


Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске
«Центр цифрового образования детей IT-Куб г. Норильск»

РЕКОМЕНДОВАНО

Заведующий по учебной части

 Н. В. Грицюк

протокол № 9

от «5» сентября 2023 г.

УТВЕРЖДАЮ



Е. А. Дыптан

Приказ № 20-02-36

от «5» сентября 2023 г.

Дополнительная общеобразовательная общеразвивающая
программа технической направленности

«Python 3 для новичков. Основы Python»

Срок реализации: 1 год

Возраст детей: 13-17 лет

Составители программы: Комаров Е. А.

Норильск, 2023

СОДЕРЖАНИЕ

1. Пояснительная записка	3
1.1. Новизна ДООП	3
1.2. Актуальность ДООП	4
1.3. Педагогическая целесообразность	4
1.4. Цель ДООП	4
1.5. Возраст обучающихся, участвующих в ДООП	5
1.6. Условия вхождения в ДООП	5
1.7. Срок реализации ДООП	5
1.8. Режим занятий, формы и методы обучения	5
1.9. Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов	6
2. Учебно-тематический план	8
3. Содержание программы	11
4. Материально-техническое и информационно-методическое обеспечение	17
5. Рекомендуемая литература	18
6. Сведения о составителях ДООП	19
7. Приложение 1	20

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

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

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне т.к. выделяется малое количество времени для изучения программирования. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является лишь формальное восприятие учащимися основ программирования и неумение применять полученные знания на практике.

Изучение программирования невозможно без регулярной практики написания программ на каком-либо языке. В данной программе для изучения выбран язык программирования Python. Этот выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это снижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических частях программирования, а не на заучивании синтаксиса языка. При этом Python является очень востребованным языком; он отлично подходит для знакомства с различными современными парадигмами программирования и активно применяется в самых разных областях от разработки веб-приложений до машинного обучения.

1.1 Новизна ДООП

Новизна программы заключается в возможности получения обучающимися универсальных компетенций, необходимых при дальнейшем изучении не только информационных технологий, но и предметов гуманитарного и естественно-научного цикла. Также в результате изучения парадигмы объектно-ориентированного подхода к программированию происходит формирование базовых знаний и умений для работы с большинством популярных языков и необходимых при освоении других IT-направлений. Программа является практико-ориентированной. Освоение подростками навыков программирования происходит в процессе практической и самостоятельной работы. Это позволяет обучающимся получать не только теоретические знания в области программирования, но и уверенно овладевать навыками и инструментами разработки продуктов.

1.2 Актуальность ДООП

Актуальность обусловлена повышенным спросом на изучение языков программирования детьми, в частности языка Python. Python изучается в школьном курсе информатики, необходим для решения олимпиадных и конкурсных заданий, а также сдачи ЕГЭ, имеет прикладной характер и может использоваться для решения повседневных задач. Кроме того, Python является востребованным языком программирования, используемым профессиональными инженерами во многих сферах IT-индустрии, поэтому знание данного языка даже на базовом уровне повышает шансы будущих выпускников на трудоустройство.

1.3 Педагогическая целесообразность:

В связи с простой синтаксиса в сравнении с другими языками программирования (ясность кода, быстрота реализации) при изучении Python есть возможность сформировать у обучающихся представление о базовых понятиях структурного программирования (данных, переменных, ветвлениях, циклах и функциях). В то же время Python является востребованным языком, он отлично подходит для концепции объектно-ориентированного программирования и активно применяется в различных областях от разработки веб-приложения до машинного обучения. Научившись программировать на языке Python, обучающиеся получают мощный и удобный в использовании инструмент для решения учебных задач и для создания собственных проектов. Вместе с тем, чистота и ясность его конструкций позволит в дальнейшем с легкостью выучить любой другой язык программирования.

1.4 Цель ДООП.

Обучить основным принципам и этапам программирования и разработки программного обеспечения на основе языка программирования Python для последующего решения поставленных технических задач, а также для реализации творческих проектов.

Обучающие задачи:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- изучение конструкций языка программирования Python;
- знакомство с модулями Python: math, python-docx, tkinter, requests и др;
- знакомство с объектно-ориентированным программированием;
- знакомство с разработкой новых классов данных;
- приобретение навыка работы в интегрированной среде разработки на языке Python;
- знакомство с понятием проекта и алгоритмом его разработки.

Воспитательные задачи:

- развитие у учащихся инициативность и самостоятельность;
- мотивация к созданию собственных проектов;
- развитие стремление к получению качественного законченного результата в проектной деятельности;
- воспитание социально значимые качества личности человека: ответственность, коммуникабельность, добросовестность, пытливость ума и критичность мышления.

Развивающие задачи:

- развитие логического, абстрактного и образного типов мышления;
- развитие творческих способностей;
- формирование самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач.

Программа представляет 4 этапа. Обучение начинается со знакомства с языком Python и его синтаксисом. Учащиеся узнают, как работают переменные, какие типы данных бывают, как работать с коллекциями данных. Затем происходит знакомство с функциями Python - передача аргументов, декораторы, учащиеся работают с функциями высшего порядка. На 3-м этапе происходит знакомство с распространенными библиотеками. Последний этап - объектно-ориентированное программирование.

Особенностью данной программы является привлечение детей среднего школьного возраста к изучению программирования при помощи языка Python, т. к. он обладает следующими достоинствами:

- Python – это текстовый язык программирования. Он универсален, пригоден для создания самых разнообразных программ, от текстовых процессоров до веб-браузеров;
- Python – простой и удобный язык. По сравнению со многими другими языками читать и составлять программы на Python несложно;
- В Python есть библиотеки готовых процедур для использования в своих программах. Это позволяет создавать сложные программы быстро;
- Python используется как язык программирования крупными корпорациями, такими Google.

1.5 Возраст обучающихся, участвующих в ДООП

Программа «Python3 для новичков. Основы Python» ориентирована на обучающихся от 13 до 17 лет. Занятия проводятся в группе из 12 человек.

1.6 Условия вхождения в ДООП

Набор на Программу осуществляется в соответствии с Положением о наборе в Филиал АНО ДТ «Красноярский «Кванториум» в г. Норильске «Центр цифрового образования детей IT-Куб г. Норильск».

Поступающий на программу должен владеть базовыми умениями работы на компьютере (создание и удаление файлов; работа в простейшем текстовом и графическом редакторе; запуск, просмотр презентаций и видеороликов).

1.7 Срок реализации ДООП

Программа рассчитана на 1 учебный год. Нагрузка на обучающегося составляет 144 учебных часа за год.

1.8 Режим занятий, формы и методы обучения

Учебные занятия проходят по очной форме обучения. Режим занятий – 2 раза в неделю по 2 академических часа (1 академический час 40 минут) с обязательным перерывом.

При проведении занятий используются комбинированные занятия – изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа.

При проведении занятий используются следующие формы работы:

- демонстрационная, когда обучающиеся слушают объяснения педагога и наблюдают за демонстрационным экраном или экранами компьютеров на ученических рабочих местах;
- фронтальная, когда обучающиеся синхронно работают под управлением педагога;
- самостоятельная, когда обучающиеся выполняют индивидуальные задания в течение части занятия.

1.9 Ожидаемые образовательные результаты и эффекты, способы предъявления и отслеживания результатов.

Учащиеся будут уметь:

- объяснять и использовать на практике как простые, так и сложные структуры данных и конструкций для работы с ними;
- искать и обрабатывать ошибки в коде;
- разбивать решение задач на подзадачи;
- писать грамотный и красивый код;
- находить, оценивать, использовать информацию из различных источников, необходимую для решения профессиональных задач, в том числе на основе системного подхода;
- грамотно работать в команде, в зависимости от целей и ситуации.

Учащиеся будут знать:

- основные классические алгоритмы и способы их реализации;
- синтаксис языка программирования Python;
- основные элементы программирования: ввод/вывод информации, данные, операции с данными, условное выполнение, циклы, подпрограммы.

Сформированные навыки:

- соблюдение требований техники безопасности;
- работы в изучаемых программных средах;
- навыки составления алгоритмов;
- применения на практике основных команд и операторов изучаемых языков;
- разработки, тестирования и отладки несложных программ;
- навыки работы в сети Интернет для поиска информации.

Личностные результаты:

- Учащиеся приобретут навыки самостоятельной организации своей деятельности; формирования основ саморазвития и самовоспитания.
- У обучающихся сформируется готовность и способность к самостоятельной, творческой деятельности, к образованию, в том числе самообразованию; готовность к осознанному выбору будущей профессии.

Метапредметные результаты:

- умение планировать; умение анализировать; алгоритмизировать;

- коммуникативные навыки:
 - умение договариваться с другими людьми;
 - работать в команде;
 - аргументировать свою позицию;
 - развить эмоциональный интеллект – способность понимать чужие чувства и контролировать свои;
- навыки самоорганизации и тайм-менеджмент;
- нестандартное мышление, креативные навыки;
- умение работать с информацией:
 - анализ информации;
 - компьютерная грамотность;
- стрессоустойчивость.

Опыт:

Проектной деятельности, создания, редактирования, оформления, сохранения, передачи информационных объектов различного типа с помощью современных программных средств; информационной деятельности в различных сферах; эффективного применения информационных образовательных ресурсов в учебной деятельности, в том числе самообразовании; эффективной организации индивидуального информационного пространства.

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	Введение в программирование	22	10	12	
1.1.	Введение.	2	1	1	
1.2.	Устройство языка Python.	2	1	1	
1.3.	Данные	2	1	1	
1.4.	Обработка данных	2	1	1	
1.5.	Построение командной работы.	2	1	1	
1.6.	Условный оператор	2	1	1	
1.7.	Циклы	2	1	1	
1.8.	Команды циклов остановок и прерывания циклов.	2	1	1	
1.9.	Циклы с подсчетом «range».	2	1	1	
1.10.	Вложенные циклы.	2	1	1	
1.11.	Групповая работа. Решение задач.	2	0	2	Практическая работа
2	Базовые конструкции Python	26	12	14	
2.1	Знакомство с коллекциями.	2	1	1	
2.2	Знакомство со строками.	2	1	1	
2.3.	Срезы в строках.	2	1	1	
2.4	Методы строк.	2	1	1	
2.5	Знакомство со списками.	4	2	2	
2.6	Вложения.	2	1	1	

2.7	Кортежи.	2	1	1	
2.8	Словари	2	1	1	
2.9	Множества.	2	1	1	
2.10	Файлы.	2	1	1	
2.11	Групповая практическая работа.	4	1	3	Практическая работа
3	Знакомство с функциями	26	9	17	
3.1	Функции.	4	1	3	
3.2	Области видимости.	2	1	1	
3.3	Передача параметров в функции.	4	2	2	
3.4	Функции с переменным числом аргументов	2	1	1	
3.5	Функции как объект. Лямбда-функции.	6	2	4	
3.6	Обработка коллекций. Поточковый ввод sys.stdin	4	1	3	
3.7	Повторение пройденного материала.	2	2	0	
3.8	Аттестация по итогу пошедшего материала	2	0	2	Выполнение проверочной работы
4	Работа с Модулями	36	12	24	
4.1	Модули.	2	1	1	
4.2	Знакомство с модулями Python. Часть №1 (random)	2	1	1	
4.3	Знакомство с модулями Python. Часть №2 интерфейс. Tkinter	10	2	8	
4.4	Знакомство с модулями Python Часть №3 (морфология). Морфологический анализатор русского языка модуль rymorphy2.	2	1	1	
4.5	Морфологический анализ текста с помощью rymorphy2.	2	1	1	
4.6	Знакомство с модулями Python. Часть №4 работа с документами. Модуль python - docx	4	2	2	
4.7	Знакомство с модулями Python. Часть №5 работа с различными типами файлов. Модули для работы с json-файлами	2	1	1	
4.8	Знакомство с модулями Python. Часть №6 работа с модулем requests.	2	1	1	

4.9	Работа с запросами по API	2	0	2	
4.10	Групповая практическая работа.	8	2	6	Практическая работа
5	Введение в ООП	34	10	24	
5.1	Введение в ООП.	2	1	1	
5.2	Основы написания классов.	6	3	3	
5.3	Проектирование и разработка классов.	6	2	4	
5.4	Групповая практическая работа	4	1	3	
5.5	Разработка проекта.	16	3	13	Защита проекта
	Итого	144	55	89	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

ТЕМА 1. Введение в программирование

1.1. Введение.

Теоретическая работа: Вводная лекция, знакомство с учениками. Проведение инструктажа по технике безопасности. Знакомство с историей языков программирования.

Практическая работа: Знакомство со средой разработки (IDLE) Pycharm. Установка и первый запуск.

1.2. Устройство языка Python.

Теоретическая работа: Иерархическое устройство языка Python. Знакомство с соглашением, о том, как писать код для языка Python per8.

Практическая работа: Использование команд input и print.

1.3. Данные.

Теоретическая работа: Знакомство с основными типами данных. Правила объявления переменных. Знакомство с Арифметическими действиями с переменными.

Практическая работа: Использование умножения, деления, сложения, вычитания с переменными одного типа, переменными разных типов.

1.4. Обработка данных.

Теоретическая работа: Знакомство с различными способами ввода данных.

Практическая работа: Решение простейших задач на работу с арифметическими действиями.

1.5. Построение командной работы.

Теоретическая работа: знакомство с групповой работой. Разделение задач на подзадачи. Умение ставить цели и задачи для группы.

1.6. Условный оператор.

Теоретическая работа: Знакомство с логическими функциями и/или/не, использование логических элементов в ветвлении. Правила оформления блоков ветвления с помощью отступов.

Практическая работа: Создание ветвлений, использование множественных ветвлений при решении задач.

1.7. Циклы.

Теоретическая работа: Знакомство с понятием цикла, общий вид цикла.

Практическая работа: Практическое использование циклов While и for. Практическое использование отладчика в Pycharm

1.8. Команды циклов остановок и прерывания циклов.

Теоретическая работа: Знакомство с командами остановки и прерывания цикла break, continue. Преимущества и недостатки использования этих команд.

Практическая работа: Апробация полученных знаний на практике при решении задач.

1.9. Циклы с подсчетом «range».

Теоретическая работа: Знакомство с функцией range в циклах for. Генерация ряда чисел в заданном диапазоне.

Практическая работа: Использование аргументов при вызове функции range (). Создание возрастающих и убывающих последовательностей с различным шагом.

1.10. Вложенные циклы.

Теоретическая работа: Знакомство с вложенными циклами.

1.11. Групповая работа. Решение задач.

Практическая работа: Разделение класса на группы. Разделение обязанностей в группе. Решение задач по пройденным темам.

ТЕМА 2. Базовые конструкции в Python

2.1 Знакомство с коллекциями.

Теоретическая работа: Коллекция – «контейнер», содержащий различные элементы.

2.2 Знакомство со строками.

Теоретическая работа: Знакомство со строками. Функции и методы строк. Индексация. Операции над строками. Конкатенация. Повторение

Практическая работа: Выполнение базовых операций над строками: конкатенация, дублирование, определение длины строки (функция len), доступ к символам по индексу.

2.3. Срезы в строках.

Теоретическая работа: Знакомство со срезами, аргументы среза, отрицательные параметры.

Практическая работа: решение задач на извлечение среза из строки.

2.4 Методы строк.

Теоретическая работа: Знакомство с методами специфичными для строк.

Практическая работа: Преобразование строк с помощью метода split. Поиск символов по строке с помощью метода find, замена символов в строке с помощью символа replace.

2.5. Знакомство со списками.

Теоретическая работа: что такое список, его отличие от строки, различные операции над списками, методы специфичные для списка.

Практическая работа: Решение задач на арифметические действия со списками, поиск минимального или максимального числа в списке, добавить элемент или удалить элемент из списка.

2.6. Вложения.

Теоретическая работа: Строки, вложенные в списки, и списки, вложенные в списки,

особенности индексации. Как обработать матрицу с помощью вложений.

2.7. Кортежи.

Теоретическая работа: Что такое кортежи, как и для чего их используют. В чем отличие от списков.

Практическая работа: Решение задач на использование кортежей.

2.8. Словари

Теоретическая работа: Способы создания словарей. Генераторы словарей. Ключи. Способы создания. Операции над словарями. Генераторы словарей. Методы: Clear, сору и др.

Практическая работа: Выполнение заданий по созданию словарей. Поиску по ключу, изменение и вывод словарей.

2.9. Множества.

Теоретическая работа: что такое множество, методы множества.

Практическая работа: Решение задач на создание, преобразование, удаление коллекций. Использование циклов для получения данных из коллекций.

2.10. Файлы.

Теоретическая работа: Способы создания файла. Экспорт данных из файла. Чтение файла. Сохранение в файл.

Практическая работа: Получение exe файла из проекта в Pycharm с помощью pyinstaller.

2.11. Групповая практическая работа.

Практическая работа: Применение навыков тимбилдинга для распределения по группам. Получение проблемной ситуации. Групповая работа по решению проблемы.

ТЕМА 3. Знакомство с функциями

3.1. Функции.

Теоретическая работа: Знакомство с функциями. Как возвращать значение из функции. Возвращения значений из функции. Инструкции return, def.

Практическая работа: с помощью инструкции def объявляем функцию, с помощью инструкции return возвращаем значение функции.

3.2. Области видимости.

Теоретическая работа: Области локальной и глобальной видимости, области нелокальной видимости. Операторы global, nonlocal,

Практическая работа: Использование инструкций global и nonlocal, при решении задач с функциями.

3.3. Передача параметров в функции.

Теоретическая работа: Отличие между переменной и значением. Функции, изменяющие значение аргумента. Id Объектов. Изменяемость и неизменяемость объектов.

Практическая работа: Написание программ с получением функцией различных значений, и преобразование значений согласно коду программы.

3.4. Функции с переменным числом аргументов.

Теоретическая работа: Распаковка и запаковка значений. Аргументы по умолчанию. Именованные аргументы. Инструкция pass. Согласованность аргументов.

Практическая работа: Работа с функциями. Получение неизвестного количества именованных и неименованных аргументов.

3.5. Функции как объект. Лямбда-функции

Теоретическая работа: Функция как объект. Функция filter, Лямбда-функции, Функция map. Списочные выражения. Поточковый ввод.

Практическая работа: Применение функций высшего порядка для решения задач.

3.6. Обработка коллекций. Поточковый ввод sys.stdin

Теоретическая работа: Итерируемые объекты. Функции max, min, sorted и использование ключей сортировки. Функции any и all. Поточковый ввод sys.stdin.

ТЕМА 4. Работа с модулями

4.1. Модули.

Теоретическая работа: Способы установки модулей. Импортирование функций из модулей. Импортирование всех функций из модуля, перезагрузка модуля.

Практическая работа: Знакомство с Pip, работа с командной строкой.

4.2. Знакомство с модулями Python. Часть №1 (random)

Теоретическая работа: Знакомство со встроенным модулем Random, основные функции генератора.

Практическая работа: Разбор функций генератора, написание программ с использованием генератора чисел.

4.3. Знакомство с модулями Python. Часть № 2 интерфейс Tkinter

Теоретическая работа: Знакомство с графическими модулями. Создание пользовательского интерфейса для программ. Расположение элементов пользовательского интерфейса в окне.

4.4. Знакомство с модулями Python Часть №3 (морфология).

Морфологический анализатор русского языка модуль rymorphy2.

Теоретическая работа: Морфологический анализатор rymorphy2. Функции. Разбор слов.

4.5. Морфологический анализ текста с помощью rymorphy2.

Практическая работа: Практическая задача на морфологический разбор текста, поиск слова в тексте, поиск ошибок в тексте.

4.6. Знакомство с модулями Python. Часть №4 работа с документами.

Модуль python – docx

Теоретическая работа: Знакомство с функциями модуля python-docx, создание шаблонов для заполнения текстового документа.

Практическая работа: Практическая задача на морфологический разбор текста, поиск слова в тексте, поиск ошибок в тексте.

4.7. Знакомство с модулями Python. Часть №5 работа с различными типами файлов. Модули для работы с json-файлами

Теоретическая работа: Знакомство с типами файлов CSV, JSON, YAML, XML, проблемы кодировок.

Практическая работа: Получение данных из файлов форматов: CSV, JSON, YAML, XML.

4.8. Знакомство с модулями Python. Часть №6 работа с модулем requests.

Теоретическая работа: Что такое Http, запросы http, методы http, параметры в адресе заголовки, коды ответов, библиотека Requests

Практическая работа: Обработка Request запросов, получение данных.

4.9. Работа с запросами по API.

Теоретическая работа: Примеры работы с API. Работа с яндекс.полигон. Примеры других полигонов и API.

Практическая работа: Сохранение файла на яндекс.диске.

4.10. Групповая практическая работа.

Теоретическая работа: Постановка задачи и разбиение на группы. Знакомство с API VK

Практическая работа: Выполнение работы по поставленной задаче.

ТЕМА 5. Введение в ООП

5.1. Введение в ООП.

Теоретическая работа: Знакомство с особенностями ООП в общем и его реализацией в языке Python. Классы. Методы. Вызов методов.

Практическая работа: Написание собственных классов.

5.2. Основы написания классов.

Теоретическая работа: Знакомство с наследованием параметров класса. Оператор class. Наследование. Дерево атрибутов.

Практическая работа: Создание классов и наследование между ними.

5.3. Проектирование и разработка классов.

Теоретическая работа: Знакомство с дополнительными функциями и методами при создании классов.

Практическая работа: Выполнение заданий на написание классов.

5.4. Групповая практическая работа

Практическая работа: Решение задач

5.5 Разработка проекта.

Теоретическая работа: Распределение по группам, выбор темы проекта.

Практическая работа: Выполнение проекта по выбранной теме. Подготовка к защите проекта. Защита проекта.

4. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ И ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ

Материально – техническое обеспечение:

1. Стол преподавателя
2. Стул преподавателя
3. Стол обучающегося
4. Стул обучающегося
5. Рабочая станция преподавателя
6. Ноутбук обучающегося
7. Интерактивная доска
8. Проектор
9. МФУ
10. Точки подключения к электрической сети

Программное обеспечение:

11. Операционная система Windows 10
12. Пакет программ MS OFFICE
13. PyCharm

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная литература:

- К. Ю. Поляков, Е. А. Еремин. Информатика. Углубленный уровень.
- Учебник для 10 класса в двух частях. М.: БИНОМ. Лаборатория знаний, 2019
- М. Лутц. Изучаем PYTHON. СПб.: Символ-Плюс, 2019
- Задачи по программированию. Под ред. С. М. Окулова. М.: БИНОМ, Лаборатория знаний, 2017
- С. М. Окулов. Основы программирования. М.: БИНОМ. Лаборатория знаний, 2017

Дополнительная литература:

- Информатика и ИКТ. Задачник – практикум в двух частях. Под ред. И. Г. Семакина и Е.К. Хеннера. М.: БИНОМ. Лаборатория знаний, 2014

Электронные ресурсы:

- Сайт pythonworld.ru – «Python 3 для начинающих»
- Сайт pythontutor.ru – «Питонтьютор»

6. СВЕДЕНИЯ О СОСТАВИТЕЛЯХ ДООП

Комаров Е.А. педагог дополнительного образования, преподаватель по направлениям «Python для новичков» и «VR/AR». Образование высшее. ФГБОУ ВО «Казанский государственный энергетический университет» диплом по направлению «Электроэнергетика и электротехника» квалификация «Бакалавр». АНО ДПО «ФИПКИП» квалификация «учитель информатики»

ПРИМЕРЫ ЗАДАНИЙ И КРИТЕРИИ ОЦЕНИВАНИЯ

Примеры заданий для проверочной работы:

1. Что будет напечатано при выполнении следующего кода? `print(type(1/2))`

<input type="radio"/> class <'int'>	<input type="radio"/> class <'float'>
<input type="radio"/> class <'number'>	<input type="radio"/> class <'tuple'>

2. Что выведет следующая программа?

```
x = 4.5
y = 2
print(x // y)
```

<input type="radio"/> 2	<input type="radio"/> 9
<input type="radio"/> 2.25	<input type="radio"/> 20.25

3. Что будет напечатано при запуске программы?

```
x = True
y = False
z = False
if not x or y:
    print(1)
elif not x or not y and z:
    print(2)
elif not x or y or not y and x:
    print(3)
else:
    print(4)
```

<input type="radio"/> 1	<input type="radio"/> 3
<input type="radio"/> 2	<input type="radio"/> 4

4. Как обозначается тип данных целых чисел?

<input type="radio"/> Float	<input type="radio"/> Int
<input type="radio"/> Str	<input type="radio"/> Vool

5. Что будет выведено на экран?

```
a=5
b=6
print(a)
```

<input type="radio"/> 5	<input type="radio"/> 11
<input type="radio"/> 6	<input type="radio"/> 1

6. Чему будет выведено на экран после выполнения кода?

a = 5

b = 10/2

```
print(a==b)
```

<input type="radio"/> True	<input type="radio"/> False
<input type="radio"/> ошибка	<input type="radio"/> 0

7. Чему равно c?

a=2

b=5

```
if a>b:
```

```
    c=a
```

```
else:
```

```
    c=b
```

<input type="radio"/> 2	<input type="radio"/> 0
<input type="radio"/> 5	<input type="radio"/> Нет правильного ответа

8. До каких пор будут выполняться операторы в теле цикла: while x<100:

<input type="radio"/> Пока x меньше или равен 100	<input type="radio"/> Пока x больше 100
<input type="radio"/> Пока x меньше 100	<input type="radio"/> Пока x равен 100

9. Какие значения будет принимать переменная i, при организации цикла:

For i in range(12,3,-3):

<input type="radio"/> 9, 6, 3	<input type="radio"/> 9, 6
<input type="radio"/> 12, 9, 6, 3	<input type="radio"/> 12, 9, 6

10. Дана программная строка: for i in range(0,10,2): , число 2 - это...

<input type="radio"/> Начало цикла	<input type="radio"/> Счетчик
<input type="radio"/> Конец цикла	<input type="radio"/> Шаг

Критерий оценивания.

Оценка за контроль ключевых компетенций учащихся производится по пятибалльной системе. При выполнении заданий ставится отметка:

«3» - за 50-70% правильно выполненных заданий,

«4» - за 70-85% правильно выполненных заданий,

«5» - за правильное выполнение более 85% заданий.