

МИНИСТЕРСТВО ОБРАЗОВАНИЯ ИРКУТСКОЙ ОБЛАСТИ

**ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ИРКУТСКОЙ ОБЛАСТИ
«ИРКУТСКИЙ КОЛЛЕДЖ АВТОМОБИЛЬНОГО ТРАНСПОРТА И
ДОРОЖНОГО СТРОИТЕЛЬСТВА»**

ЦЕНТР ЦИФРОВОГО ОБРАЗОВАНИЯ ДЕТЕЙ «IT-КУБ. ИРКУТСК»



УТВЕРЖДЕНО

Директор

_____ А.А. Русанов

приказ № _____ от _____ 2024 г.

Дополнительная общеобразовательная программа

«ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON»

Направленность: техническая

Категория обучающихся: 14-17 лет

Объем: 90 часов

Форма обучения: заочная с применением ДОТ

г. Иркутск, 2024

Разработчик/разработчики

_____/_____
Подпись Ф.И.О.

_____/_____
Подпись Ф.И.О.

Методист центра цифрового образования
детей «IT-куб»

_____/_____
Подпись Ф.И.О.

Руководитель центра цифрового
образования детей «IT-куб»

_____/_____
Подпись Ф.И.О.

Заместитель директора по дополнительному
профессиональному образованию

_____/_____
Подпись Ф.И.О.

1. ОБЩАЯ ХАРАКТЕРИСТИКА ПРОГРАММЫ

1.1. Нормативно-правовые основания разработки программы

Нормативную правовую основу разработки программы составляют:

–Федеральный закон от 29.12.2012 г. №273 «Об образовании в Российской Федерации»;

– Приказ Министерства просвещения РФ от 27 июля 2022 г. N 629 “Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам”;

–Приказ Министерства науки и высшего образования РФ и Министерства просвещения РФ от 5 августа 2020 г. N 882/391"Об организации и осуществлении образовательной деятельности при сетевой форме реализации образовательных программ

–"Методические рекомендации по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) / Приложение к письму Министерства образования и науки Российской Федерации № 09-3242 от 18 ноября 2015 г.;

–Федеральный государственный образовательный стандарт среднего общего образования (утв. приказом Министерства образования и науки РФ от 17 мая 2012 г. N 413). (С изменениями и дополнениями от: 29 декабря 2014 г., 31 декабря 2015 г., 29 июня 2017 г., 24 сентября, 11 декабря 2020 г.);

–Федеральный государственный образовательный стандарт основного общего образования (утв. приказом Министерства образования и науки РФ от 17 декабря 2010 г. N 1897) (С изменениями и дополнениями от: 29 декабря 2014 г., 31 декабря 2015 г., 11 декабря 2020 г.);

–Постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020 г. № 28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

–Устав института;

–Положение о разработке и реализации дополнительных общеразвивающих программ.

1.2. Актуальность программы состоит в том, что активизация познавательного процесса позволяет учащимся более полно выразить свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

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

Дополнительный блок «Основы проектной деятельности» познакомит обучающихся с основными принципами создания и реализации проекта, правильной постановки цели и задач проекта.

1.3. Направленность программы – техническая

1.4. Адресат программы:

К освоению программы допускаются обучающиеся по общеобразовательным программам в возрасте от 14 до 17 лет.

1.5. Цель, задачи и планируемые результаты освоения программы:

Цель: способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python.

Основные задачи:

- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;

- развить умение создания, подготовки и реализации тематического проекта;

- научить составлять и читать блок-схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
- объяснить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Планируемые результаты освоения:

В результате изучения курса должны быть достигнуты определенные результаты.

Предметные:

- умеют составлять алгоритмы для решения задач;
- умеют реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- владеют основными навыками программирования на языке Python;
- умеют отлаживать и тестировать программы, написанные на языке Python;
- научить отлаживать и тестировать программы, делать выводы о работе этих программ.

Личностные:

- воспитать уважительное отношение к преподавателям и сверстникам, культуру поведения во время занятий и совместной продуктивной деятельности;
- сформировать культуру занятий, направленную на воспитание личностных и социальных качеств;
- сформированы способности к саморазвитию и самообразованию средствами информационных технологий на основе иллюстрированной среды программирования, мотивации к обучению и познанию;
- сформированы умения работать индивидуально и в группе для решения поставленной задачи;
- сформировано целостное мировоззрение, соответствующее современному уровню развития информационных технологий;
- сформировано осознанное позитивное отношение к другому человеку, его мнению, результату его деятельности;
- обеспечено усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой;
- сформирована культура начального программирования.

Метапредметные:

- развито умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- развито умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата;
- развито умение критически оценивать правильность решения учебно-исследовательской задачи;
- сформировано владение основами самоконтроля, способность к принятию решений;
- сформирована мотивация к профессиональному самоопределению учащихся.

1.6. Объем и срок освоения программы: объем программы 90 часов, срок освоения программы – 34 недели

1.7. Форма обучения: заочная с применением ДОТ.

1.8. Формы аттестации. Оценочные материалы.

- текущий контроль (тестирование) входит в количество часов по изучению разделов;

- итоговая аттестация (творческий проект).

1.9. Режим занятий – 90 часов, при заочной форме обучения 1 раз в неделю, 2

часа в день (перерыв между занятиями 10 минут).

1.10. Особенности организации образовательной деятельности. Содержание обучения, представленное в программе «Программирование на языке «Python», позволяет вести обучение в режиме актуального познания. Практическая направленность курса на создание внешних образовательных продуктов – блок-схем, алгоритмов, программ – способствует выявлению фактов, которые невозможно объяснить на основе имеющихся у учащихся знаний. –образовательная деятельность организована в традиционной форме: лекционно-практические занятия (возможно применение ДОТ).

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

1.11. Форма итоговой аттестации – защита проекта

2. УЧЕБНЫЙ ПЛАН

2.1.1. Учебный план по очной форме обучения

№	Наименование разделов, дисциплин (модулей)	всего часов	Аудиторная нагрузка			Промежуточная аттестация
			теоретичес- кие занятия	практич. занятия	СРС	форма
I	Раздел 1. Основы проектной деятельности	14	6	6	2	
1.1.	Общие понятия проектной деятельности	2	2			
1.2.	Методологический аппарат проекта	10	4	4	2	
1.3.	Оформление проекта	2		2		Представление структуры проекта
II	Раздел 2. Знакомство с языком Python. Переменные и выражения	12	6	4	2	Тестирование, решение практических задач
2.1.	Общие сведения о языке. Режимы работы. Знакомство с языком Python.	4	2		2	
2.2.	Переменные. Работа со справочной системой. Ввод и вывод.	4	2	2		
2.3.	Задачи на элементарные действия с числами. Переменные и выражения.	4	2	2		
III	Раздел 3. Условные предложения. Циклы	20	8	8	4	Тестирование, решение практических задач
3.1.	Логические выражения и операторы. Условный оператор. Множественное ветвление.	4	2	2		
3.2.	Реализация ветвления в языке Python. Составление	6	2	2	2	

	программ с ветвлением. Условные предложения.					
3.3.	Оператор цикла с условием. Числа Фибоначчи. Оператор цикла for. Решение задачи с циклом for.	4	2	2		
3.4.	Вложенные циклы. Реализация циклических алгоритмов. Составление программ с циклом. Циклы.	6	2	2	2	
IV	Раздел 4. Функции. Строки - последовательность символов	12	6	2	4	Тестирование, решение практических задач
4.1.	Создание функций. Локальные и глобальные переменные.	4	2		2	
4.2.	Рекурсивные функции. Функции.	4	2	2		
4.3.	Строки и срезы. Строки – последовательность символов.	4	2		2	
V	Раздел 5. Сложные типы данных. Стиль программирования и отладка программ	18	8	6	4	Тестирование, решение практических задач
5.1.	Списки и срезы списков. Матрицы.	4	2	2		
5.2.	Кортежи. Введение в словари.	4	2	2		
5.3.	Множества в языке Python. Сложные типы данных.	4	2		2	
5.4.	Стиль программирования. Отладка программ.	4	2	2		
5.5.	Что дальше? Перспективы использования Python.	2			2	
VI	Раздел 6. Рефлексивная деятельность по проекту	12	4	6	2	
6.1.	Требования к презентации проекта	6	2	2	2	
6.2.	Разработка сметы расходов на реализацию проекта	6	2	4		Презентация сметы расходов на проект
Итоговая аттестация		2		2		Презентация проекта
Итого:		90	38	34	18	

¹ По темам разделов II, III на практических занятиях допускается деление на подгруппы (не менее 3 человек в подгруппе)

2.1. Для реализации дополнительной общеразвивающей программы предусмотрена заочная форма обучения с применением ДОТ.

2.2. Срок освоения ДОП

2.2.1. Срок освоения ДОП при очной форме обучения составляет 34 недели, в том числе:

Обучение по разделам (дисциплинам)	33 нед
Итоговая аттестация	1 нед
Итого	34 нед.

3. СОДЕРЖАНИЕ ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

Наименование, содержание раздела, дисциплины (практические, теоретические занятия)	Всего часов
Раздел 1. Основы проектной деятельности	14
1.1. Общие понятия проектной деятельности.	2
Теоретическое занятие 1.1. Понятие «Проект», типы проектов (информационный, исследовательский, прикладной, игровой (ролевой), творческий, социальный, инженерный), этапы работы над проектом, структура проекта	2
1.2. Методологический аппарат проекта	10
Теоретическое занятие 1.2.1. Как сформулировать проблематику проекта. Понятие «проблема», требования к формулировке проблемы. Формулировка проблемы проекта с использованием метода SWOT-анализа. Целевая аудитория проекта.	2
Практическое задание 1.2.2. Формулировка проблемы проекта с использованием метода SWOT-анализа. Разработка портрета целевой аудитории проекта.	2
Теоретическое занятие 1.2.3. Формулировка идеи проекта. Понятие «идея проекта», требования к формулировке идеи проекта. Требования к формулировке актуальности проекта. Формулировка актуальности проекта. Требования к формулировке цели проекта. SMART технология постановки цели проекта. Требования к формулировке задач проекта.	2
Практическое занятие 1.2.4. Формулировка идеи, актуальности, цели и задач проекта	2
Самостоятельная работа 1.2.5. Определение сроков реализации проектов. Формулировка «контрольных точек» реализации проекта	2
Тема 1.3. Оформление проекта	2
Практические занятия 1.3. Требования к оформлению проекта	2
Раздел 2. Знакомство с языком Python. Переменные и выражения	12
Тема 2.1. Общие сведения о языке. Установка программы Python	2
Теоретическое занятие 2.1. Общие сведения о языке. Режимы работы. Установка программы Python	2
Тема 2.2. Переменные. Работа со справочной системой. Ввод и вывод	4
Теоретическое занятие 2.2. Типы данных. Преобразование типов. Переменные. Оператор присваивания. Имена переменных и ключевые слова. Ввод данных с клавиатуры. Вывод данных на экран.	2
Практическое занятие 2.2.1. Работа со справочной системой. Запрос ввода. Вывод структурированных данных. Создание переменных. Присваивание значений переменной. Логическое название переменных.	2
Тема 2.3. Задачи на элементарные действия с числами. Переменные и выражения.	4
Теоретическое занятие 2.3. Задачи на элементарные действия с числами.	1
Практическое занятие 2.3.1. Задачи на элементарные действия с числами. Тестирование по пройденным темам.	1
Раздел 3. Условные предложения. Циклы	20
Тема 3.1. Логические выражения и операторы. Условный оператор. Множественное ветвление	4

Теоретическое занятие 3.1. Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Множественное ветвление.	2
Практическое занятие 3.1.1. Логические выражения. Условный оператор. Множественное ветвление.	2
Тема 3.2. Реализация ветвления в языке Python. Составление программ с ветвлением	6
Теоретическое занятие 3.2. Реализация ветвления в языке Python. Алгоритм составления программ с ветвлением.	2
Практическое занятие 3.2.1 Реализация ветвления в языке Python. Составление программ с ветвлением.	2
Самостоятельная работа 3.2.2 Реализация ветвления в языке Python. Составление программ с ветвлением. Тестирование по пройденным темам.	2
Тема 3.3. Оператор цикла с условием. Числа Фибоначчи. Оператор цикла for. Решение задачи с циклом for.	4
Теоретическое занятие 4.1. Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for.	2
Практическое занятие 4.1. Числа Фибоначчи. Решение задачи с циклом for.	2
Тема 3.4. Вложенные циклы. Реализация циклических алгоритмов. Составление программ с циклом	6
Теоретическое занятие 3.4. Вложенные циклы. Алгоритм составления программ с циклом.	2
Практическое занятие 3.4.1. Реализация циклических алгоритмов. Составление программ с циклом.	2
Самостоятельная работа 3.4.2. Реализация циклических алгоритмов. Решение задачи циклических алгоритмов. Составление программ с циклом.	2
Раздел 4. Функции. Строки - последовательности символов	12
Тема 4.1. Создание функций. Локальные и глобальные переменные.	4
Теоретическое занятие 4.1. Создание функций. Параметры и аргументы. Локальные переменные. Глобальные переменные.	2
Самостоятельная работа 4.1.1. Создание функций с параметром. Использование локальных и глобальных переменных.	2
Тема 4.2. Рекурсивные функции	4
Теоретическое занятие 4.2. Рекурсивные функции. Вычисление факториала. Числа Фибоначчи.	2
Практическое занятие 4.2.1. Использование рекурсивных функций. Решение задачи с рекурсивными функциями. Тестирование по пройденным темам.	2
Тема 4.3. Строки и срезы. Строки - последовательности символов	4
Теоретическое занятие 6.1. Составной тип данных - строка. Доступ по индексу. Длина строки и отрицательные индексы. Преобразование типов. Применение цикла для обхода строки. Строки нельзя изменить. Сравнение строк. Оператор in. Модуль string. Операторы для всех типов последовательностей (строки, списки, кортежи).	2
Самостоятельная работа 6.1. Использование срезов и строк. Тестирование по пройденным темам.	2
Раздел 5. Сложные типы данных. Сложные типы данных	18
Тема 5.1. Списки и срезы списков. Матрицы	4
Теоретическое занятие 5.1. Списки. Тип список (list). Индексы. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Использование срезов. Удаление и клонирование списков. Списочные параметры. Функция range().	2
Практическое занятие 5.1.1 Вложенные списки. Строки и списки Решение задач со списками. Матрицы. Генераторы списков в Python.	2

Тема 5.2. Кортежи. Введение в словари	4
Теоретическое занятие 5.2. Кортежи. Присваивание кортежей. Введение в словари. Тип словарь (dict). Словарные операции. Словарные методы	2
Практическое занятие 5.2.1 Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения	2
Тема 5.3. Множества в языке Python. Сложные типы данных	4
Теоретическое занятие 5.3. Множества в языке Python. Множества. Множественный тип данных. Описание множеств. Операции, допустимые над множествами: объединение, пересечение, разность, включение.	2
Самостоятельная работа 5.3.1 Тестирование по пройденным темам.	2
Тема 5.4. Стиль программирования	4
Теоретическое занятие 5.4. Стиль программирования. Отладка программ.	2
Практическое занятие 5.4.1. Стиль программирования. Отладка программ. Определение собственного стиля программирования.	2
Тема 5.5. Что дальше? Перспективы использования языка Python	2
Самостоятельная работа 5.5.1 Перспективы использования языка Python	2
Раздел 6. Рефлексивная деятельность по проекту	12
Теоретическое занятие 6.1. Требования к презентации проекта	2
Практическое занятие 6.1. Разработка презентации проекта	4
Теоретическое занятие 6.2. Требования к смете проекта	2
Практическое занятие 6.2. Разработка сметы проекта	4
Итоговая аттестация(защита проекта)	2
Итого	90

4. ОРГАНИЗАЦИОННО-ПЕДАГОГИЧЕСКИЕ УСЛОВИЯ

4.1. Материально-техническое обеспечение

Технические средства обучения:

Реализация программы по адресу: Рябикова, 63;

- мебель, оборудование и расходные материалы (Приложение 1)

4.2. Информационное обеспечение обучения

Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы:

Основные источники:

1. Марк Лутц. Изучаем Python – Том 1, 5-е издание, 2019.
2. Марк Лутц. Изучаем Python – Том 2, 5-е издание, 2020.
3. Бейдер Дэн. Чистый Python. Тонкости программирования для профи, 2024.
4. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
5. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
6. Python 3.12.3 документация [Электронный ресурс] – Режим доступа: <https://docs.python.org/3/>, свободный.

Дополнительные источники:

1. Сайт / справочные материалы [Электронный ресурс] – Режим доступа: <https://metanit.com/python/>, свободный.
2. Сайт / интерактивный сборник задач для практики программирования [Электронный ресурс] – Режим доступа: <http://pythontutor.ru/>, свободный.
3. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/431>, свободный.
4. Сайт / среда разработки для языка Python [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.
5. Сайт проекта Open Book Project. Практические примеры на Python Криса Мейерса [Электронный ресурс] – Режим доступа: openbookproject.net, свободный.

6. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/58852> , свободный.

6. Сайт / Адаптивный тренажер Python [Электронный ресурс] – Режим доступа: <https://stepik.org/course/68343> , свободный.

Интернет-источники:

1. Сайт <https://www.python.org> — Официальный сайт языка программирования Python.

2. Сайт <https://stepik.org/course/82541/> — Курс для повышения уровня программирования на Python.

4.3. **Кадровое обеспечение:** Реализация программы обеспечивается педагогическими кадрами, имеющими образование, соответствующее профилю/направленности программы.

4.4. Организация образовательного процесса

Каждый обучающийся имеет рабочее место с доступом к сети Интернет (при необходимости), к современным профессиональным базам данных, информационным справочным и поисковым системам, электронной библиотеке «Библиоклуб».

Программа обеспечивается учебно-методическим комплексом и материалами по всем дисциплинам, разделам (модулям).

Каждый обучающийся обеспечен не менее чем одним учебным печатным и/или электронным изданием по каждой дисциплине (модулю) (включая электронные базы периодических изданий).

Внеаудиторная работа обучающихся сопровождается методическим обеспечением и обоснованием времени, затрачиваемого на ее выполнение.

Образовательная деятельность обучающихся предусматривает следующие виды учебных занятий и учебных работ: лекции, практические занятия, выполнение проектной работы

5. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ДОПОЛНИТЕЛЬНОЙ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

5.1. Текущая оценка результатов освоения программы осуществляется преподавателем в форме тестирования.

5.2. Освоение ДОП заканчивается итоговой аттестацией обучающихся. Лица, успешно освоившие ДОП и прошедшие итоговую аттестацию, получают сертификат.

5.3. В соответствии с учебным планом итоговая аттестация по программе «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON» осуществляется в форме проекта.

5.4. Порядок проведения итоговой аттестации:

Общий уровень достижений учащихся оценивается зачет/незачет по следующей шкале:

28 - 15 баллов: «зачет»;

15 - 0 баллов: «незачет»

Критерии оценки проекта:

1. Планирование и раскрытие плана, развитие темы. Высший балл ставится, если ученик определяет и четко описывает цели своего проекта, дает последовательное и полное описание того, как он собирается достичь этих целей, причем реализация проекта полностью соответствует предложенному им плану.

2. Сбор информации. Высший балл ставится, если персональный проект содержит достаточное количество относящейся к делу информации и ссылок на различные источники.

3. Выбор и использование методов и приемов. Высший балл ставится, если проект полностью соответствует целям и задачам, определенным автором, причем выбранные и эффективно использованные средства приводят к созданию итогового продукта высокого качества.

4. Анализ информации. Высший балл по этому критерию ставится, если проект четко отражает глубину анализа и актуальность собственного видения идей учащимся, при этом содержит по-настоящему личностный подход к теме.

5. Организация письменной работы. Высший балл ставится, если структура проекта и письменной работы (отчета) отражает логику и последовательность работы, если

использованы адекватные способы представления материала (диаграммы, графики, сноски, макеты, модели и т. д.).

6. Анализ процесса и результата. Высший балл ставится, если учащийся последовательно и полно анализирует проект с точки зрения поставленных целей, демонстрирует понимание общих перспектив, относящихся к выбранному пути.

7. Личное участие. Считается в большей степени успешной такая работа, в которой наличествует собственный интерес автора, энтузиазм, активное взаимодействие с участниками и потенциальными потребителями конечного продукта и, наконец, если ребенок обнаружил собственное мнение в ходе выполнения проекта.

С критериями оценивания проектов учащиеся знакомятся заранее. Также они сами могут предложить какие-либо дополнения в содержание критериев или даже дополнительные критерии, которые, на их взгляд, необходимо включить в критериальную шкалу. Критерии оценивания являются своего рода инструкцией при работе над проектом. Кроме того, обучающиеся, будучи осведомленными о критериях оценивания их проектной деятельности, могут улучшить отдельные параметры, предлагаемые для оценивания, тем самым получить возможность достижения наивысшего результата.

Максимальный уровень достижений учащихся по критериям:

Критерии	Максимальный уровень достижений учащихся	
1	Планирование и раскрытие плана, развитие темы	4
2	Сбор информации	4
3	Выбор и использование методов и приемов	4
4	Анализ информации	4
5	Организация письменной работы	4
7	Анализ процесса и результата	4
7	Личное участие	4
ИТОГО		28

Перечень материально-технического обеспечения

Аппаратное обеспечение:

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

Программное обеспечение:

Операционная система Windows 7 Профессиональная или выше;
Интерпретатор Python версии 3.7 и выше;
IDE Visual Studio Code или JetBrains PyCharm;
Adobe Acrobat Reader или другой просмотрщик PDF файлов;
WinRAR;
Пакет офисных программ;
Adobe Photoshop или другой растровый графический редактор;
Любой браузер для интернет серфинга.

Необходимое оборудование:

Ноутбук HP 255 G8 R5-3500U - 13 шт
ВЕБ-камера Logitech "HD C615, черный (960-001056) 1 шт
Наушники Edifier G1SE - 12 шт
Интерактивная панель TeachTouch 6.0 75", 20 касаний
Кабель HDMI – 1 шт

Мебель:

Доска магнитно-маркерная 100*150 см Ecopomy
Доска магнитно-маркерная Brauberg флипчарт 700*1000мм
Стул ученический (полумягкие цветные) 12 шт
Стул учительский (кресла полумягкие черные) 1 шт
Стол ученический (цветная коемочка) 12 шт
Стол учительский 1 шт

Входящий контроль

Низкий уровень: 0-70% выполненных заданий;

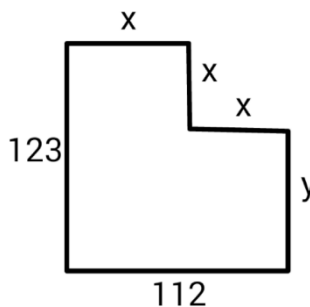
Средний уровень: 70-85% правильно выполненных заданий;

Высокий уровень: 85-100% правильно выполненных заданий.

Для обучающихся – 14-17 лет

Фамилия, Имя	
---------------------	--

1. Из прямоугольника вырезали квадрат со стороной x , получилась фигура как на рисунке. Чему равна сумма цифр y ?



2. В алфавите племени мумба-юмба 32 буквы. Любое слово в языке этого племени состоит из пяти букв и должно одинаково читаться справа налево и слева направо, при этом первые две буквы слова обязательно различаются, а третья совпадает с пятой.

Каково максимальное количество слов в этом языке?

3. Катя наклеила на рулет тонкие поперечные кольца трёх разных цветов. Если разрезать по серым кольцам, получится 25 кусков рулета, если по малиновым — 47 кусков, а если по зеленым — 31 кусок.

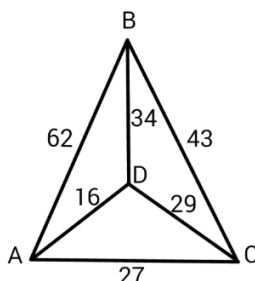
Сколько кусков рулета получится, если разрезать по кольцам всех трёх цветов?

Примечания

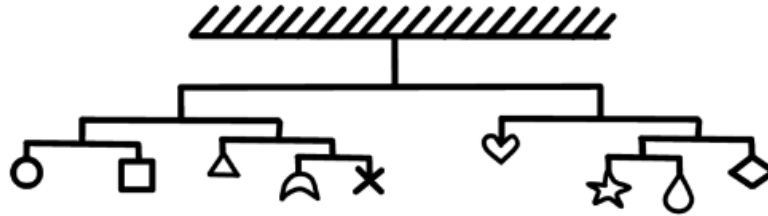
Учтите, что наклеить кольцо одного цвета на кольцо другого нельзя.

4. На рисунке показано расположение городов А, В, С и D и расстояния между ними. Турист выходит из города В и собирается посетить остальные города, побывав в каждом по разу.

Какова наименьшая возможная длина маршрута, если он хочет закончить свой путь в том же городе?



5. Фигурки, общей массой 432 грамма, при помощи невесомых нитей и планок собрали в конструкцию, изображённую на рисунке. Оказалось, что все её части находятся в равновесии. Сколько весит ромб?



6. Люди переезжают в города, за год численность людей удваивается. Если люди заселят весь город за 12 лет, то сколько лет понадобится, чтобы занять лишь половину города?

7. Сравните пары слов. Сколько среди них полностью идентичных?

O/Sanmarco

O/Samnarco

Ф.Wagonerrte

Ф.Wagonertre

A.S.Schmetterling

A.S.Schnetterling

N.V.Murfreesboroque

N.V.Munfreesboroque

P.S.Splendoursec

P.S.Sqlendoursec

Семь человек выясняли, какой сегодня день недели.

Первый сказал: «Послезавтра – воскресенье».

Второй: «Вчера был понедельник».

Третий: «Завтра будет суббота».

Четвертый: «Завтра будет среда».

Пятый: «Вчера был четверг».

Шестой: «Позавчера было воскресенье».

Седьмой: «Позавчера была среда».

Какой сегодня день недели, если трое ошибаются?

8. Вам предложены несколько высказываний и следствие из них (выделено жирным).

Согласны ли Вы с этим следствием?

1. Все клёны — растения.

2. Некоторые растения быстро желтеют.

Значит, некоторые клёны быстро желтеют.

Да

Нет

9. Гусеница прогрызает яблоко диаметром 6 сантиметров насквозь за 16 секунд, вылезая снаружи полностью.

Известно, что середину яблока она начинает грызть уже через 6 секунд после начала пути.

Какова длина гусеницы в сантиметрах?

Для учащихся 11-13 лет

Фамилия, Имя.	
---------------	--

1. Назовите два числа, у которых количество цифр равно количеству букв, составляющих название каждого из этих чисел.

--	--

2. Собака была привязана к десятиметровой веревке, а прошла двести метров. Как ей это удалось?

--

3. Есть дорога, по которой может проехать только одна машина. По дороге едут две машины: одна с горы, другая под гору. Как им разъехаться?

--

4. Как с помощью только одной палочки образовать на столе треугольник?

--

5. В каком месяце болтливая девочка говорит меньше всего?

--

6. Что становится на треть больше, если его поставить вверх ногами?

--

7. Представьте себе, что вы кондуктор. Поезд везет сто вагонов, в каждом вагоне 10 купе, в каждом купе 4 пассажира. Сколько лет кондуктору?

--

8. Юра разрезал огромную пиццу на 10 кусков. Затем он взял один из кусков и разрезал его еще на 10. После этого из имеющихся кусков он выбрал два и разрезал каждый из них на 10.

Вопрос: Сколько кусков пиццы получилось у Юры.

--

9. В компьютерной игре нужно победить монстра. Изначально у Юры было только 9 выстрелов. Но за каждое попадание он получал дополнительно еще 3 выстрела.

Вопрос: Сколько раз Юра попал в цель, если всего он выстрелил 30 раз, израсходовав все выстрелы?

--

Итоговый контроль
Диагностическая карта

№/ п	Имя	Входящая Диагностика	Промежуточ- ная диагностика	Итоговая диагностика
1				
2				
3+				
	Итого в %	Н С В		Н С В

Результативность отслеживается с помощью карты наблюдений, анализа участия детей в совместной продуктивной деятельности, разработки и защите творческих проектов. Усвоение программы возможно по 3-м уровням: низкий (Н), средний (С), высокий (В).

Низкий уровень

Учащиеся должны знать / понимать:

- технику безопасного поведения во время занятий;
- правила поведения в общественных местах,
- понятие программы Python; общую структуру программы;
- основные типы данных;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- основные циклы с условием;
- основные правила записи циклов условием;
- формат записи цикла с параметром;
- понятие функции;
- основные принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- операции со строками;
- способ описания списка;
- способ описания кортежа;
- способ описания словаря;
- основные операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ.

Учащиеся должны уметь:

- понимать учебную задачу, сохранять ее содержание в процессе ее выполнения под руководством педагога;
- работать в паре, малой группе;

- выполнить установку программы под руководством педагога;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе под руководством педагога;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием под руководством педагога;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи по наводящим вопросам педагога;
- создавать и использовать основные функции;
- описывать и соединять строки;
- находить подстроку в строке с помощью педагога;
- находить количество слов в строке;
- вводить и выводить элементы списка под руководством педагога;
- приводить примеры использования вложенных списков (матриц) по наводящим вопросам педагога;
- описывать множества под руководством педагога;
- определять принадлежность элемента множеству по наводящим вопросам;
- определять вид ошибок и находить ошибки в программе под руководством педагога;
- составлять элементарные алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python под руководством педагога.

Средний уровень

Учащиеся должны знать / понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности;
- основные приемы взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- основные способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;

- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- основные шаги работы над проектом, его презентации.

Учащиеся должны уметь:

- уважительно относиться к преподавателям и сверстникам;
- применять некоторые приемы логического (абстрактное) мышления;
- концентрировать внимание на одном или двух объектах;
- понимать причины успеха/неуспеха с помощью анализа педагога;
- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;
- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать и сохранять цели и задачи учебной деятельности в ходе занятия;
- планировать свою деятельность с помощью взрослого;
- сотрудничать с взрослыми и сверстниками в процессе выполнения учебной задачи;
- понимать причины успеха/неуспеха учебной деятельности под руководством взрослого;
- делать выводы в ходе полученных заданий; выстраивать логические цепи рассуждений под руководством педагога;

- выражать творческие идеи, разработать творческий проект на основе образца;
- конструктивно взаимодействовать в составе группы в ходе работы над проектом.

Высокий уровень

Учащиеся должны знать / понимать:

- основные положения техники безопасности на занятиях, правила поведения в общественных местах, правила дорожной безопасности, правила поведения во время чрезвычайных происшествий;
- приемы конструктивного взаимодействия в группе сверстников;
- понятие программы;
- структуру программы на Python;
- режимы работы с Python.
- общую структуру программы;
- типы данных;
- целые, вещественные типы данных и операции над ними;
- оператор присваивания;
- назначение условного оператора;
- способ записи условного оператора;
- логический тип данных;
- логические операторы or, and, not;
- циклы с условием и их виды;
- правила записи циклов условием;
- назначение и особенности использования цикла с параметром;
- формат записи цикла с параметром;
- примеры использования циклов различных типов.
- понятие функции;
- способы описания функции;
- принципы структурного программирования;
- понятие локальных переменных подпрограмм;
- понятие формальных и фактических параметров подпрограмм;
- способ передачи параметров.
- назначение строкового типа данных;
- операторы для работы со строками;
- процедуры и функции для работы со строками;
- операции со строками;
- сложные типы данных;
- способ описания списка;
- способ доступа к элементам списка;
- способ описания кортежа;
- способ описания словаря;
- операции, выполняемые со списками, кортежами и словарями;
- понятие множества;
- способы описания множества;
- операторы работы с множествами.
- что такое стиль программирования;
- правила именования объектов;
- основные рекомендации при написании программ;
- правила и этапы работы над проектом;
- приемы успешной презентации проекта.

Учащиеся должны уметь:

- выполнить установку программы;
- выполнить простейшую программу в интерактивной среде;
- написать комментарии в программе;
- решать задачи на элементарные действия с числами;
- использовать условный оператор;
- создавать сложные условия с помощью логических операторов;

- определять вид цикла, наиболее удобный для решения поставленной задачи;
- использовать цикл с условием;
- определять целесообразность применения и использования цикла с параметром для решения поставленной задачи;
- создавать и использовать функции;
- использовать механизм параметров для передачи значений;
- описывать строки;
- соединять строки;
- находить длину строки;
- вырезать часть строки;
- находить подстроку в строке;
- находить количество слов в строке;
- описывать списки;
- вводить элементы списка;
- выводить элементы списка;
- выполнять поиск элемента в списке, поиск минимума и максимума, нахождение суммы элементов списка;
- использовать вложенные списки;
- приводить примеры использования вложенных списков (матриц);
- описывать множества;
- определять принадлежность элемента множеству;
- вводить элементы множества;
- выводить элементы множества.
- определять вид ошибок и находить ошибки в программе.
- составлять алгоритмы для решения задач;
- реализовывать алгоритмы на компьютере в виде программ, написанных на языке Python;
- отлаживать и тестировать программы, написанные на языке Python;
- понимать учебную задачу, анализировать достижение результата;
- делать выводы в ходе полученных заданий; самостоятельно выстраивать логические цепи рассуждений;
- понимать причины успеха/неуспеха, конструктивно принимать причины неуспеха, исправлять ошибки с минимальным участием педагога;
- планировать свою деятельность, находить оригинальные способы выполнения поставленной творческой задачи;
- создать проектировочную команду и организовать ее деятельность;
- разработать творческий проект по предложенной или самостоятельно выбранной теме в составе творческой группы;
- самостоятельно подготовить презентацию и защитить проект.

Вопросы для промежуточного контроля по усвоению материала
(Промежуточный контроль проводится в последнюю неделю декабря)

Составьте выражение для вычисления в интерпретаторе Python 3 и вставьте в поле ответа результат вычисления:

1111·1111111

— **произведение чисел 1111 (5 единиц) и 111111 (7 единиц)**

Запишите число **1.2345e3** в виде **десятичной дроби**.

Составьте и запишите выражение для вычисления:

2014.0¹⁴(Возвестив 14 степень)

Обратите внимание на запись числа: это вещественное число.

Приведите к целому типу число 2.99

Расставьте скобки в выражении

a and b or not a and not b

в соответствии с порядком вычисления выражения (приоритетом операций). Всего потребуется 5 пар скобок (внешние скобки входят в их число).

Найдите результат выражения для заданных значений a и b

Учитывайте регистр символов при ответе.

a = True

b = False

a and b or not a and not b

Отметьте выражения, значения которых равны True:

- "239" < "30" and 239 < 30
- "239" < "30" and 239 > 30
- "239" > "30" and 239 < 30
- "239" > "30" and 239 > 30

Укажите результат выражения:

"123" + "42"

Какое значение будет у переменной i после выполнения фрагмента программы?

i = 0

while i <= 10:

i = i + 1

if i > 7:

i = i + 2

Сколько итераций цикла будет выполнено в этом фрагменте программы?

i = 0

while i <= 10:

i = i + 1

if i > 7:

i = i + 2

Сколько всего знаков * будет выведено после исполнения фрагмента программы:

i = 0

while i < 5:

print('*')

if i % 2 == 0:

print('')**

if i > 2:

print('*')**

i = i + 1

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

i = 0

s = 0

while i < 10:

i = i + 1

s = s + i

if s > 15:

break

i = i + 1

Определите, какое значение будет иметь переменная i после выполнения следующего фрагмента программы:

i = 0

s = 0

while i < 10:

i = i + 1

s = s + i

if s > 15:

continue

i = i + 1

Задачи для промежуточного контроля по усвоению материала

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, *, mod, pow, div, где
mod — это взятие остатка от деления,
pow — возведение в степень,
div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!".

Обратите внимание, что на вход программе приходят вещественные числа.

Жители страны Малевии часто экспериментируют с планировкой комнат. Комнаты бывают треугольные, прямоугольные и круглые. Чтобы быстро вычислять жилплощадь, требуется написать программу, на вход которой подаётся тип фигуры комнаты и соответствующие параметры, которая бы выводила площадь получившейся комнаты. Для числа π в стране Малевии используют значение 3.14.

Формат ввода, который используют Малевийцы:

Треугольник

a

b

c

где a, b и c — длины сторон треугольника

прямоугольник

a

b

где a и b — длины сторон прямоугольника

круг

r

где r — радиус окружности

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.

Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.

На вход программе подаётся строка из шести цифр. (Пример:123321)

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и **сразу после этого** выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, **после этого считывание продолжать не нужно**.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если $n = 7$, то программа должна вывести 1 2 2 3 3 3 4.

Выведите таблицу размером $n \times n$, заполненную числами от 1 до n^2 по спирали, выходящей из левого верхнего угла и закрученной по часовой стрелке, как показано в примере (здесь $n=5$)