

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ НИЖЕГОРОДСКОЙ ОБЛАСТИ
Государственное бюджетное профессиональное образовательное учреждение
«АРЗАМАССКИЙ ТЕХНИКУМ СТРОИТЕЛЬСТВА И ПРЕДПРИНИМАТЕЛЬСТВА»

Принята
на заседании педагогического совета
протокол № 9 от 30.08.2023 г.

УТВЕРЖДЕНО
приказом ГБПОУ АТСП
от 06.09.2023 г. № 600-а

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
«Программирование на языке Python»
Направленность – техническая

Возраст обучающихся: 15-18 лет

Срок реализации: 1 год

Объем: 144 часов

Авторы-составители:

Прокопчик Дарья Николаевна,
педагог дополнительного образования

Дятлов Антон Сергеевич,
педагог дополнительного образования

Лазарева Анастасия Михайловна,
методист

Арзамас
2023

Содержание

| | |
|---|----|
| 1. Комплекс основных характеристик программы | 3 |
| 1.1 Пояснительная записка | 3 |
| 1.2 Цель и задачи | 9 |
| 1.3 Содержание программы | 10 |
| 1.3.1 Учебный план | 10 |
| 1.3.2 Содержание учебного плана | 22 |
| 1.4 Планируемые результаты | 30 |
| 2. Комплекс организационно-педагогических условий реализации программы..... | 31 |
| 2.1 Календарный учебный график..... | 31 |
| 2.2 Условия реализации программы | 32 |
| 2.3 Формы аттестации и оценочные материалы | 34 |
| 2.4 Методические материалы..... | 36 |
| Список литературы | 39 |
| Приложение | 40 |

1. Комплекс основных характеристик программы

1.1 Пояснительная записка

Программирование занимает одну из значительных ниш в современном мире. Нынешнее программирование многогранно и используется в таких важных сферах как строительство, бизнес и экономика, медицина, биология и физика. Большой процент физического труда в промышленности заменен на машинный и роботизированный труд, который управляется посредством программного обеспечения, что обеспечивает существенный прирост скорости, точности операций и эффективности производства. Такое богатство разнообразия применений обеспечивается солидным выбором языков программирования, у каждого из которых есть свои плюсы и минусы.

Для реализации программы выбран язык Python – это один из самых популярных, быстро развивающихся языков в современном мире, при этом он имеет достаточно простой синтаксис и подходит для начального обучения.

Дополнительная общеобразовательная программа «Программирование на языке Python» имеет *техническую направленность*, в ее основу заложены принципы модульности и практической направленности, что обеспечит вариативность обучения. Содержание учебных модулей направлено на детальное изучение алгоритмизации, реализацию межпредметных связей, организацию проектной и исследовательской деятельности обучающихся.

Основанием для проектирования и реализации данной общеразвивающей программы служит *перечень следующих нормативных правовых актов и государственных программных документов:*

– Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 30.12.2021) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.01.2022);

– Приказ Министерства просвещения РФ от 9 ноября 2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

– Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»).

– СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» - Постановление Главного государственного санитарного врача от 28.09.2020 №28;

– СП 1.2.3685-21 «Гигиенические нормативы и требования к обеспечению безопасности и (или) безвредности для человека факторов среды обитания» - Постановление Главного государственного санитарного врача от 28.01.2021 №2.

– Приказ Минобрнауки России № 845, Минпросвещения России № 369 от 30.07.2020 «Об утверждении Порядка зачета организацией, осуществляющей образовательную деятельность, результатов освоения обучающимися учебных предметов, курсов, дисциплин (модулей), практики, дополнительных образовательных программ в других организациях, осуществляющих образовательную деятельность»;

– Методические рекомендации для субъектов Российской Федерации по вопросам реализации основных и дополнительных общеобразовательных программ в сетевой форме № МР-81/02вн от 28.06.2019.

Актуальность программы обусловлена современным этапом развития общества, характеризующимся ускоренными темпами освоения техники и технологий, потребностью общества в технически грамотных специалистах и

полностью отвечает социальному заказу по подготовке квалифицированных кадров в области IT-технологий. Непрерывно требуются новые идеи для создания конкурентоспособной продукции, подготовки высококвалифицированных кадров. Творческие способности и профессиональное мастерство специалистов становится главной производительной силой общества, и, в целях приумножения достижений во всех областях науки и техники, необходимо планомерное и заблаговременное развитие у молодежи творческих и технических способностей, а также повышение статуса информационного образования в обществе.

Прогностичность программы «Программирование на языке Python» заключается в том, что она отражает требования и актуальные тенденции не только сегодняшнего, но и завтрашнего дня, а также имеет междисциплинарный характер, что полностью отражает современные тенденции построения как дополнительных общеобразовательных общеразвивающих программ, так и образования в целом.

Отличительная особенность программы дополнительной общеобразовательной общеразвивающей программы «Программирование на языке Python» включает в себя ознакомление с теоретическими аспектами и синтаксисом языка, а также обучаются практическим навыкам программирования в среде Python.

Процесс обучения по данной программе построена таким образом, что полученные теоретические знания сразу же применяются и закрепляются на практике. Подобный подход способствует лучшему усвоению материала.

Адресат программы: дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» предназначена для обучающихся в возрасте 15-18 лет, не имеющих ограниченных возможностей здоровья, проявляющих интерес к программированию.

Формы занятий групповые, количество обучающихся в группе – 8-12 человек.

Группа формируется **по возрасту** 15-18 лет.

Место проведения занятий: 607220, Нижегородская обл., г. Арзамас, ул. Мира, д. 1.

Возрастные особенности группы. Содержание программы учитывает возрастные и психологические особенности обучающихся в возрасте 15-18 лет, указанные в ДООП и определяющие выбор форм проведения занятий с ними. Выделенные нами возрастные периоды при формировании групп 15-18 лет базируются на психологических особенностях развития старшего подросткового возраста.

Содержание программы учитывает возрастные и психологические особенности обучающихся в возрасте 15-18 лет, которые определяют выбор форм проведения занятий с ними. Обучающиеся этого возраста отличаются открытием своего внутреннего мира, внутреннего «Я». Главным измерением времени в самосознании является будущее, к которому они себя готовят. Ведущая деятельность в этом возрасте – учебно-профессиональная, в процессе которой формируются такие новообразования, как мировоззрение, профессиональные интересы, самосознание, мечта и идеалы, поэтому основной формой проведения занятий выбраны практические занятия.

Также следует отметить, что обучающиеся данной возрастной группы характеризуются такими психическими процессами, как стремление углубленно понять себя, разобраться в своих чувствах, настроениях, мнениях, отношениях. Это порождает у обучающихся стремление к самоутверждению, самовыражению (проявления себя в тех качествах, которые они считают наиболее ценными) и самовоспитанию. Эти процессы позволяют положить начало созданию начального профессионального самоопределения.

Режим занятий, объем программы: длительность одного занятия составляет 2 академических часа (по 45 минут) с перерывом (переменой) в 10 минут, периодичность занятий – 2 раза в неделю.

Срок освоения программы определяется содержанием программы и составляет 1 год (144 часа в год).

Формы обучения: очная с применением дистанционных образовательных технологий (Закон №273-ФЗ, гл.2, ст.17, п.2.).

Виды занятий: теоретические занятия, практические занятия, метод проектов. Также программа курса включает групповые и индивидуальные формы работы обучающихся (в зависимости от темы занятия).

По типу организации взаимодействия педагога с обучающимися при реализации программы используются личностно-ориентированные технологии, технологии сотрудничества.

Реализация программы предполагает использование здоровьесберегающих технологий.

Здоровьесберегающая деятельность реализуется:

- через создание безопасных материально-технических условий;
- включением в занятие динамических пауз, периодической смены деятельности обучающихся;
- контролем педагога за соблюдением обучающимися правил работы за персональным компьютером;
- через создание благоприятного психологического климата в учебной группе в целом.

Объем общеразвивающей программы: 144 часа. Форма организации образовательной деятельности – групповая.

Педагогическая целесообразность программы «Программирование на языке Python» заключается в том, что в современных условиях технологическое образование становится необходимостью, поскольку настоящий этап развития общества характеризуется интенсивным внедрением во все сферы человеческой деятельности новых наукоемких технологий. Поэтому раннее привлечение обучающихся к техническому творчеству в процессе изучения языков программирования является актуальным и полностью отвечает интересам обучающихся этой возрастной группы, их способностям и возможностям.

Аннотация

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» объемом 144 часа, рассчитана на обучающихся возрастной категории с 15 до 18 лет, состоит из 9 разделов, в результате освоения которых обучающийся овладеет навыком программирования и проектирования приложений с применением на языке программирования Python, навыком использования соответствующей технической документации, а также навыком работы в соответствующих средах разработки.

Настоящая программа предполагает использование языка программирования Python, интегрированной среды разработки программного обеспечения PyCharm.

В процессе создания программного обеспечения на языке программирования Python обучающиеся получают дополнительные знания в области Информационных технологий.

1.2 Цель и задачи

Цель: изучение основных синтаксиса и конструкций языка программирования Python и проектирование компьютерных программ с использованием различных модулей.

Для успешной реализации поставленной цели необходимо решить ряд педагогических, развивающих и воспитательных *задач*:

Обучающие:

- изучить конструкции языка программирования Python;
- изучить принципы и метода функционального и объектно-ориентированного программирования на языке Python;
- изучить основные структуры данных и типовые методы обработки этих структур;

Развивающие:

- развить навыки разработки алгоритмов и программ на основе изучения языка программирования Python;
- развить навыки алгоритмического и логического мышления;

Воспитательные:

- сформировать самостоятельность и творческий подход к решению задач;
- изучить этику групповой работы, отношений делового сотрудничества, взаимоуважения;
- способствовать воспитанию личностных качеств: целеустремленности, настойчивости, самостоятельности, чувства коллективизма и взаимной поддержки, чувство такта.

1.3 Содержание программы

1.3.1 Учебный план

Таблица 1

| №, п/п | Название раздела, темы | Описание тем | Кол-во часов | | |
|---|------------------------|---|--------------|----------|----------|
| | | | Всего | Теория | Практика |
| Раздел 1. Знакомство с основными понятиями языка программирования Python | | | 4 | 3 | 1 |
| 1-2 | Введение | Введение. Ознакомительное занятие. Правила поведения и техника безопасности в кабинете. Правила работы с компьютером. | 2 | 2 | - |
| 3-4 | Знакомство со средой | Введение в язык программирования Python. Изучение базовых понятий Среда программирования PyCharm. Написание первой программы. | 2 | 1 | 1 |
| Раздел 2. Основы программирования на языке Python | | | 14 | 7 | 7 |
| 5-6 | Условный оператор | Знакомство с условным оператором if: Синтаксис условной инструкции if, Синтаксис условной инструкции if – else, Синтаксис условной инструкции if – elif – else. | 2 | 1 | 1 |

| | | | | | |
|-------------------------------------|--------------------------------|--|-----------|-----------|-----------|
| 7-8 | Простые встроенные функции | Знакомство со встроенными функциями языка Python. | 2 | 1 | 1 |
| 9-10 | Знакомство с циклом while | Знакомство оператором цикла while: синтаксис цикла. | 2 | 1 | 1 |
| 11-12 | Отладчик | Знакомство с отладчиком кода, использование режима отладки. | 2 | 1 | 1 |
| 13-14 | Знакомство с циклом for | Знакомство оператором цикла for: понятие циклов, синтаксис цикла, функция range. | 2 | 1 | 1 |
| 15-16 | True и False, break и continue | Знакомство логическими переменными и операторами циклов: break, pass и continue. | 2 | 1 | 1 |
| 17-18 | Вложенные циклы | Знакомство с понятием вложенных циклов. | 2 | 1 | 1 |
| Раздел 3. Последовательности | | | 24 | 12 | 12 |

| | | | | | |
|-------|-----------------------------------|--|---|---|---|
| 19-20 | Множества | Знакомство со множествами: определение множества, задание множества. | 2 | 1 | 1 |
| 21-22 | Строки. Индексация | Знакомство с понятием индексации. Использование индексации в строках. | 2 | 1 | 1 |
| 23-24 | Строки. Срезы | Знакомство со срезами: синтаксис описания срезов, выборка элементов. | 2 | 1 | 1 |
| 25-26 | Пробная СР | Проведение пробной самостоятельной работы. | 2 | 1 | 1 |
| 27-28 | Знакомство со списками | Знакомство со списками: определение списка, задание списка, работа с элементами списков. | 2 | 1 | 1 |
| 29-30 | Кортежи. Преобразование коллекций | Знакомство с кортежами: определение кортежей, задание кортежей, работа с элементами кортежа. | 2 | 1 | 1 |

| | | | | | |
|--------------------------|---|---|-----------|----------|----------|
| 31-32 | Методы split и join. Списочные выражения | Знакомство с методами split и join. Понятие структуры и использование списочных выражений. | 2 | 1 | 1 |
| 33-34 | Методы списков и строк | Изучение методов списков и строк. | 2 | 1 | 1 |
| 35-36 | Самостоятельная работа | Выполнение самостоятельной работы. | 2 | - | 2 |
| 37-38 | Разбор СР | Разбор проведенной самостоятельной работы. | 2 | 2 | - |
| 39-40 | Вложенные списки | Знакомство с понятием вложенных списков и их структурой. | 2 | 1 | 1 |
| 41-42 | Знакомство со словарями | Знакомство со словарями: определение словарей, задание словарей, работа с элементами словаря. | 2 | 1 | 1 |
| Раздел 4. Функции | | | 14 | 5 | 9 |

| | | | | | |
|-------|--|---|---|---|---|
| 43-44 | Функции | Знакомство с интерфейсом программы PyCharm. Знакомство с функциями: определение, объявление, вызов функции. | 2 | 1 | 1 |
| 45-46 | Функции. Возвращение значений из функции | Возврат значений из функции. Знакомство с оператором return. | 2 | 1 | 1 |
| 47-48 | Области видимости переменных | Знакомство с локальными и глобальными переменными в Python. Области видимости переменных в Python. | 2 | 1 | 1 |
| 49-50 | Функции: передача параметров | Изучение механизма передачи параметров в функции. | 2 | 1 | 1 |
| 51-52 | Функции с переменным числом аргументов | Использование функций с переменным числом аргументов. | 2 | 1 | 1 |
| 53-54 | Решение задач | Отработка навыков решения задач по всему пройденному разделу. | 2 | - | 2 |

| | | | | | |
|--|---|--|-----------|----------|----------|
| 56-55 | Самостоятельная работа | Выполнение самостоятельной работы. | 2 | - | 2 |
| Раздел 5. Библиотеки языка Python | | | 12 | 6 | 6 |
| 56-57 | Библиотеки Python. Часть №1 (Модули стандартной библиотеки) | Знакомство со стандартной библиотеками языка Python (random, time и т.п.). | 2 | 1 | 1 |
| 58-59 | Библиотеки Python. Часть №2 (графика) | Знакомство с графической библиотекой языка Python. | 2 | 1 | 1 |
| 60-61 | Библиотеки Python. Часть №3 (графика + звук) | Знакомство с библиотекой для обработки звука и видео. | 2 | 1 | 1 |
| 63-62 | Библиотеки Python. Часть №4 (морфология) | Знакомство с библиотекой для обработки слов. | 2 | 1 | 1 |
| 64-65 | Библиотеки Python. Часть №5 (документы) | Знакомство с библиотекой для работы с документами. | 2 | 1 | 1 |

| | | | | | |
|-------------------------|---------------------------------|---|-----------|----------|----------|
| 66-67 | Библиотеки Python. Numpy | Знакомство с библиотекой Numpy для работы с многомерными массивами. | 2 | 1 | 1 |
| 68-69 | Промежуточная аттестация | | 2 | - | 2 |
| Раздел 6. PyGame | | | 10 | 5 | 5 |
| 70-71 | PyGame | Знакомство с библиотекой PyGame и создание первого приложения. | 2 | 1 | 1 |
| 72-73 | Обработка событий | Создание и применения обработчика событий в графическом окне PyGame | 2 | 1 | 1 |
| 74-75 | Изображения | Добавление изображений в графическое окно и взаимодействие с ними. | 2 | 1 | 1 |
| 76-77 | Самостоятельная работа | Выполнение самостоятельной работы. | 2 | - | 2 |

| | | | | | |
|--|------------------------------|---|-----------|----------|----------|
| 78-79 | Разбор СР | Разбор проведенной самостоятельной работы. | 2 | 2 | - |
| Раздел 7. Объектно-ориентированное программирование | | | 16 | 7 | 9 |
| 80-81 | Введение в ООП | Знакомство с понятием ООП, его основными принципами и сущностями объектно-ориентированного подхода к разработке ПО. | 2 | 1 | 1 |
| 82-83 | Введение в ООП. Полиморфизм | Использование механизмов полиморфизма. | 2 | 1 | 1 |
| 84-85 | ООП. Определение операторов. | Изучение механизма переопределения операторов. | 2 | 1 | 1 |
| 86-89 | ООП. Наследование | Использование механизмов наследования. | 4 | 2 | 2 |
| 90-92 | Решение задач | Отработка навыков решения задач по всему пройденному разделу. | 2 | - | 2 |

| | | | | | |
|------------------------|------------------------------------|--|-----------|----------|-----------|
| 93-94 | Самостоятельная работа | Выполнение самостоятельной работы. | 2 | - | 2 |
| 95-96 | Разбор задач из СР | Разбор проведенной самостоятельной работы. | 2 | 2 | - |
| Раздел 8. PyQt5 | | | 30 | 7 | 23 |
| 97-100 | Знакомство с пакетом PyQt5 | Знакомство с библиотекой PyQt5. Изучение основных команд и виджетов. | 4 | 2 | 2 |
| 101-104 | Знакомство с программой QTDesigner | Знакомство с программой QTDesigner, Разработка дизайна и использование его в своих программах. | 4 | 2 | 2 |
| 105-108 | Решение задач | Отработка навыков создания программ оконного интерфейса. | 4 | - | 4 |
| 109-110 | Знакомство с БД | Знакомство с понятием базы данных. Знакомство с программой SQLite и привязка БД к своей программе. | 2 | 1 | 1 |

| | | | | | |
|---|-----------------------------------|--|-----------|----------|-----------|
| 111-114 | Решение задач | Отработка навыков создания программ с оконным интерфейсом. | 4 | - | 4 |
| 115-116 | Сигналы и слоты | Знакомство с слотами и сигналами, использование их в приложениях. | 2 | 1 | 1 |
| 117-118 | Решение задач | Отработка навыков создания программ с оконным интерфейсом. | 2 | - | 2 |
| 119-120 | Создание exe-файла | Знакомство с модулем pyinstaller и создание exe-файла программного кода. | 2 | 1 | 1 |
| 121-124 | Решение задач | Отработка навыков создания программ по всему пройденному разделу. | 4 | - | 4 |
| Раздел 9. Проектная деятельность | | | 18 | 7 | 11 |
| 125-126 | Введение в проектную деятельность | Понятие проекта, проектной деятельности, проектной культуры. Проекты в современном мире. | 2 | - | 2 |

| | | | | | |
|---------|--|---|---|---|---|
| 127-128 | Этапы проекта | Этапы работы над проектом. Определение тематик проектов. Формирование проектных групп. Формирование целей проекта. Содержание портфолио проекта. | 2 | 1 | 1 |
| 129-130 | Работа над индивидуальным проектом (часть 1) | Выбор темы индивидуального или группового проекта. | 2 | 2 | - |
| 131-132 | Работа над индивидуальным проектом (часть 2) | Составление плана выполнения индивидуального или группового проекта. Распределение ролей. | 2 | 1 | 1 |
| 133-134 | Работа над индивидуальным проектом (часть 3) | Реализация индивидуального или группового проекта. | 2 | - | 2 |
| 135-136 | Работа над индивидуальным проектом (часть 4) | Анализ сделанных проектов. Правки и дополнения. | 2 | 1 | 1 |
| 137-138 | Оформление проекта | Оформление индивидуального проекта. Технические требования к оформлению проектных работ. Изучение требований к составлению презентаций для защиты индивидуального проекта | 2 | 1 | 1 |

| | | | | | |
|---------------|--|--|------------|-----------|-----------|
| 139-140 | Требования и подготовка публичного выступления | Подготовка к выступлению. Общие рекомендации. Техника подготовки. Психологический аспект готовности к выступлению. Культура выступления: соблюдение правил этикета, ответы на вопросы, заключительное слово. | 2 | 1 | 1 |
| 141-142 | Консультации | Подготовка к защите индивидуального проекта | 2 | - | 2 |
| 143-144 | Итоговая аттестация | | 2 | - | 2 |
| Итого: | | | 144 | 59 | 85 |

1.3.2 Содержание учебного плана

Раздел 1. Знакомство с основными понятиями языка программирования Python

1.1 Введение

Теория: Введение. Ознакомительное занятие. Правила поведения и техника безопасности в кабинете. Правила работы с компьютером.

1.2 Знакомство со средой

Теория: Введение в язык программирования Python. Изучение базовых понятий Среда программирования PyCharm. Написание первой программы.

Раздел 2. Основы программирования на языке Python

2.1 Условный оператор

Теория: Знакомство с условным оператором if: синтаксис конструкции.

Практика: Решение примеров.

2.2 Простые встроенные функции

Теория: Знакомство со встроенными функциями языка Python.

Практика: Решение примеров.

2.3 Знакомство с циклом while

Теория: Знакомство оператором цикла while: синтаксис цикла.

Практика: Решение примеров.

2.4 Отладчик

Теория: Знакомство с отладчиком кода, использование режима отладки.

Практика: Решение примеров.

2.5 Знакомство с циклом for

Теория: Знакомство оператором цикла for: понятие циклов, синтаксис цикла, функция range.

Практика: Решение примеров.

2.6 True и False, break и continue

Теория: Знакомство логическими переменными и операторами циклов: break, pass и continue.

Практика: Решение примеров.

2.7 Вложенные циклы.

Теория: Знакомство с понятием вложенных циклов.

Практика: Решение примеров.

Раздел 3. Последовательности

3.1 Множества

Теория: Знакомство со множествами: определение множества, задание множества.

Практика: Решение примеров.

3.2 Строки. Индексация

Теория: Знакомство с понятием индексации. Использование индексации в строках

Практика: Решение примеров.

3.3 Строки. Срезы

Теория: Знакомство со срезами: синтаксис описания срезов, выборка элементов.

Практика: Решение примеров.

3.4 Самостоятельная работа

Практика: Выполнение пробной самостоятельной работы.

3.5 Знакомство со списками

Теория: Знакомство со списками: определение списка, задание списка, работа с элементами списков.

Практика: Решение примеров.

3.6 Кортежи. Преобразование коллекций

Теория: Знакомство с кортежами: определение кортежей, задание кортежей, работа с элементами кортежа.

Практика: Решение примеров.

3.7 Методы split и join. Списочные выражения

Теория: Знакомство с методами split и join. Понятие структуры и использование списочных выражений.

Практика: Решение примеров.

3.8 Методы списков и строк

Теория: Изучение методов списков и строк.

Практика: Решение примеров.

3.9 Самостоятельная работа

Практика: Выполнение самостоятельной работы.

3.10 Разбор СР

Теория: Разбор проведенной самостоятельной работы.

3.11 Вложенные списки

Теория: Знакомство с понятием вложенных списков и их структурой.

Практика: Решение примеров.

3.12 Знакомство со словарями

Теория: Знакомство со словарями: определение словарей, задание словарей, работа с элементами словаря.

Раздел 4. Функции

4.1 Функции

Теория: Знакомство с интерфейсом программы PyCharm. Знакомство с функциями: определение, объявление, вызов функции.

Практика: Решение примеров.

4.2 Функции. Возвращение значений из функции

Теория: Возврат значений из функции. Знакомство с оператором return.

Практика: Решение примеров.

4.3 Области видимости переменных

Теория: Знакомство с локальными и глобальными переменными в Python. Области видимости переменных в Python.

Практика: Решение примеров.

4.4 Функции: передача параметров

Теория: Изучение механизма передачи параметров в функции.

Практика: Решение примеров.

4.5 Функции с переменным числом аргументов

Теория: Использование функций с переменным числом аргументов.

Практика: Решение примеров.

4.8 Решение задач

Практика: Решение примеров.

4.9 Самостоятельная работа

Практика: Выполнение самостоятельной работы.

Раздел 5. Библиотеки языка Python

5.1 Библиотеки Python. Часть №1 (Модули стандартной библиотеки)

Теория: Знакомство со стандартными библиотеками языка Python (random, time и т.п.).

Практика: Решение примеров.

5.2 Библиотеки Python. Часть №2 (графика)

Теория: Знакомство с графической библиотекой языка Python.

Практика: Решение примеров.

5.3 Библиотеки Python. Часть №3 (графика + звук)

Теория: Знакомство с библиотекой для обработки звука и видео.

Практика: Решение примеров.

5.4 Библиотеки Python. Часть №4 (морфология)

Теория: Знакомство с библиотекой для обработки слов.

Практика: Решение примеров.

5.5 Библиотеки Python. Часть №5 (документы)

Теория: Знакомство с библиотекой для работы с документами.

Практика: Решение примеров.

5.6 Библиотеки Python. Numpy

Теория: Знакомство с библиотекой Numpy для работы с многомерными массивами.

Практика: Решение примеров.

Промежуточная аттестация

Результатом освоения следующих разделов: «Введение», «Основы программирования на языке Python», «Последовательности», «Функции» и «Библиотеки языка Python» общеобразовательной общеразвивающей программы «Программирование на языке Python» является проведение промежуточной аттестации в форме решения задач.

Раздел 6. PyGame

6.1 PyGame

Теория: Знакомство с библиотекой PyGame.

Практика: Создание первого приложения.

6.2 Обработка событий

Теория: Создание и применения обработчика событий в графическом окне PyGame

Практика: Создание приложений.

6.3 Изображения

Теория: Добавление изображений в графическое окно и взаимодействие с ними.

Практика: Создание приложений

6.4 Самостоятельная работа

Практика: Выполнение самостоятельной работы.

6.5 Разбор СМ

Практика: Разбор проведенной самостоятельной работы.

Раздел 7. Объектно-ориентированное программирование

7.1 Знакомство с ООП

Теория: Знакомство с понятием ООП, его основными принципами и сущностями объектно-ориентированного подхода к разработке ПО.

Практика: Решение примеров.

7.2 Введение в ООП. Полиморфизм

Теория: Использование механизмов полиморфизма.

Практика: Решение примеров.

7.3 ООП. Определение операторов.

Теория: Изучение механизма переопределения операторов.

Практика: Решение примеров.

7.4 ООП. Наследование

Теория: Использование механизмов наследования.

Практика: Решение примеров.

7.5 Решение задач

Практика: Решение примеров.

7.6 Самостоятельная работа

Практика: Выполнение самостоятельной работы.

7.7 Разбор задач из СМ

Практика: Разбор проведенной самостоятельной работы.

Раздел 8. PyQt5

8.1 Знакомство с пакетом PyQt5

Теория: Знакомство с библиотекой PyQt5. Изучение основных команд и виджетов.

Практика: Решение примеров.

8.2 Знакомство с программой QTDesigner

Теория: Знакомство с программой QTDesigner, Разработка дизайна и использование его в своих программах.

Практика: Решение примеров.

8.3 Знакомство с БД

Теория: Знакомство с понятием базы данных. Знакомство с программой SQLite и привязка БД к своей программе.

Практика: Решение примеров.

8.4 Сигналы и слоты

Теория: Знакомство с слотами и сигналами, использование их в приложениях.

Практика: Решение примеров.

8.5 Создание exe-файла

Теория: Знакомство с модулем pyinstaller и создание exe-файла программного кода.

Практика: Решение примеров.

Раздел 9. Проектная деятельность

9.1 Введение в проектную деятельность

Теория: Понятие проекта, проектной деятельности, проектной культуры. Проекты в современном мире.

9.2 Этапы проекта

Теория: Этапы работы над проектом. Формирование целей проекта. Содержание портфолио проекта.

Практика: Определение тематик проектов. Формирование проектных групп.

9.3 Работа над индивидуальным проектом (часть 1)

Теория: анализ различных проектов с использованием ИИ. Выбор темы индивидуального или группового проекта.

9.4 Работа над индивидуальным проектом (часть 2)

Теория: Составление плана выполнения индивидуального или группового проекта. Распределение ролей.

Практика: Начало выполнения проекта

9.5 Работа над индивидуальным проектом (часть 3)

Практика: Реализация индивидуального или группового проекта.

9.6 Работа над индивидуальным проектом (часть 4)

Теория: Анализ сделанных проектов.

Практика: реализация правок и дополнений в проекты.

9.7 Оформление проекта

Теория: Технические требования к оформлению проектных работ. Изучение требований к составлению презентаций для защиты индивидуального проекта

Практика: Оформление индивидуального проекта.

9.8 Требования и подготовка публичного выступления

Теория: Подготовка к выступлению. Общие рекомендации. Техника подготовки. Психологический аспект готовности к выступлению. Культура выступления: соблюдение правил этикета, ответы на вопросы, заключительное слово.

Практика: Подготовка к выступлению.

9.9 Консультации

Практика: Подготовка к защите индивидуального проекта.

Итоговая аттестация

Итоговая аттестация реализуется в виде защиты индивидуально-группового проекта обучающимися на тему, выбранную ранее или предложенную педагогом.

1.4 Планируемые результаты

Предметные результаты:

- знание основных синтаксических конструкций языка программирования Python;
- формирование представления о структуре программного кода;
- умение применять на практике базовые алгоритмические конструкции;
- знание содержания этапов разработки программы: алгоритмизация-кодирование-отладка-тестирование;
- умение применять дополнительные возможности языка Python для выражения различных алгоритмических ситуаций;
- знание алгоритмов и программ на языке Python для решения простых, сложных и нестандартных задач.

Личностные результаты:

- формирование устойчивый интерес к техническому творчеству, мотивацию к изучению современных направлений в программировании;
- формирование коммуникативных навыков, умение работать в команде;
- развитие логического и творческого мышления;
- развитие внимания, аккуратности, терпения у обучающихся;
- формирование использование принципов здоровьесбережения;
- формирование уважительного отношения к своему и чужому труду, бережного отношения к используемому оборудованию.

Метапредметные результаты:

- формирование способности планировать последовательность шагов для достижения целей;
- развитие умения осуществлять самостоятельный поиск информации, анализировать и обобщать ее;
- формирование умение работать в паре и в коллективе;
- развитие умений формулировать, аргументировать и отстаивать свое мнение.

2. Комплекс организационно-педагогических условий реализации программы

2.1 Календарный учебный график

| Год обучения | сентябрь | | | | | октябрь | | | | | ноябрь | | | | | декабрь | | | | | январь | | | | | февраль | | | | | март | | | | | апрель | | | | | май | | | | | Всего учебных недель/ часов | Всего часов по программе |
|--------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|--------------|---------------|---------------|---------------|---------------|--|--------|-----|--|--|--|-----------------------------|--------------------------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | | | | | | | | |
| 1 год обу- | 2 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 2 | | 37/144 | 144 | | | | | |
| | | | | | | | | | | | | | | | | | | 2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 01.09 - 03.09 | 04.09 - 10.09 | 11.09 - 17.09 | 18.09 - 24.09 | 25.09 - 01.10 | 02.10 - 08.10 | 09.10 - 15.10 | 16.10 - 22.10 | 23.10 - 29.10 | 30.10 - 05.11 | 06.11 - 12.11 | 13.11 - 19.11 | 20.11 - 26.11 | 27.11 - 03.12 | 04.12 - 10.12 | 11.12 - 17.12 | 18.12 - 24.12 | 25.12 - 31.12 | 01.01 - 07.01 | 08.01 - 14.01 | 15.01 - 21.01 | 22.01 - 28.01 | 29.01 - 04.02 | 05.02 - 11.02 | 12.02 - 18.02 | 19.02 - 25.02 | 26.02 - 03.03 | 04.03 - 10.03 | 11.03 - 17.03 | 18.03 - 24.03 | 25.03 - 31.03 | 01.04 - 07.04 | 08.04 - 14.04 | 15.04 - 21.04 | 22.04, 28.04 | 29.04 - 05.05 | 06.05 - 12.05 | 13.05 - 19.05 | 20.05 - 26.05 | | | | | | | | |

| Условные обозначения: | |
|-----------------------|---|
| | Занятия по расписанию |
| | Каникулярный период |
| | Итоговая/промежуточная аттестация обучающихся |

2.2 Условия реализации программы

Материально-техническое обеспечение:

Требования к помещению:

- помещение для занятий, отвечающее требованиям СанПин для учреждений дополнительного образования;
- качественное освещение;
- столы, стулья по количеству обучающихся и 1 рабочее место для педагога.

Оборудование:

- компьютеры или ноутбуки на каждого обучающегося и преподавателя;
- манипулятор типа мышь;
- МФУ (принтер, сканер, копир);
- Wi-Fi;
- мультимедийный проектор либо интерактивная доска для показа презентаций.

- наушники 13 штук

Расходные материалы:

- маркеры;
- бумага писчая;
- permanent маркеры;
- шариковые ручки.

Информационное обеспечение:

- операционная система Windows;
- поддерживаемые браузеры: Yandex Browser, Chrome, Chrome Mobile, Firefox, Opera, Safari, Mobile Safari, Edge, Python;
- среда PyCharm Community Edition;
- пакет PyQt5;
- пакет библиотек со SciPy: numpy, scipy, matplotlib, ipython + ipythonnotebook, sympy, pandas;
- рекомендуется установить ПО Anaconda.

Методическое обеспечение:

Методические пособия, разработанные педагогом с учетом конкретных задач, упражнений, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. В процессе используется коллективная деятельность и индивидуальное обучение.

Кадровое обеспечение:

Программа реализуется педагогами дополнительного образования Прокочик Дарьей Николаевной, Дятловым Антоном Сергеевичем.

При реализации программы другим педагогом стоит учитывать, что педагогу необходимо познакомиться с технологией обучения программированию на языке Python.

2.3 Формы аттестации и оценочные материалы

Система отслеживания результатов обучающихся выстроена следующим образом:

- входящая диагностика;
- текущий контроль;
- промежуточная аттестация;
- итоговая аттестация.

Входящая диагностика по программе «Программирование на языке Python» проводится в начале освоения программы с целью выявления первоначального уровня знаний, умений и навыков обучающихся. Обучающемуся задается ряд общих вопросов, касающихся области алгоритмизации, информатики, программирования и информационных технологий в целом. Вопросы подготавливает педагог, примерный список вопросов представлен в Приложении.

Текущий контроль осуществляется путем наблюдения, определения качества выполнения заданий, отслеживания динамики развития обучающегося. Способы проверки уровня освоения тем: опрос, выполнение упражнений, наблюдение. Формы проведения итогов по каждой теме и каждому разделу общеразвивающей программы соответствуют целям и задачам ДООП.

Промежуточная аттестация осуществляется в форме решения задач и оценивается по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице 3.

Таблица 3

| Баллы, набранные обучающимся | Уровень освоения |
|------------------------------|------------------|
| 0–30 баллов | низкий |
| 31–70 баллов | средний |
| 71–100 баллов | высокий |

Результат решения задач оценивается педагогом. Примерные практические задания и критерии их оценивания представлены в Приложении.

В конце освоения дополнительной общеобразовательной общеразвивающей программы обучающиеся реализуется итоговая аттестация в виде защиты индивидуальных/групповых проектов. Темы для проектов обучающиеся выбирают на свое усмотрение при согласовании с педагогом или получают одну из тем проекта, подготовленных педагогом. Список предлагаемых тем проектов для выполнения обучающимися согласуется с учебной частью не позднее чем за 3 месяца до проведения итоговой аттестации по дополнительной общеобразовательной общеразвивающей программе «Программирование на языке Python».

Индивидуальный/групповой проект оценивается формируемой комиссией по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице 4:

Таблица 4

| Баллы, набранные обучающимся | Уровень освоения |
|------------------------------|------------------|
| 0–30 баллов | низкий |
| 31–70 баллов | средний |
| 71–100 баллов | высокий |

Критерии оценивания индивидуального проекта представлены в Приложении.

Состав комиссии (не менее 3 человек): в обязательном порядке входит педагог; приветствуется привлечение ИТ - профессионалов, представителей высших и других учебных заведений, администрации учебной организации.

Компонентами оценки индивидуального проекта являются (по мере убывания значимости): качество ИП, отзыв педагога, уровень презентации и защиты проекта. Если проект выполнен группой учащихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально.

2.4 Методические материалы

В образовательном процессе используются следующие *методы*:

- устные (беседы, лекции, объяснения);
- демонстрационные (демонстрация возможностей устройства);
- практические (написание программы).

Программой предусмотрены следующие виды деятельности обучающихся:

- работа с технической и справочной литературой;
- проектирование;
- программирование;
- эксперимент, испытание.

Выбор методов обучения осуществляется исходя из анализа уровня готовности обучающихся к освоению содержания раздела, степени сложности материала, типа учебного занятия.

Формы обучения:

– *фронтальная* – предполагает работу педагога сразу со всеми обучающимися в едином темпе и с общими задачами. Для реализации обучения используется компьютер педагога с мультимедиа проектором, посредством которых учебный материал демонстрируется на общий экран;

– *коллективная* – это форма сотрудничества, при котором коллектив обучает каждого своего члена и каждый член коллектива активно участвует в обучении своих товарищей по совместной учебной работе;

– *групповая* – предполагает, что занятия проводятся с подгруппой. Для этого группа разделяется на подгруппы не более 6 человек, работа в которых регулируется педагогом;

– *индивидуальная* – подразумевает взаимодействие педагога с одним обучающимся. Как правило данная форма используется в сочетании с фронтальной. Часть занятия (объяснение новой темы) проводится фронтально, затем обучающиеся выполняют индивидуальные задачи или общие задачи в индивидуальном темпе.

– **дистанционная** – взаимодействие педагога и обучающихся между собой на расстоянии, отражающее все присущие учебному процессу компоненты. Для реализации дистанционной формы обучения весь дидактический материал размещается в свободном доступе в сети Интернет, происходит свободное общение педагога и обучающихся в социальных сетях, по электронной почте, посредством видеоконференции или в общем чате. Кроме того, дистанционное обучение позволяет проводить консультации обучающихся при самостоятельной работе дома. Налаженная система сетевого взаимодействия обучающегося и педагога, позволяет не ограничивать процесс обучения нахождением в учебной аудитории, обеспечить возможность непрерывного обучения в том числе, для часто болеющих обучающихся или всех обучающихся в период сезонных карантинов (например, по гриппу) и температурных ограничениях посещения занятий.

Формы организации учебного занятия:

В образовательном процессе помимо традиционного учебного занятия используются многообразные формы, которые несут учебную нагрузку и могут использоваться как активные способы освоения обучающимися образовательной программы, в соответствии с возрастом, составом группы, содержанием учебного раздела: беседа, лекция, мастер-класс, практическое занятие, защита проектов, конкурс, викторина, диспут, круглый стол, «мозговой штурм».

Методы воспитания: мотивация, убеждение, поощрение, упражнение, стимулирование, создание ситуации успеха и др.

Педагогические технологии:

- индивидуализации обучения;
- группового обучения;
- коллективного взаимообучения;
- дифференцированного обучения;
- разноуровневого обучения;
- проблемного обучения;
- развивающего обучения;

- дистанционного обучения;
- игровой деятельности;
- коммуникативная технология обучения;
- коллективной творческой деятельности;
- решения изобретательских задач;
- здоровьесберегающая технология.

Дидактические материалы:

Методические пособия, разработанные педагогом с учетом конкретных задач, варианты демонстрационных программ, материалы по терминологии ПО, инструкции по настройке оборудования, учебная и техническая литература. Используются педагогические технологии индивидуализации обучения и коллективной деятельности.

Список литературы

Перечень учебной литературы, необходимой для освоения курса

Основные источники

1. Бизли Дэвид., Python. Исчерпывающее руководство – СПб: Питер, 2023. – 368 с.
2. Бейдер Дэн, Эймос Дэвид, Яблонски Джоанна, Хейслер Флетчер. Знакомство с Python. – СПб.: Питер, 2023. – 512 с.
3. Злотопольский Д.1400 задач по программированию. – М.: ДМК Пресс, 2020. – 192 с.

Дополнительные источники

1. Вейдт В.П. О воспитании подростков: в помощь классному руководителю: Методическое пособие / В.П. Вейдт. – Калининград: Издательство Калининградского областного института развития образования, 2019. – 140 с.
2. Дауни Аллен Д 21 Основы Python. Научитесь думать как программист / Аллен Б. Дауни ; пер. с англ. С. Черникова ; [науч. ред. А. Родионов]. — Москва : Манн, Иванов и Фербер, 2021. — 304 с.
3. Любанович Билл Л93 Простой Python. Современный стиль программирования. — СПб.: Питер, 2016. — 480 с.: ил. — (Серия «Бестселлеры O'Reilly»)

Электронные ресурсы

1. Python 3 для начинающих: <https://pythonworld.ru/samouchitel-python> (дата обращения 01.06.2023)
2. Домашняя страница Python: <http://www.python.org> (дата обращения 01.06.2023)
3. Питонтьютор. Бесплатный курс по программированию с нуля: <https://pythontutor.ru/> (дата обращения 01.06.2023)

Контрольно-измерительные материалы

к дополнительной общеобразовательной общеразвивающей программе

«Программирование на языке Python»

Дополнительной общеобразовательной программой «Программирование на языке Python» предусмотрены:

– Входящая диагностика – оценка исходного уровня знаний, умений и навыков обучающихся перед началом образовательного процесса. Форма – собеседование.

– Текущий контроль – это оценка качества усвоения обучающимися содержания общеобразовательной программы в период обучения. Форма – педагогическое наблюдение, опрос.

– Промежуточная аттестация - это оценка качества усвоения обучающимися содержания общеобразовательной программы по окончании полугодичного обучения. Форма – решение задач.

– Итоговая аттестация – это оценка качества усвоения обучающимися уровня достижений, заявленных в дополнительной общеобразовательной программе по завершении всего образовательного курса программы. Форма – защита индивидуального/группового проекта.

1. Материалы входящей диагностики

Входящая диагностика реализуется в форме собеседования. Вопросы для собеседования подготавливаются педагогом и могут изменяться. Примерные вопросы для собеседования представлены ниже:

1. Что такое блок-схема?
2. Что такое переменные?
3. Как работает программный код?
4. Что по-вашему значит «Поддерживать и обновлять программу»?
5. Алгоритм называется линейным, если ...

Примерные ответы на вопросы к собеседованию:

Таблица 1

| 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|
| Это графическое представление программы. Блок-схема помогает понять логику работы программы или ее части при проектировании | Это хранилище данных. Сюда можно положить какое-то значение (например, число, строку или другой тип данных) | Код программы содержит инструкции, которые являются исполняемыми командами, выполняемыми машиной. | Это процесс улучшения и обновления программы для удовлетворения новых требований к оборудованию или программному обеспечению. | Команды выполняются в порядке их естественного следования друг за другом независимо от каких-либо условий |

Критерии оценивания вопросов собеседования.

Каждый вопрос оценивается от 0 до 20 баллов:

0 баллов – ответ неверный.

10 баллов – ответ частично верный или раскрыт не полностью.

20 баллов – ответ верный.

Баллы за каждый вопрос суммируются и переводятся в уровень освоения согласно таблице 2.

Таблица 2

| Баллы, набранные обучающимся | Уровень освоения |
|------------------------------|------------------|
| 0–30 баллов | низкий |
| 31–70 баллов | средний |
| 71–100 баллов | высокий |

2. Материалы текущего контроля

Текущий контроль осуществляется путём наблюдения, определения качества выполнения заданий, отслеживания динамики развития обучающегося. Способы проверки уровня освоения тем: опрос, выполнение упражнений, наблюдение, оценка выполненных самостоятельных работ.

Примеры вопросов для опросов:

1. Как сравнивается (больше или меньше) число со строкой?
2. С помощью какой функции можно проверить принадлежность объекта определенному классу?
3. Чем цикл `while` отличается от цикла `for`?
4. У какого типа данных мы не можем получить элемент по индексу или ключу?
5. Что означает ошибка `IndexError`?
6. Может ли `lambda`-функция быть без аргументов?
7. С помощью какой библиотеки можно установить и управлять сторонними программными пакетами на Python?
8. Что значит режим открытия файла `'x'`.

Ключ к опросу:

1. При таком сравнении происходит ошибка.
2. С помощью функции `isinstance`.
3. `while` – при задании неизвестного кол-ва прохождения цикла, `for` – строго заданное кол-во прохождения цикла.
4. У типа множества – `set`.
5. Индекс не входит в диапазон элементов.
6. `Lambda`-функция может быть без аргументов.
7. Модуль `pip`.
8. Открытие на запись, если файла не существует, иначе исключение.

3. Материалы промежуточной аттестации

Промежуточная аттестация осуществляется в форме решения задач.

Пример задачи для промежуточной аттестации:

В ходе ожесточенной схватки пиратам удалось заполучить карту сокровищ, но взамен их корабль сильно потрепало. Кое-как они успели добраться до ближайшего порта города N. Один капитан торгового судна согласился отвести капитана пиратов и двух его помощников на остров с сокровищами, но только с условием: они оставляют их на острове и плывут дальше в город M. Ровно через 8 часов они будут возвращаться в город N и заберут их с острова. Но если они задержатся, то корабль уплывет дальше, и пираты останутся на острове. Такой расклад устроил пиратов. Приплыв на остров и развернув карту, они увидели, что предыдущий владелец подстраховался и нарисовал дополнительные кресты. Их было 3 штуки. Подсчитайте, хватит ли времени у пиратов обойти все метки на карте, чтобы затем уплыть с кладом.

Карта представляет собой клеточное поле размером 10x10. Пираты втроем стоят в левом нижнем углу. Каждый пират может двигаться отдельно для экономии времени. Передвижение (переход с одной клетки карты на другую) занимает 15 минут. Переход может быть совершен в одну из четырех сторон: прямо, назад, влево, вправо. (Двигаться по диагонали они не могут). Еще нужно раскопать метку, чтобы проверить на наличие клада – это занимает 30 минут. Также учитывайте, что всем пиратам нужно вернуться в начальную точку сбора. Уехать с острова должны все пираты.

Формат ввода:

В строке через запятые вводятся 3 пары натуральных чисел, сами пары через пробел — координаты всех меток. (Нумерация координат начинается с правого нижнего угла).

Формат вывода:

В первой строке выводится три числа – сколько времени в минутах занял поиск сокровищ (т.е. с момента прибытия на остров и до момента возвращения на начальную точку сбора) для каждого пирата в любом порядке.

Во второй строке вывести слово YES – если пираты успели на торговый корабль, и NO – если не успели.

Пример 1:

Ввод:

2 3, 5 9, 7 6

Вывод:

360 300 450

YES

Пояснение: Как выглядит карта:

| | |
|----|----------------------|
| 0 | 1 2 3 4 5 6 7 8 9 10 |
| 10 | 0 0 0 0 0 0 0 0 0 0 |
| 9 | 0 0 0 0 x 0 0 0 0 0 |
| 8 | 0 0 0 0 0 0 0 0 0 0 |
| 7 | 0 0 0 0 0 0 0 0 0 0 |
| 6 | 0 0 0 0 0 0 x 0 0 0 |
| 5 | 0 0 0 0 0 0 0 0 0 0 |
| 4 | 0 0 0 0 0 0 0 0 0 0 |
| 3 | 0 x 0 0 0 0 0 0 0 0 |
| 2 | 0 0 0 0 0 0 0 0 0 0 |
| 1 | 0 0 0 0 0 0 0 0 0 п |

Результаты решения задачи оцениваются по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице 3.

Таблица 3

| Баллы, набранные обучающимся | Уровень освоения |
|------------------------------|------------------|
| 0–30 баллов | низкий |
| 31–70 баллов | средний |
| 71–100 баллов | высокий |

Критерии оценки задачи (баллы от 0 до максимального (10)):

1. программа не выдает ошибок (Максимально 10 баллов);
2. степень завершенности проекта (Максимально 30 баллов);
3. код выдает правильный результат с тестовыми данными (Максимально 10 баллов);
4. код выдает правильный результат с данными, не представленными в задании (Максимально 10 баллов);
5. наличие структуры решения (Максимально 10 баллов);
6. сложность алгоритмов, структур данных (Максимально 10 баллов);
7. объем кода (Максимально 10 баллов);
8. наличие комментариев в коде (Максимально 10 баллов).

4. Материалы итоговой аттестации

Итоговая аттестация осуществляется в форме защиты индивидуальных/групповых проектов.

Темы для проектов обучающиеся выбирают на свое усмотрение при согласовании с педагогом или получают одну из тем проекта, подготовленных педагогом. Список предлагаемых тем проектов для выполнения обучающимися согласуется с учебной частью не позднее чем за 3 месяца до проведения итоговой аттестации по дополнительной общеобразовательной общеразвивающей программе «Программирование на языке Python».

Индивидуальный/групповой проект оценивается формируемой комиссией по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице 4.

Таблица 4

| Баллы, набранные обучающимся | Уровень освоения |
|------------------------------|------------------|
| 0–30 баллов | низкий |
| 31–70 баллов | средний |
| 71–100 баллов | высокий |

Состав комиссии (не менее 3 человек): в обязательном порядке входит педагог; приветствуется привлечение ИТ - профессионалов, представителей высших и других учебных заведений, администрации учебной организации.

Компонентами оценки индивидуального/группового проекта являются (по мере убывания значимости): качество ИП, отзыв педагога, уровень презентации и защиты проекта. Если проект выполнен группой учащихся, то при оценивании учитывается не только уровень исполнения проекта в целом, но и личный вклад каждого из авторов. Решение принимается коллегиально.

Критерии оценки индивидуального/группового проекта (критерий

Баллы от 0 до максимального (10)):

1. новизна проекта (Максимально 10 баллов);
2. степень завершенности проекта (Максимально 10 баллов);
3. креативность проекта (Максимально 10 баллов);
4. использование современных библиотек (Максимально 10 баллов);
5. наличие структуры проекта (Максимально 10 баллов);
6. сложность алгоритмов, структур данных (Максимально 10 баллов);
7. объем своего кода (Максимально 10 баллов);
8. презентация (Максимально 10 баллов);
9. наличие дизайна (Максимально 10 баллов);
10. наличие комментариев в коде (Максимально 10 баллов).