
- работать с внешними библиотеками и утилитой pip;
- объективно оценивать результаты своей работы.

Формы занятий, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- контрольная работа.

Тема 3.1. Повторение материала 1-го и 2-го модулей.

Теория. Условные операторы. Операторы цикла. Работа со списками, их методами.

Практика. Решение задач.

Тема 3.2. Функции. Детальное изучение. Функциональная парадигма программирования.

Теория. Понятие функции. Именные функции, инструкция def. Аргументы функции. Анонимные функции, инструкция lambda.

Практика. Решения задач на тему «Функции и функциональная парадигма программирования».

Тема 3.3. Словари и множества.

Теория. Словари (dict) и работа с ними. Методы словарей. Множества (set и frozenset).

Практика. Решение задач на тему «Словари и множества».

Тема 3.4. Обзор стандартной библиотеки Python.

Теория. Работа с модулями: создание, подключение инструкциями `import` и `from`. Модуль `os`, `sys`. Модуль `contextlib`. Модуль `abc`. Модуль `datetime`. Модуль `collections`.

Практика. Решение задач на тему «Стандартная библиотека Python»

Тема 3.5. Дополнительные библиотеки Python.

Теория. Модуль `string`, `re`, `struct`, `CSV`, `smtplib`. Утилита `pip`.

Практика. Решение задач на тему «Дополнительные библиотеки Python».

Тема 3.6. Введение в ООП.

Теория. Объектно-ориентированное программирование. Классы и объекты.

Практика. Решение задач на тему «Введение в ООП».

Тема 3.7. Итоговая контрольная работа по темам модулей 1, 2 и 3.

Форма подведения итогов: контрольная работа.

2. РАЗДЕЛ ОСНОВЫ ПРОМЫШЛЕННОГО ПРОГРАММИРОВАНИЯ

Модуль 1. Программирование на Python.

Объектно-ориентированное программирование. Наследование и исключения. Рекурсия. Работа с файлами. Файловые форматы: `CSV`, `JSON`, `XML`.

На этом этапе обучающиеся продолжают осваивать язык Python и решать тренировочные задачи. В течение модуля ученики заканчивают изучение объектно-ориентированного подхода к программированию, рассматривают механизм поиска и обработки исключительных ситуаций (ошибок), знакомятся с различными файловыми.

В течение модуля ученики решают большое количество задач различной сложности и пишут 2 самостоятельные работы.

Учащиеся должны знать:

- базовые и сложные конструкции языка программирования Python;
- понятие наследования, исключения, рекурсии;

- определения и описания классов и объектов;
- понятия «инкапсуляция», «наследование», «полиморфизм»;
- механизм поиска и обработки исключительных ситуаций (ошибок).

Учащиеся должны уметь:

- составлять программный код с использованием разных файловых форматов;
- решать тренировочные задачи;
- искать информацию в свободных источниках и структурировать ее;
- объективно оценивать результаты своей работы.

Формы занятий, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа.

Тема 1.1. Повторение.

Теория. Интегрированные среды, исполнение кода.

Практика. Решение задач на основные конструкции и структуры данных.

Решение задач на классы и библиотеки.

Тема 1.2. ООП. Наследование.

Теория. Объектно-ориентированное программирование. Наследование.

Инкапсуляция. Полиморфизм.

Практика. Решение задач по теме «ООП. Наследование».

Тема 1.3. Обработка исключительных ситуаций в Python.

Теория. Исключения в Python. Конструкция try - except для обработки исключений.

Практика. Решение задач по теме «Обработка исключительных ситуаций в Python»

Тема 1.4. Рекурсия.

Теория. Рекурсивные функции. Условия завершения рекурсии. Оператор return.

Практика. Решение задач по теме «Рекурсия».

Тема 1.5. Файлы в Python. Работа с различными типами файлов.

Теория. Файлы. Работа с файлами (чтение, запись). Использование библиотек.

Практика. Решение задач по теме «Файлы в Python».

Форма подведения итогов: самостоятельная работа по теме модуля.

Модуль 2. Тестирование программ и командные методы работы

В этом модуле ученики разбирают устройство командной строки и принципы работы с ней, детально изучают принципы тестирования программ и соответствующие библиотеки языка Python, а также погружаются в методику групповой работы над проектом В течение модуля школьники по традиции решают большое количество задач и пишут самостоятельную работу.

Учащиеся должны знать:

- базовые и сложные конструкции языка программирования Python;
- устройство командной строки и принципы работы с ней;
- библиотеки языка Python.

Учащиеся должны уметь:

- работать в команде;
- работать в командной строке;
- тестировать и осуществлять отладку программ;
- объективно оценивать результаты своей работы.

Формы занятий, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- самостоятельная работа.

Тема 2.1. Подходы к тестированию.

Теория. Создание простых тестов. Использование библиотек unittest и

py.test.

Практика. Решение задач на тему «Подходы к тестированию».

Тема 2.2. Работа с командной строкой. Библиотека argparse.

Теория. Параметры командной строки (именованные, обязательные и необязательные). Использование библиотеки argparse. Позиционные аргументы.

Практика. Решение задач по теме «Работа с командной строкой».

Тема 2.3. Репозитории. Совместная работа над проектами.

Теория. Определение репозитория. Обзор популярных хранилищ. Принципы работы с репозиториями.

Практика. Решение задач на тему «Репозитории».

Тема 2.4. Другие методы списков и строк.

Теория. Списки. Функции и методы списков. Байтовые строки (bytes и bytearray).

Практика. Решение задач на методы списков и строк.

Тема 2.5. Решение задач по темам модуля 2.

Практика. Решение задач на тему «Функции».

Форма подведения итогов: контрольная работа по темам модулей 1 и 2.

Модуль 3. Проекты GUI, Игры и API

Модуль GUI открывает группу проектных тем курса. В теме «GUI» ученики рассматривают принципы создания пользовательских интерфейсов на языке Python с использованием соответствующих библиотек. Тема «Игры» предлагает к изучению различные виды компьютерных игр, технологии и алгоритмы для их проектирования. Тема «API» раскрывает способы интеграции различных сетевых приложений и сервисов.

Каждая тема содержит мини-проекты и итоговый проект, который ученики по желанию защищают на итоговом занятии.

Учащиеся должны знать:

- основные понятия объектно-ориентированного проектирования;

- способы интеграции различных сетевых приложений и сервисов;
- принципы создания пользовательских интерфейсов на языке Python с использованием соответствующих библиотек.

Учащиеся должны уметь:

- работать с информацией: находить, оценивать и использовать информацию из различных источников;
- разрабатывать эффективные алгоритмы и программ на основе изучения языка программирования Python;
- использовать специальные средств и библиотеки языка Python;
- самостоятельно изучать новые технологии;
- работать в команде и использовать соответствующие технологии для организации командной работы;
- генерировать идеи;
- объективно оценивать результаты своей работы.

Формы занятий, используемые при изучении данного модуля:

- лекционная,
- групповая работа,
- групповые консультации,
- защита проектов.

Тема 3.1. Изучение виджетов.

Теория. Обзор библиотек для создания графического интерфейса пользователя. Перетаскивание (drag and drop). Диалоговые окна. События и сигналы. Управление макетом. Меню и тулбары.

Практика. Создание собственного виджета в PyQt5.

Тема 3.2. Решения задач на тему «GUI».

Практика. Решения задач на тему 3.2.

Форма подведения итогов. Самостоятельная работа.

Тема 3.3. Работа над проектом по теме «GUI» и его защита.

Практика. Организация работы над проектом. Постановка проблемной ситуации, формулировка цели и задач. Работа над проектом. Подготовка к

защите.

Форма подведения итогов. Публичная защита проекта.

Тема 3.4. Изучение игровых технологий и средств языка Python для их реализации.

Теория. Классические игры на клетчатом поле. Изображения. Спрайты. Столкновения и другие взаимодействия. Украшения игры.

Практика. Решения задач на тему «Игры».

Форма подведения итогов. Самостоятельная работа.

Тема 3.5. Работа над проектом по теме «Игры» и его защита.

Практика. Организация работы над проектом. Постановка проблемной ситуации, формулировка цели и задач. Работа над проектом. Подготовка к защите.

Форма подведения итогов. Публичная защита проекта.

Тема 3.6. Изучение некоторых API и работу с ними с помощью языка Python.

Теория. Знакомство с API. Работа с протоколом HTTP. Введение в Telegram Bot API.

Практика. Решение задач на API Яндекс.Карт. Создание сценариев диалогов в Telegram Bot API.

Форма подведения итогов. Самостоятельная работа.

Тема 3.7. Работа над проектом по теме «API» и его защита.

Практика. Организация работы над проектом. Постановка проблемной ситуации, формулировка цели и задач. Работа над проектом. Подготовка к защите.

Форма подведения итогов. Публичная защита проекта.

**МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ
ПРОГРАММЫ «ПРОГРАММИРОВАНИЕ НА PYTHON»**

Тема модуля	Форма занятий	Приёмы и методы организации и образовательного процесса	Дидактический материал. Электронные источники	Техническое оснащение и расходный материал	Форма подведения итогов
1. РАЗДЕЛ «ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ PYTHON»					
Модуль 1. Введение в программирование	Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.	1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лица (http://anytask.org); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com); 4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru);	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	Контрольная работа.

			6. Официальная документация языка Python (http://docs.python.org).		
Модуль 2. Базовые конструкции в Python	Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.	1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru); 2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org); 3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com); 4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru); 6. Официальная документация языка Python (http://docs.python.org).	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	Контрольная работа.
Модуль 3. Решение прикладных задач в Python	Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод	1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. 	Контрольная работа.

		<p>мозгового штурма. Проблемно-поисковый.</p>	<p>одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru);</p> <p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org);</p> <p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com);</p> <p>4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru);</p> <p>6. Официальная документация языка Python (http://docs.python.org).</p>	<ul style="list-style-type: none"> ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	
--	--	---	--	---	--

2. РАЗДЕЛ ОСНОВЫ ПРОМЫШЛЕННОГО ПРОГРАММИРОВАНИЯ

<p>Модуль 1. Программирование на Python</p>	<p>Комбинированная</p>	<p>Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru);</p> <p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать</p>	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет, на которых установлено следующие программное обеспечение: операционная система Windows (версия не ниже 7); Android Studio; Eclipse. 	<p>Самостоятельная работа.</p>
---	------------------------	---	---	---	--------------------------------

			<p>большинство внутренних процессов Яндекс.Лицея (http://anytask.org);</p> <p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com);</p> <p>4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru);</p> <p>6. Официальная документация языка Python (http://docs.python.org).</p>	<ul style="list-style-type: none"> ● Презентационное оборудование. ● Мобильные устройства на базе ОС Android. 	
Модуль 2. Базовые конструкции в Python	Комбинированная	Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru);</p> <p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org);</p> <p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com);</p> <p>4. Федеральный портал Единая коллекция</p>	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	Контрольная работа.

			<p>образовательных ресурсов (http://school-collection.edu.ru);</p> <p>6. Официальная документация языка Python (http://docs.python.org).</p>		
Модуль 3. Проекты GUI, Игры и API	Комбинированная	<p>Метод проектов. Объяснительно-иллюстративный. Метод мозгового штурма. Проблемно-поисковый.</p>	<p>1. Яндекс.Контест, сервис, позволяющий программистам со всего мира соревноваться на предложенных задачах. Сервис способен одновременно обрабатывать терабайты данных, поэтому легко выдержит нагрузку в более чем тысячу участников. (https://contest.yandex.ru);</p> <p>2. LMS AnyTask – система управления обучением, позволяющая автоматизировать большинство внутренних процессов Яндекс.Лицея (http://anytask.org);</p> <p>3. Сервис PythonTutor, позволяющий визуализировать исполнение кода на языке Python (http://pythontutor.com);</p> <p>4. Федеральный портал Единая коллекция образовательных ресурсов (http://school-collection.edu.ru);</p> <p>6. Официальная документация языка Python (http://docs.python.org).</p>	<ul style="list-style-type: none"> ● Ноутбуки с мышкой и доступом к сети Интернет. ● Презентационное оборудование. ● Принтер. ● Маркерная доска. 	Защита проекта

СПИСОК ИСТОЧНИКОВ ИНФОРМАЦИИ

использованных при написании программы:

1. К. Ю. Поляков, Е. А. Еремин. Информатика. Углублённый уровень. Учебник для 10 класса в 2 частях. М.: БИНОМ. Лаборатория знаний, 2014.
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.

рекомендованных обучающимся:

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.
3. Материалы и презентации к урокам в LMS Яндекс.Лицея.
4. Сайт pythonworld.ru — «Python 3 для начинающих».
5. Сайт pythontutor.ru — «Питонтьютор».
6. <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwB1> — Лекции А.В. Умнова, прочитанные в Школе Анализа Данных Яндекса.