

Министерство образования и науки Мурманской области
Государственное автономное негосударственное образовательное учреждение
Мурманской области «Центр образования «Лапландия»

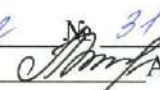
ПРИНЯТА

методическим советом

Протокол

от 14.06.2022

№ 31

Председатель  А.Ю. Решетова

«УТВЕРЖДЕНА»

Приказом ГАОУ МО

«ЦО «Лапландия»

от

№

Директор  С.В. Кулаков



КВАНТОРИУМ-5.1

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ТЕХНИЧЕСКОЙ НАПРАВЛЕННОСТИ
«Информационные технологии.
Программирование на языке Python. Линия 1»

Возраст учащихся: 11-13 лет

Срок реализации программы: 2 года

Автор-составитель:

Рзаев Роман Александрович,
педагог дополнительного образования

Мурманск
2022

1. Пояснительная записка

1.1 Область применения программы.

Дополнительная общеобразовательная общеразвивающая программа «Информационные технологии. Программирование на Python» (далее – Программа) может применяться в учреждениях дополнительного образования при наличии материально-технического обеспечения, педагогических кадров и соблюдении санитарных норм.

Направленность программы: **техническая.**

1.2. Нормативно-правовая база разработки и реализации программы

Программа разработана в соответствии с нормативными правовыми актами и государственными программными документами:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»;
- Стратегия научно-технологического развития Российской Федерации, утвержденная Указом Президента РФ от 01.12.2016 № 642;
- Распоряжение Правительства РФ от 31 марта 2022 г. N 678-р «Об утверждении Концепции развития дополнительного образования детей до 2030 г. и плана мероприятий по ее реализации»;
- Приказ Министерства просвещения Российской Федерации от 09.11.2018 г. № 196 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Письмо Министерства образования и науки РФ от 25.07.2016 № 09-1790 «Рекомендации по совершенствованию дополнительных образовательных программ, созданию детских технопарков, центров молодежного инновационного творчества и внедрению иных форм подготовки детей и молодежи по программам инженерной направленности»;
- Постановление Главного государственного санитарного врача РФ от 28.09.2020 № 28 «Об утверждении СанПиН 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Постановление Главного государственного санитарного врача РФ от 28.01.2021 №2 «Об утверждении санитарных правил и норм СанПиН 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания».

1.3. Актуальность программы

Актуальность образовательной общеразвивающей программы «Программирование на языке Python» (далее – Программа) вызвана потребностью современного информационного общества в высокообразованных, адаптированных к изменениям специалистах в IT-сфере.

Программа предназначена для приобретения учащимися навыков языка программирования Python и его использования при решении практических задач.

Программа курса 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 обучающимися для последующего применения в учебной, познавательной деятельности и повседневной жизни.

Педагогическая целесообразность данной программы заключается в том, что она отвечает потребностям общества; федеральным государственным стандартам общего образования, ориентированным на формирование компетентной, творческой личности; специфике образовательной деятельности Детского технопарка «Кванториум», в частности направлению «IT-квантум». Программа направлена на формирование «soft» и «hard» компетенций, развитие критического, продуктивного, командного мышления имеет практико-ориентированную направленность.

Содержание программы определяется с учётом возрастных особенностей обучающихся, широкими возможностями социализации в процессе общения.

1.4. Цель программы

Целью реализации программы является формирование у учащихся умения разрабатывать эффективные алгоритмы и реализовывать их в виде программ, написанных на языке программирования Python.

1.5 Задачи программы

Задачи программы на 1-й год обучения:

Образовательные:

- освоение основного синтаксиса языка программирования Python;
- приобретение навыков работы с инструментальными программными средами;
- овладение общими понятиями и принципами программирования;

Развивающие:

- умение искать и анализировать информацию в открытом доступе;
- развитие навыков командной работы;
- выявление одаренных детей, обеспечение соответствующих условий для их образования и творческого развития.

Воспитательные:

- формирование и развитие потребностей в техническом творчестве у обучающихся;
- создание и обеспечение необходимых условий для личностного развития, профессионального самоопределения и творческой реализации в инженерной сфере.

Задачи программы на 2-й год обучения:

Образовательные:

- расширить представления обучающихся о возможностях языка Python;
- освоить основные парадигмы программирования используя язык Python;
- научить формулировать и анализировать комбинированные алгоритмы, а также составлять и отлаживать программы;

Развивающие:

- развивать у обучающихся интерес к программированию;
- расширять кругозор обучающихся в области программирования;
- научить моделированию различных процессов реального мира в компьютере;

Воспитательные:

- формирование эмоционально-позитивной установки в оценке собственных возможностей и возможностей других;
- формирование позитивной мотивации к учебе и труду;
- воспитание умения продуктивно общаться и работать в коллективе, команде.

1.6. Адресат программы: программа предназначена для учащихся в возрасте 11-13 лет.

1.7. Форма реализации программы: очные занятия.

1.8. Уровень программы: базовый.

1.9. Срок реализации программы: 2 года. Объем программы составляет 306 часов (первый год обучения: 144 часа - программирование, 18 – модуль «математика»; второй год обучения: 144 часа - программирование).

1.10. Форма организации занятий: индивидуальная, групповая.

1.11. Режим занятий:

Первый год обучения:

- Программирование: 2 раза в неделю по 2 академических часа.
- Модуль «математика»: 18 часов в течении учебного года.

Второй год обучения:

- Программирование: 2 раза в неделю по 2 академических часа.

Режим занятий соответствует санитарно-эпидемиологическим требованиям к учреждениям дополнительного образования детей.

1.12. Виды учебных занятий: лекция, практическая работа, дискуссия, самостоятельная работа, соревнование, проектная деятельность. Основной акцент сделан на практическую часть занятий.

1.13. Ожидаемые результаты обучения

1 год обучения

Предметные результаты:

- знает принципы структурного программирования на языке Python;
- умеет создавать и выполнять программы для решения несложных алгоритмических задач в программе Python;
- умеет формально выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов;
- умеет создавать и выполнять программы для решения несложных алгоритмических задач в программе Python.

Метапредметные компетенции:

- готовность принимать и сохранять цели и задачи учебной деятельности, с помощью наставника находить средства ее осуществления;
- способность с помощью наставника адекватно оценивать правильность выполнения задания и вносить необходимые коррективы;
- способность с помощью наставника планировать свои действия в соответствии с поставленной целью.

Личностные результаты:

- проявляет интерес к информатике и программированию, стремление использовать полученные знания в процессе обучения другим предметам и в жизни;
- проявляет способность связать учебное содержание с собственным жизненным опытом и личными смыслами;
- демонстрирует готовность к осуществлению индивидуальной и коллективной работы.

2 год обучения

Предметные результаты:

- уверенно использует принципы структурного программирования на языке Python;
- знает принципы построения сложных проектов;
- умение самостоятельно составить и записать алгоритмы на языке Python для решения различных задач..

Метапредметные компетенции:

- осуществляет самостоятельное выделение и формулирование познавательной цели;
- использует общих приёмов решения поставленной задачи;
- контролирует и оценивает процесс и результат деятельности.

Личностные результаты:

- проявляет стойкий интерес к информатике и программированию, стремление использовать полученные знания в процессе обучения другим предметам и в жизни;
- понимает значимость подготовки в области информационных технологий в условиях развития информационного общества;
- демонстрирует готовность к самостоятельным поступкам и действиям, принятию ответственности за их результаты.

1.14. Формы аттестации: презентация собственного проекта.

2. Учебно-тематический план

Первый год обучения

№ п/п	Название раздела, темы	Количество часов			Формы аттеста- ции/ контроля
		Всего	Теория	Практика	
1.	Введение в образовательную программу, техника безопасности.				Тестирование
2.	Введение в язык программирования Python				Практическая работа
3.	Линейные алгоритмы и ихреализация на языке Python				Самостоятельная работа
4.	Разветвляющиеся алгоритмы и их реализация на языке Python				Контрольная работа
5.	Кейс 1. «Угадай число»	1	4		Демонстрация решения
6.	Циклические алгоритмы и их реализация на языке Python				Самостоятельная работа
7.	Кейс 2. «Калькулятор»	1	4	10	Демонстрация решения
8.	Этапы решения задач на языке				Практическая работа
9.	Кейс 3. «Игра «Виселица»	18	4	14	Демонстрация решения
10.	Вспомогательные алгоритмы и их реализация на языке Python				Самостоятельная работа
11.	Кейс 4. «Викторина»		4		Демонстрация решения
12.	Формирование общекультурных компетенций				Наблюдение
13.	Модуль «Математика»				Тестирование
14.	Заключительное занятие				Наблюдение
	Итого	162	4	1	

Второй год обучения

№ п/п	Название раздела, темы	Количество часов			Формы аттеста- ции/ контроля
		Всего	Теория	Практика	
1.	Вводное занятие, техника безопасности.				Наблюдение
2.	Инструменты разработчика	16	8	8	Практическая работа
3.	Дополнительные сведения о Python	16	4	12	Практическая работа
4.	Функциональное программирование	16	4	12	Практическая работа

№ п/п	Название раздела, темы	Количество часов			Формы аттеста- ции/ контроля
		Всего	Теория	Практика	
5.	Модули	24	12	12	Практическая ра- бота
6.	Объектно-ориентированное про- граммирование	20	8	12	Практическая ра- бота
7.	Работа с базами данных				Практическая ра- бота
8.	Работа над собственными проек- тами				Презентация про- екта
9.	Формирование общекультурных компетенций				Наблюдение
10.	Заключительное занятие.				Наблюдение
	Итого				

3. Содержание программы (краткое описание тем программы (теоретических и практических видов занятий с указанием часов).

Первый год обучения

1. Введение в образовательную программу, техника безопасности (2 ч):

- Теория (1 ч): Введение в образовательную программу. Ознакомление учащихся с программой, приемами и формами работы. Вводный инструктаж по технике безопасности.
- Практика (1 ч): Знакомство с группой. Игры на командообразование. Первичная диагностика.

2. Введение в язык программирования Python (8 ч):

- Теория (2 ч): Понятия «алгоритм» и «программа». Элементы языка. Структура программы. Операции и переменные. Типы данных. Знакомство с основными типами переменных, синтаксисом языка программирования, основными процедурами ввода исходных данных и вывода результатов.
- Практика (6 ч): Установка среды программирования Python. Работа в IDLE. Практикум по созданию элементарных программ ввода-вывода данных, работа со средой, отладка программ. Анализ возможных синтаксических ошибок. Решение задач.

3. Линейные алгоритмы и их реализация на языке Python (10 ч):

- Теория (2 ч): Знакомство со структурой линейного алгоритма, правилами записи арифметических выражений. Различные типы данных, допустимые операции над. Выполнение операции присваивания в оперативной памяти (далее - ОП) компьютера. Различные типы данных, допустимые операции над вещественными числами и ресурсы оперативной памяти.
- Практика (8 ч): Создание блок-схемы линейного алгоритма. Запись арифметических выражений на языке программирования. Практикум по разработке линейных алгоритмов. Ввод и отладка программ, реализующих линейный алгоритм обработки целых чисел. Анализ готовых линейных программ. Разбор типичных задач с линейной структурой алгоритма. Практикум по реализации линейных алгоритмов вещественных чисел. Ввод и отладка программ, реализующих линейный алгоритм обработки вещественных чисел. Самостоятельная работа по составлению линейного алгоритма, написанию программы, вводу и отладке программного кода, анализу результатов.

4. Разветвляющиеся алгоритмы и их реализация на языке Python (14 ч):

- Теория (2ч): Понятие алгоритма «выбор», графическое изображение. Типовые задачи, использующие алгоритм «выбор», изображение алгоритма в виде блок-схемы. Составление и анализ алгоритмов. Понятие множественного выбора, изображение на блок-схеме.

- Практика (8ч): Составление алгоритмов на «полное ветвление». Практикум по реализации алгоритмов «полное ветвление» и «неполное ветвление». Практикум по составлению алгоритмов для решения задач со сложными условиями и каскадными ветвлениями. Ввод и отладка программ в среде Python. Контрольная работа по составлению алгоритмов, написанию программного кода, вводу и отладке программ в среде Python. Анализ работы.
- 5. Кейс 1**
- 6. Циклические алгоритмы и их реализация на языке Python (24 ч):**
- Теория (4 ч): Циклические алгоритмы. Цикл с предусловием. Цикл с постусловием. Цикл с параметром. Графическое изображение циклов. Инструкции управления циклами. Вложенные циклы.
 - Практика (20 ч): Практикум по составлению блок-схем. Решение задач по теме «Циклические алгоритмы» - составление алгоритмов, написание программного кода, ввод и отладка программ в среде Python. Самостоятельная работа по теме «Циклические алгоритмы». Решение задач по разработке и отладке программ, реализующих вложенные циклы. Анализ разработанных алгоритмов. Оценка эффективности разработанных алгоритмов.
- 7. Кейс 2**
- 8. Этапы решения задач на языке Python (6 ч):**
- Теория (1 ч): Этапы и особенности решения задачи на компьютере. Этапы создания алгоритма. Использование принципа последовательного конструирования алгоритма.
 - Практика (5 ч): Разработка алгоритмов. Написание программного кода, ввод и отладка программ в среде Python.
- 9. Кейс 3**
- 10. Вспомогательные алгоритмы и их реализация на языке Python (12 ч):**
- Теория (3 ч): Понятие вспомогательного алгоритма. Формат записи вспомогательного алгоритма в виде функции. Польза функций при решении задач. Модульность и повторное использование кода. Понятие рекурсии. Рекурсивные алгоритмы и их особенности.
 - Практика (9 ч): Практикум по программированию подзадач с помощью отдельных функций. Практикум по использованию рекурсии. Итоговая самостоятельная работа по теме «Вспомогательные алгоритмы и их реализация на языке Python».
- 11. Кейс 4**
- 12. Формирование общекультурных компетенций (4 ч):**
- Практика (4 ч): Участие в мероприятиях по формированию общекультурных компетенций.
- 13. Модуль «Математика» (18 ч) (Приложение 1):**
- Теория (8 ч): Введение в теорию графов. Использование графов для представления информации. Элементы теории множеств: понятие множества, операции над ними. Основы теории вероятности и математической статистики. Математические модели в вычислениях.
 - Практика (10 ч): Использование вычислительных сервисов и систем для решения прикладных задач. Работа с сервисами Wolfram Alpha, Nigma Mathematics и других для выполнения расчетов.
- 14. Заключительное занятие. Подведение итогов обучения (2 ч.):**
- Теория (1 ч.). Обзор пройденного материала.
 - Практика (1 ч.). Обсуждение итогов обучения.

Второй год обучения

- 1. Вводное занятие, техника безопасности (2 ч):**
- Теория (1 ч): Ознакомление учащихся с программой на второй год обучения, приемами и формами работы. Инструктаж по технике безопасности.
 - Практика (1 ч): Знакомство с группой. Игры на командообразование.
- 2. Инструменты разработчика (16 ч):**
- Теория (8 ч): Инструментальная среда разработчика PyCharm Community, основные сведения. Логгирование. Отладка. Средства организации командной работы. Документирование кода.

- Практика (8 ч): Актуализация знаний. Установка PyCharm Community. Работа с PyCharm Community. Практикум по созданию программ ввода-вывода данных. Отладка программ. Решение задач.
- 3. Дополнительные сведения о Python (16 ч):**
 - Теория (4 ч): Строки. Списки. Вложенные списки. Словари. Файлы.
 - Практика (12 ч): Работа со строками. Работа со списками. Работа со словарями. Использование файлов. Практикум по решению задач.
- 4. Функциональное программирование (16 ч):**
 - Теория (4 ч): Функция как способ структурировать код. Функциональное программирование. Лямбда функции.
 - Практика (12 ч): Создание и использование функций. Использование лямбда функций. Практикум по решению задач с использованием функционального программирования.
- 5. Модули (24 ч):**
 - Теория (22 ч): Модульная структура Python. Экскурс по возможностям модулей. Модули random, math. Модуль Tkinter. Модуль PyGame.
 - Практика (12 ч): Использование модулей random, math. Использование модуля Tkinter. Использование модуля PyGame. Практикум использованию модулей.
- 6. Объектно-ориентированное программирование (20 ч):**
 - Теория (8 ч): Введение в ООП. Классы и Объекты. Полиморфизм. Наследование. Инкапсуляция.
 - Практика (12 ч): Манипуляция с объектами. Задачи на пройденные парадигмы ООП, написание программы из списка предложенных.
- 7. Работа с базами данных (20 ч):**
 - Теория (8 ч): Базы данных – общие сведения. Виды баз данных. Запросы.
 - Практика (12 ч): Реализация базы данных на примере sqlite. Использование запросов. Практикум по созданию и использованию баз данных.
- 8. Работа над собственными проектами (24 ч):**
 - Теория (4 ч): Постановка задачи. Дополнительные сведения для реализации проектов.
 - Практика (20 ч): Выбор тем для разработки. Разработка технического задания. Выполнение технического задания. Подготовка к защите проекта. Презентация и защита проекта.
- 9. Формирование общекультурных компетенций (4 ч):**
 - Практика (4 ч): Участие в мероприятиях по формированию общекультурных компетенций.
- 10. Заключительное занятие. (2 ч):**
 - Теория (1 ч): Подведение итогов обучения.
 - Практика (1 ч): Рефлексия по пройденной программе.

4. Комплекс организационно-педагогических условий

4.1 Календарный учебный график (приложение 2 к программе)

4.2. Ресурсное обеспечение программы

- персональный компьютер с доступом в сеть интернет – на каждого обучающегося;
- компьютер для учителя, доска, проектор;
- программное обеспечение Python v.3;
- инструментальная среда Wing 101;
- инструментальная среда PyCharm Community.

4.3. Методическое обеспечение программы

Для реализации программы используются следующие формы и методы обучения.

Формы обучения: лекция, практикум, работа со специальной литературой, мини-конференция, обсуждение вариантов решения задачи.

Методы обучения:

- Словесные (указания педагога, объяснение нового материала (лекции), индивидуальная консультация)

- Работа с литературными источниками (книги, журналы, публикации) и с электронными источниками информации (Интернет).
- Практическая работа (задания, тесты, составление алгоритмов, схем, решение задач).
- Проблемного обучения (самостоятельный поиск учащимися ответа на поставленную проблему).

Система оценки и фиксации образовательных результатов

В процессе обучения осуществляется контроль за уровнем сформированности знаний, умений и навыков.

Система контроля за усвоением учащимися программы складывается из следующих элементов: опрос, зачеты, самостоятельные работы, соревнования (где можно определить уровень каждого игрока и команды), конкурсы, тесты. Результаты проверки уровня усвоения программы фиксируются педагогом в специально разработанных листах учебных достижений:

В течение обучения осуществляется ряд диагностических срезов по определению уровня усвоения учащимися программы:

- входная диагностика посредством бесед, анкетирования, тестов, где выясняется начальный уровень знаний, умений и навыков учащихся, а так же выявляются их творческие способности.
- промежуточные диагностики (в середине и конце первого года обучения, в середине второго года обучения) позволяют выявить достигнутый на данном этапе уровень ЗУН учащихся, в соответствии с пройденным материалом программы. Предлагаются контрольные тесты, выполнение практических заданий.
- итоговая диагностика проводится в конце обучения (итоговый показ творческих проектов) и предполагает комплексную проверку образовательных результатов по всем ключевым направлениям. Данный контроль позволяет проанализировать степень усвоения программы учащимися.

Результаты контроля фиксируются в диагностической карте:

**Диагностическая карта по образовательной программе
дополнительного образования детей**

Педагог _____

д/о _____

Группа _____

№ _____

_____ год

обучения

Уровень теоретических знаний и практических умений и навыков
Форма проведения _____

№ п/п	ФИО обучающегося	Оценка теоретиче- ских знаний	Оценка практиче- ских умений и навыков	Итоговая оценка
1				
2				
3				
4				
5				
6				
...				

Сводные показатели освоения дополнительной общеобразовательной программы

Уровни освоения программы (в %):

Низкий _____

Средний _____

Высокий _____

Оценка уровней освоения программы

Уровни / количество %	Параметры	Общие критерии оценки результативности обучения	Показатели
Высокий уровень/ 80-100%	Теоретические знания.	Оценка уровня теоретических знаний по программным требованиям: широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии	Учащийся освоил материал в полном объеме. Знает и понимает значение терминов, самостоятельно ориентируется в содержании материала по темам. Учащийся заинтересован, проявляет устойчивое внимание к выполнению заданий.
	Практические умения и навыки.	Оценка уровня практической подготовки учащихся: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность практической деятельности	Способен применять практические умения и навыки во время выполнения самостоятельных заданий. Правильно и по назначению применяет инструменты. Работу аккуратно доводит до конца. Может оценить результаты выполнения своего задания и дать оценку работы своего товарища.
Средний уровень/ 50%-79%	Теоретические знания.	Оценка уровня теоретических знаний по программным требованиям: широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии	Учащийся освоил базовые знания, ориентируется в содержании материала по темам, иногда обращается за помощью к педагогу. Учащийся заинтересован, но не всегда проявляет устойчивое внимание к выполнению задания.
	Практические умения и навыки.	Оценка уровня практической подготовки учащихся: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность практической деятельности	Владеет базовыми навыками и умениями, но не всегда может выполнить самостоятельное задание, затрудняется и просит помощи педагога. В работе допускает небрежность, делает ошибки, но может устранить их после наводящих вопросов или самостоятельно. Оценить результаты своей деятельности может с подсказкой педагога.

Уровни / количество %	Параметры	Общие критерии оценки результативности обучения	Показатели
Низкий уровень / Ниже 50%	Теоретические знания.	Оценка уровня теоретических знаний по программным требованиям: широта кругозора, свобода восприятия теоретической информации, развитость практических навыков работы со специальной литературой, осмысленность и свобода использования специальной терминологии	Владеет минимальными знаниями, ориентируется в содержании материала по темам только с помощью педагога.
	Практические умения и навыки.	Оценка уровня практической подготовки учащихся: соответствие развития уровня практических умений и навыков программным требованиям, свобода владения специальным оборудованием и оснащением, качество выполнения практического задания, технологичность практической деятельности	Владеет минимальными начальными навыками и умениями. Учащийся способен выполнять каждую операцию только с подсказкой педагога или товарищей. Не всегда правильно применяет необходимый инструмент или не использует во все. В работе допускает грубые ошибки, не может их найти их даже после указания. Не способен самостоятельно оценить результаты своей работы.

5. Список литературы

Список литературы для педагога:

1. Джейсон Бриггс. Python для детей. Самоучитель по программированию. — М.: Манн, Иванов и Фербер, 2017.
2. Доусен, М. «Программируем на Python» изд. «Питер», серия Бестселлеры O'Reilly, 2016. — 416 с.
3. Задачи по программированию / под ред. С. М. Окулова — М.: БИНОМ. Лаборатория знаний, 2014. — 341 с.
4. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. 2-е изд.— СПб.: Питер, 2018. - 496 с.
5. Информатика. Задачник - практикум в 2т. / Под ред. И.Г. Семакина, Е.К. Хеннера: Том 1. — М.: Бином. Лаборатория Знаний, 2002. — 297 с.
6. Лутц, М. «Изучаем Python», 4 издание, — Пер. с англ. — СПб.: Символ-Плюс, 2011. — 1280 с.
7. Методика преподавания информатики: Учеб. Пособие для студ. Пед. вузов / М.П. Лапчик, И.Г. Семакин. Е.К. Хеннер; Под общей ред. М.П. Лапчик. - М.: -Издательский центр «Академия», 2001. — 624 с.
8. Окулов, С. М. Основы программирования. - М.: ЮНИМЕДИАСТАЙЛ, 2002. - 424 с.
9. Поляков, К.Ю. Язык Python: избранные алгоритмы Информатика, №9, №10/2014, С. 18.
10. Поляков, К.Ю., Язык Python глазами учителя. Информатика, № 9/2014, С. 4.
11. Россум, Г. и др. Язык программирования Python, 2001 — 454 с. 11. Саммерфильд, Марк. «Python на практике», пер. А. Слинкин //изд. «ДМК-Пресс», 2014. — 338 с.
12. Учим Python, делая крутые игры / Эл Свейгарт; перю с англ. М.А. Райтмана. - Москва: Эксмо, 2018. - 416 с.
13. Федоров, Д. Ю. Основы программирования на примере языка Python: учеб.пособие / Д. Ю. Федоров. — СПб., 2016. — 176 с.
14. Э.Фримен, К. Сьерра, Б.Бейтс. Паттерны проектирования. — СПб.: Питер, 2011. - 656 с.

Список рекомендуемой литературы для учащихся и родителей:

1. Джейсон Бриггс. Python для детей. Самоучитель по программированию. — М.: Манн, Иванов и Фербер, 2017.
2. Изучаем Python. Программирование игр, визуализация данных, веб-приложения. 2-е изд.— СПб.: Питер, 2018. — 496 с.
3. Учим Python, делая крутые игры / Эл Свейгарт; перю с англ. М.А. Райтмана. - Москва: Эксмо, 2018. - 416 с.
4. Задачи по программированию / под ред. С. М. Окулова — М.: БИНОМ. Лаборатория знаний, 2014. — 341 с.
5. Лутц, М. «Изучаем Python», 4 издание, — Пер. с англ. — СПб.: Символ-Плюс, 2011. — 1280 с.
6. Поляков, К.Ю. Язык Python: избранные алгоритмы Информатика, №9, №10/2014, С. 18.
7. Россум, Г. и др. Язык программирования Python, 2001 — 454 с. 11. Саммерфильд, Марк. «Python на практике», пер. А. Слинкин //изд. «ДМК-Пресс», 2014. — 338 с.

Модуль «Математика»

Дополнительная общеобразовательная общеразвивающая программа технической направленности «Математика. Вводный модуль» призвана формировать метапредметные связи, целостное видение у обучающихся современных методов, задач и направлений исследований; служит для определения будущих интересов обучающихся. Модуль предназначен для развития логики, формирования структурированного мышления, применения математических знаний на практике. Модуль включает в себя введение в основные разделы геометрии, теории множеств, теории вероятностей, теории графов. Также значительный акцент уделяется изучению базы знаний Wolfram Alpha. В результате освоения программы учащиеся будут способны применять базовые знания по математике для решения проектных и практических задач.

Hard-компетенции:

- умение выполнять математические расчеты устно и с помощью вычислительной техники;
- умение использовать современные прикладные сервисы, такие как Wolfram Alpha и др., для решения задач;
- знание и понимание основ комбинаторики, теории множеств, математической логики и их практического применения;
- знание и понимание основ теории вероятности и математической статистики их практического применения;
- знание и понимание систем координат и принципов построения графиков функции и их исследования;
- знание и понимание основ теории графов и ее основных алгоритмов, таких как поиск кратчайшего пути, жадный алгоритм и др.;
- понимание принципов решения задач дискретной математики, в частности, транспортных задач, задач коммивояжера и др.;
- понимание основ построения математических моделей с использованием численных методов, формирование навыков построения и описания математических моделей и подбора метода их решения.

Ожидаемые результаты.

Предметные:

- умение выполнять вычисления;
- умение использовать современные программные комплексы для осуществления вычислений;
- умение структурировать решаемую задачу, определять оптимальный алгоритм решения, реализовывать расчеты на каждом этапе;
- знание основных типов математических моделей, понимание методологии их применения в исследованиях;
- умение выполнять расчеты по формулам.

Модуль 1. Вычислительный практикум

Теория: Введение в теорию графов. Использование графов для представления информации. Элементы теории множеств: понятие множества, операции над ними. Основы теории вероятности и математической статистики. Математические модели в вычислениях.

Практика: Использование вычислительных сервисов и систем для решения прикладных задач. Работа с сервисами Wolfram Alpha, Nigma Mathematics и других для выполнения расчетов.

№ п/п	Название раздела, темы	Количество часов			Формы аттестации/ контроля
		Всего	Тео- рия	Прак- тика	
1	Понятие графа. Использование графов для визуализации информации.	2	2	-	Обсуждение, ведение конспекта
2	Понятие множества. Операции над множествами.	2	2	-	

3	Основы теории вероятности и математической статистики.	2	2	-	
4	Понятие математической модели. Обзор наиболее известных математических моделей и их практическое применение.	1	1	-	Обсуждение, ведение конспекта
5	Электронные таблицы: назначение, возможности, принципы работы на примере MS Excel.	5	1	4	Выполнение расчетно-вычислительного задания
6	Поисково-решающие системы и нейросети: Wolfram Alpha, Nigma Mathematics и др.	6	-	6	
Итого:		18	8	10	

Кейсы

В качестве кейс-заданий учащимся можно предлагать разработку программных продуктов от простых до сложносоставных программ, в различных областях. Далее представлены условные ситуационные задачи в рамках которых учащиеся должны создать тот или иной программный продукт.

Кейс 1. «Угадай число».

1. **Тема кейса:** Разработка программы, которая способна угадывать задуманное число, эффективным способом.
2. **Описание кейса:** Вы решили самостоятельно написать программу, которая способна угадывать целое число, которое загадал пользователь из заданного диапазона. При этом необходимо создать программу которая решает поставленную задачу эффективным способом.
3. **Цели и задачи кейса:**

- **цель:** разработать и реализовать программу «Угадай число»;

- **задачи:**

1 уровень. Найдите информацию о различных способах решения поставленной задачи.

2 уровень. Найдите информацию о понятии «Эффективность алгоритма». Проведите анализ способов решения задачи с точки зрения эффективности алгоритма. Выберите наиболее эффективной алгоритм для последующей реализации.

3 уровень. Реализуйте выбранный алгоритм на языке программирования Python.

4 уровень. Осуществите тестирование созданной программы, при необходимости внесите изменения.

Категория кейса. Базовый.

Место кейса в структуре модуля. Базовый.

Количество учебных часов. 12 часов.

Продолжительность одного занятия. 45 минут.

2 занятия		2 занятия		6 занятий	
Цель: настроить учащихся на совместную работу, командное мышление.		Цель: Научиться планировать разработку программы, навести учащихся на необходимость детальной проработки кейса		Цель: создать условия учащимся для решения кейса	
Деление на группы. Определяют проблему. Мозговой штурм. Уч-ся формулируют цель своей работы и средства достижения цели. Осуществляют поиск	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: искать информацию в свободных источниках и структурировать ее.	Знакомство с этапами разработки программных продуктов	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: Оценка эффективности алгоритма	Реализация выбранного алгоритма	Soft: 4К-компетенции, аргументированно отстаивать свою точку зрения, организаторские качества, комбинировать, видоизменять и улучшать идеи. Умение грамотно письменно формулировать свои мысли.

необходимой информации.					Hard: Умение реализовывать и отлаживать алгоритма на языке программирования Python.
2 занятия					
Цель: реализовать возможность учащихся продемонстрировать решения кейса					
Создание презентаций. Представление решений кейсов экспертной группе. Рефлексия.	Soft: командная работа, коммуникативность, основы ораторского искусства, опыт публичных выступлений, умение отвечать на вопросы, умение грамотно отстаивать свою точку зрения, умение оценивать себя. Hard: Демонстрация решений кейса, получение внешней оценки				

- **Метод работы с кейсом.** Метод проектов.
- **Минимально необходимый уровень входных компетенций.** Базовые компетенции в области алгоритмизации и программирования.

4. Предполагаемые результаты кейса:

- **личностные и социальные (soft):** умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов, умение выслушивать собеседника и вести диалог; умение планировать учебное сотрудничество с педагогом и сверстниками: определять цели, функций участников, способов взаимодействия, умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация; умение управлять поведением партнера: контроль, коррекция, оценка его действий, умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации, владение монологической и диалогической формами речи.
 - **практические умения (hard):** опыт создания программы, ее тестирования, опыт работы в среде программирования, опыт оценивания сложности алгоритмов, опыт разработки презентационных материалов для демонстрации созданного продукта.
5. **Процедуры и формы выявления образовательного результата.** Демонстрация решений кейса. Экспертные листы..
 6. **Ресурсы и материалы:**
 - Компьютер с выходом в интернет и установленной средой программирования Python.
 7. **Список рекомендуемых источников.** См. пункт «Литература и информационные ресурсы для учащихся» данной дополнительной образовательной программы.

Кейс 2. «Калькулятор».

8. **Тема кейса:** Разработка программы-калькулятора для проведения арифметических расчётов.
9. **Описание кейса:** Вы решили самостоятельно написать программу, которая способна работать в качестве интерактивного калькулятора и производить арифметические вычисления.

10. Цели и задачи кейса:

- **цель:** разработать и реализовать программу «Калькулятор»;
- **задачи:**

1 уровень. Найдите информацию о различных способах решения поставленной задачи.

2 уровень. Разработайте интерфейс взаимодействия с пользователем и алгоритм работы программы.

3 уровень. Реализуйте выбранные решения на языке программирования Python.

4 уровень. Осуществите тестирование созданной программы, при необходимости внесите изменения.

Категория кейса. Базовый.

Место кейса в структуре модуля. Базовый.

Количество учебных часов. 14 часов.

Продолжительность одного занятия. 45 минут.

2 занятия		2 занятия		8 занятий	
Цель: настроить учащихся на совместную работу, командное мышление.		Цель: Научиться планировать разработку программы, навести учащихся на необходимость детальной проработки кейса		Цель: создать условия учащимся для решения кейса	
Деление на группы. Определяют проблему. Мозговой штурм. Учащиеся формулируют цель своей работы и средства достижения цели. Осуществляют поиск необходимой информации.	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: искать информацию в свободных источниках и структурировать ее.	Знакомство с этапами разработки программных продуктов	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: Оценка эффективности алгоритма	Реализация выбранного алгоритма и интерфейса.	Soft: 4К-компетенции, аргументированно отстаивать свою точку зрения, организаторские качества, комбинировать, видоизменять и улучшать идеи. Умение грамотно письменно формулировать свои мысли. Hard: Умение реализовывать и отлаживать алгоритма на языке программирования Python.
2 занятия					
Цель: реализовать возможность учащихся продемонстрировать решения кейса					
Создание презентаций. Представление решений кейсов экспертной группе. Рефлексия.	Soft: командная работа, коммуникативность, основы ораторского искусства, опыт публичных выступлений, умение отвечать на вопросы, умение грамотно отстаивать свою точку зрения, умение оценивать себя.				

	Hard: Демонстрация решений кейса, получение внешней оценки		
--	---	--	--

- **Метод работы с кейсом.** Метод проектов.
- **Минимально необходимый уровень входных компетенций.** Базовые компетенции в области алгоритмизации и программирования.

11. Предполагаемые результаты кейса:

- **личностные и социальные (soft):** умение аргументировать свою точку зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов, умение выслушивать собеседника и вести диалог; умение планировать учебное сотрудничество с педагогом и сверстниками: определять цели, функций участников, способов взаимодействия, умение разрешать конфликты: выявление, идентификация проблемы, поиск и оценка альтернативных способов разрешения конфликта, принятие решения и его реализация; умение управлять поведением партнера: контроль, коррекция, оценка его действий, умение с достаточной полнотой и точностью выражать свои мысли в соответствии с задачами и условиями коммуникации, владение монологической и диалогической формами речи.
- **практические умения (hard):** опыт создания программы, ее тестирования, опыт работы в среде программирования, опыт оценивания сложности алгоритмов, опыт разработки презентационных материалов для демонстрации созданного продукта.

12. Процедуры и формы выявления образовательного результата. Демонстрация решений кейса. Экспертные листы.

13. Ресурсы и материалы:

- Компьютер с выходом в интернет и установленной средой программирования Python.

14. Список рекомендуемых источников. См. пункт «Литература и информационные ресурсы для учащихся» данной дополнительной образовательной программы.

Кейс 3. «Игра «Виселица»

1. **Тема кейса:** Программный продукт «Игра «Виселица»
2. **Описание кейса:** Необходимо создать игру «Виселица» по классическому сценарию. Как вывести изображения виселицы в текстовом режиме? Как обеспечить разнообразие слов в игре и их случайный выбор?
Необходимо разработать программный продукт типа «Игра «Виселица», который мог бы позволить пользователям сыграть несколько уникальных игр подряд.

3. Цели и задачи кейса:

- **цель:** разработка программного продукта для определения уровня эрудиции пользователя;
- **задачи:**

1 уровень. Найдите информацию о том, что такое «Игра «Виселица»?

2 уровень. Проанализируйте правила данной игры.

3 уровень. Разработайте базу слов для игры по выбранной тематике. Разработайте модель работы приложения. Разработайте программные интерфейсы.

4 уровень. Разработайте приложение ««Игра «Виселица»» на языке Python, осуществите подготовку приложения к распространению.

Категория кейса. Углубленный.

Место кейса в структуре модуля. Продвинутой.

Количество учебных часов. 18 часов.

Продолжительность одного занятия. 45 минут.

2 занятия		4 занятия		10 занятий	
Цель: настроить учащихся на совместную работу, командное мышление.		Цель: Научиться планировать разработку программного продукта, навести учащихся на необходимость детальной проработки кейса		Цель: создать условия учащимся для решения кейса	
Деление на группы. Определяют проблему. Мозговой штурм. Уч-ся формулируют цель своей работы и средства достижения цели. Осуществляют поиск необходимой информации.	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: искать информацию в свободных источниках и структурировать ее.	Знакомство с этапами разработки программных продуктов	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: Разработка архитектуры приложения «Викторина»	Выполнение простых заданий по формированию интерфейса приложения, по работе с файлами, по отображению текстовой информации, по обработке ввода данных.	Soft: 4К-компетенции, аргументированно отстаивать свою точку зрения, организаторские качества, комбинировать, видоизменять и улучшать идеи. Умение грамотно письменно формулировать свои мысли. Hard: Умение разрабатывать полностью завершённый программный продукт, разрабатывать логику работы приложения и его внешний вид. Готовить к распространению программный продукт.
2 занятия					
Цель: реализовать возможность учащихся продемонстрировать решения кейса					
Создание презентаций. Представление решений кейсов экспертной группе. Рефлексия.	Soft: командная работа, коммуникативность, основы ораторского искусства, опыт публичных выступлений, умение отвечать на вопросы, умение грамотно отстаивать свою точку зрения, умение оценивать себя. Hard: Демонстрация решений кейса, получение внешней оценки				

Метод работы с кейсом. Метод проектов.

Минимально необходимый уровень входных компетенций. Базовые компетенции в области алгоритмизации и программирования.

4. Предполагаемые результаты кейса:

- **личностные и социальные (soft):** умение генерировать идеи указанными методами, слушать и слышать собеседника, аргументированно отстаивать свою точку зрения, искать информацию в свободных источниках и структурировать ее. Умение комбинировать, видоизменять и улучшать идеи. Командная работа. Организаторские качества. Умение грамотно письменно формулировать свои мысли. Критическое мышление и умение объективно оценивать результаты своей работы. Основы ораторского искусства. Опыт публичных выступлений. Формирование навыков управления проектом.
 - **практические умения (hard):** опыт проектирования и разработки программных продуктов; поиск информации; работа в программе для создания презентаций; разработка интерфейса, опыт работы со строками.
5. **Процедуры и формы выявления образовательного результата.** Демонстрация решений кейса. Экспертные листы. Тестирование по hard skills.
 6. **Ресурсы и материалы:**
Компьютер с выходом в интернет и установленной средой программирования Python.
 7. **Список рекомендуемых источников:** См. пункт «Литература и информационные ресурсы для учащихся» данной дополнительной образовательной программы.

Кейс 4. «Викторина»

1. **Тема кейса:** Программный продукт «Викторина»
 2. **Описание кейса:** В социальных сетях в последнее время стали популярны приложения, которые позволяют пользователю узнать свой уровень эрудиции в различных областях. Для проверки эрудиции пользователю достаточно активировать такую программу в социальной сети и он начнет получать вопросы из различных областей знаний. Отвечая на эти вопросы пользователь может и сам и с помощью программы оценить свой уровень эрудиции. Единственный минус этих приложений — это тесная связь с социальной сетью, которой многие пользователи хотели бы избежать и необходимость доступа в интернет. Необходимо разработать программный продукт типа «викторина», который мог бы позволить пользователям отвечать на вопросы без необходимости постоянного доступа в интернет. Это можно реализовать только путем формирования оффлайн базы вопросов.
 3. **Цели и задачи кейса:**
 - **цель:** разработка программного продукта для определения уровня эрудиции пользователя;
 - **задачи:**
 - 1 уровень. Найдите информацию о том, что такое викторина?
 - 2 уровень. Проанализируйте правила создания викторины? Выполните соответствующие практические задания.
 - 3 уровень. Разработайте базу вопросов по выбранной тематике. Разработайте модель работы приложения. Разработайте программные интерфейсы.
 - 4 уровень. Разработайте приложение «Викторина» на языке Python, осуществите подготовку приложения к распространению.
- Категория кейса.** Углубленный.
- Место кейса в структуре модуля.** Продвинутый.
- Количество учебных часов.** 18 часов.
- Продолжительность одного занятия.** 45 минут.

2 занятия	4 занятия	10 занятий
Цель: настроить учащихся на совместную работу, командное мышление.	Цель: Научиться планировать разработку программного продукта, навести	Цель: создать условия учащимся для решения кейса

		учащихся на необходимость детальной проработки кейса			
Деление на группы. Определяют проблему. Мозговой штурм. Уч-ся формулируют цель своей работы и средства достижения цели. Осуществляют поиск необходимой информации.	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: искать информацию в свободных источниках и структурировать ее.	Знакомство с этапами разработки программных продуктов	Soft: 4К-компетенции, умение генерировать идеи указанными методами, слушать и слышать собеседника. Hard: Разработка архитектуры приложения «Викторина»	Выполнение простых заданий по формированию интерфейса приложения, по работе с файлами, по отображению графической информации, по обработке ввода данных.	Soft: 4К-компетенции, аргументированно отстаивать свою точку зрения, организаторские качества, комбинировать, видоизменять и улучшать идеи. Умение грамотно письменно формулировать свои мысли. Hard: Умение разрабатывать полностью завершённый программный продукт, разрабатывать логику работы приложения и его внешний вид. Готовить к распространению программный продукт.
2 занятия					
Цель: реализовать возможность учащихся продемонстрировать решения кейса					
Создание презентаций. Представление решений кейсов экспертной группе. Рефлексия.	Soft: командная работа, коммуникативность, основы ораторского искусства, опыт публичных выступлений, умение отвечать на вопросы, умение грамотно отстаивать свою точку зрения, умение оценивать себя. Hard: Демонстрация решений кейса, получение внешней оценки				

Метод работы с кейсом. Метод проектов.

Минимально необходимый уровень входных компетенций. Базовые компетенции в области алгоритмизации и программирования.

4. Предполагаемые результаты кейса:

- **личностные и социальные (soft):** умение генерировать идеи указанными методами, слушать и слышать собеседника, аргументированно отстаивать свою точку зрения, искать информацию в свободных источниках и структурировать ее. Умение комбинировать, видоизменять и улучшать идеи. Командная работа. Организаторские качества. Умение грамотно письменно формулировать свои мысли. Критическое мышление и

умение объективно оценивать результаты своей работы. Основы ораторского искусства. Опыт публичных выступлений. Формирование навыков управления проектом.

- **практические умения (hard):** опыт проектирования и разработки программных продуктов; поиск информации; работа в программе для создания презентаций; разработка интерфейса.

5. **Процедуры и формы выявления образовательного результата.** Демонстрация решений кейса. Экспертные листы. Тестирование по hard skills.
6. **Ресурсы и материалы:**
Компьютер с выходом в интернет и установленной средой программирования Python.
7. **Список рекомендуемых источников:** См. пункт «Литература и информационные ресурсы для учащихся» данной дополнительной образовательной программы.

Календарный учебный график

Педагоги: Рзаев Р.А., Шуныгина И.В.

Период обучения – 1 год

Кол-во учебных недель - 36

Количество часов – 144

Режим проведения занятий: 2 раз в неделю по 2 часа (45 минут)

Праздничные и выходные дни (по производственному календарю по шестидневной рабочей неделе).

Каникулярный период:

- осенние каникулы;
- зимние каникулы;
- весенние каникулы;
- дополнительные каникулы.

Во время осенних и зимних школьных каникул время занятий в объединении могут быть изменены.

№ п/п	Дата	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1.			ЛК/ПР	2	Введение в образовательную программу, техника безопасности.		Тестирование
2.			ЛК	2	Введение в язык программирования Python		Наблюдение
3.			ПР	2	Введение в язык программирования Python		Наблюдение
4.			ПР	2	Введение в язык программирования Python		Наблюдение
5.			ПР	2	Введение в язык программирования Python		Практическая работа
6.			ЛК	2	Линейные алгоритмы и их реализация на языке Python		Наблюдение
7.			ПР	2	Линейные алгоритмы и их реализация на языке Python		Наблюдение
8.			ПР	2	Линейные алгоритмы и их реализация на языке Python		Наблюдение
9.			ПР	2	Линейные алгоритмы и их реализация на языке Python		Наблюдение
10.			ПР	2	Линейные алгоритмы и их реализация на языке Python		Самостоятельная работа
11.			ЛК/ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Наблюдение
12.			ЛК/ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Наблюдение

№ п/п	Дата	Время проведения занятия	Форма занятия	Кол- во часов	Тема занятия	Место проведе- ния	Форма контроля
13.			ЛК/ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Наблюдение
14.			ЛК/ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Наблюдение
15.			ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Наблюдение
16.			ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Наблюдение
17.			ПР	2	Разветвляющиеся алгоритмы и их реализация на языке Python		Контрольная работа
18.			ЛК/ПР	2	Кейс 1. «Угадай число»		Наблюдение
19.			ЛК/ПР	2	Кейс 1. «Угадай число»		Наблюдение
20.			ЛК/ПР	2	Кейс 1. «Угадай число»		Наблюдение
21.			ЛК/ПР	2	Кейс 1. «Угадай число»		Наблюдение
22.			ПР	2	Кейс 1. «Угадай число»		Наблюдение
23.			ПР	2	Кейс 1. «Угадай число»		Демонстрация решения
24.			ПР	2	Формирование общекультурных компетенций		Наблюдение
25.			ЛК/ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение
26.			ЛК/ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение
27.			ЛК/ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение
28.			ЛК/ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение
29.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение
30.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение
31.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюдение

№ п/п	Дата	Время проведения занятия	Форма занятия	Кол- во часов	Тема занятия	Место проведе- ния	Форма контроля
32.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюде- ние
33.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюде- ние
34.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюде- ние
35.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Наблюде- ние
36.			ПР	2	Циклические алгоритмы и их реализация на языке Python		Практиче- ская работа
37.			ЛК/ПР	2	Кейс 2. «Калькулятор»		Наблюде- ние
38.			ЛК/ПР	2	Кейс 2. «Калькулятор»		Наблюде- ние
39.			ЛК/ПР	2	Кейс 2. «Калькулятор»		Наблюде- ние
40.			ЛК/ПР	2	Кейс 2. «Калькулятор»		Наблюде- ние
41.			ПР	2	Кейс 2. «Калькулятор»		Наблюде- ние
42.			ПР	2	Кейс 2. «Калькулятор»		Наблюде- ние
43.			ПР	2	Кейс 2. «Калькулятор»		Демонстра- ция реше- ния
44.			ЛК/ПР	2	Этапы решения задач на языке Python		Наблюде- ние
45.			ПР	2	Этапы решения задач на языке Python		Наблюде- ние
46.			ПР	2	Этапы решения задач на языке Python		Практиче- ская работа
47.			ЛК/ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
48.			ЛК/ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
49.			ЛК/ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
50.			ЛК/ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
51.			ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
52.			ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
53.			ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние
54.			ПР	2	Кейс 3. «Игра «Виселица»		Наблюде- ние

№ п/п	Дата	Время проведения занятия	Форма занятия	Кол- во часов	Тема занятия	Место проведе- ния	Форма контроля
55.			ПР	2	Кейс 3. «Игра «Виселица»		Демонстра- ция реше- ния
56.			ПР	2	Формирование общекуль- турных компетенций		Наблюде- ние
57.			ЛК/ПР	2	Вспомогательные алго- ритмы и их реализация на языке Python		Наблюде- ние
58.			ЛК/ПР	2	Вспомогательные алго- ритмы и их реализация на языке Python		Наблюде- ние
59.			ЛК/ПР	2	Вспомогательные алго- ритмы и их реализация на языке Python		Наблюде- ние
60.			ПР	2	Вспомогательные алго- ритмы и их реализация на языке Python		Наблюде- ние
61.			ПР	2	Вспомогательные алго- ритмы и их реализация на языке Python		Наблюде- ние
62.			ПР	2	Вспомогательные алго- ритмы и их реализация на языке Python		Самостоя- тельная ра- бота
63.			ЛК/ПР	2	Кейс 4. «Викторина»		Наблюде- ние
64.			ЛК/ПР	2	Кейс 4. «Викторина»		Наблюде- ние
65.			ЛК/ПР	2	Кейс 4. «Викторина»		Наблюде- ние
66.			ЛК/ПР	2	Кейс 4. «Викторина»		Наблюде- ние
67.			ПР	2	Кейс 4. «Викторина»		Наблюде- ние
68.			ПР	2	Кейс 4. «Викторина»		Наблюде- ние
69.			ПР	2	Кейс 4. «Викторина»		Наблюде- ние
70.			ПР	2	Кейс 4. «Викторина»		Наблюде- ние
71.			ПР	2	Кейс 4. «Викторина»		Демонстра- ция реше- ния
72.			ЛК/ПР	2	Заключительное занятие		Наблюде- ние