

**МУНИЦИПАЛЬНОЕ ОБЩЕОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ГРОМОВСКАЯ СРЕДНЯЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ ШКОЛА
ПРИОЗЕРСКОГО РАЙОНА ЛЕНИНГРАДСКОЙ ОБЛАСТИ**



ПРИНЯТО
Педагогическим советом
Протокол № 8
от 26 августа 2024 г

УТВЕРЖДЕНА
и.о. директора МОУ «Громовская СОШ»
Русакова О.А.
Приказ № 183 от «30» августа 2024 г



**Дополнительная общеразвивающая программа
технической направленности**

«Инфознайка» (Точка роста)

Возраст детей: 12-14 лет

Срок реализации: 2 года обучения

Разработчик: Даутова Ю.С.

п. Суходолье
2024 г.

Пояснительная записка

Дополнительная общеразвивающая программа «Инфознайка» имеет техническую направленность и реализует дополнительное образование в рамках реализации национального проекта «Точка роста»:

Программа разработана на основе:

Федеральный закон от 29.12.2012 № 273-ФЗ (ред. от 31.07.2020) «Об образовании в Российской Федерации» (с изм. и доп., вступ. в силу с 01.09.2020).

Методические рекомендации по проектированию дополнительных общеобразовательных общеразвивающих программ от 18.11.2015. Министерство образования и науки РФ

Профессиональный стандарт «Педагог (педагогическая деятельность в дошкольном, начальном общем, основном общем, среднем общем образовании), (воспитатель, учитель)»

Приказ Министерства просвещения Российской Федерации от 9 ноября 2018 г. № 196 «Об утверждении Порядка организации и осуществления образовательной

деятельности по дополнительным общеобразовательным программам»

Федеральный Закон от 29.12.2012 № 273-ФЗ «Об образовании в РФ»

Федеральный проект «Успех каждого ребенка» (утв. 7 декабря 2018 г.)

Распоряжение Администрации Ленинградской области от 04.07.2019 № 407-р

«О реализации мероприятия по созданию Центров образования цифрового и гуманитарного профилей «Точка роста» в рамках федерального проекта «Современная школа» национального проекта «Образование»

Программирование – одна из тем курса информатики. Но какова роль программирования и есть ли необходимость его изучать? Часто говорят, что в современную эпоху развитого прикладного программного обеспечения изучение программирования потеряло свое значение как средство подготовки основной массы школьников к профессиональной деятельности. С одной стороны, это действительно так, но, с другой стороны, изучение основ программирования связано с развитием целого ряда таких умений и навыков (организация деятельности, ее планирование и т.д.), которые носят общеинтеллектуальный характер и формирование которых – одна из приоритетных задач современной школы.

Курс «Учим PYTHON» позволяет посредством формирования начальных навыков программирования подготовить платформу для изучения в дальнейшем более сложных языков программирования. Данный учебный курс позволит учащемуся, прошедшему обучение, самостоятельно моделировать алгоритмические конструкции.

Одним из средств достижения запланированных результатов является язык программирования «PYTHON»

Особенности языка python:

- язык очень легок в изучении;
- язык является объектно-ориентированным, поэтому его реализация является самой понятной
- на python можно написать любую программу;
- широкий спектр применения;
- python работает во всех операционных системах, так как это интерпретируемый язык;
- python распространяется свободно на условиях лицензии GNU 2.0. Данная лицензия разрешает бессрочно использовать python на любом количестве компьютеров в любых целях без оформления каких-либо дополнительных документов.

Поэтому целесообразно использовать этот язык при изучении программирования в среднем звене. В связи с этим становится очевидным актуальность предлагаемого курса.

Актуальность и отличительные особенности программы.

На сегодняшний день существует множество языков программирования, каждый из которых имеет свои особенности. Но хочется выделить Python, как популярную универсальную среду разработки программного кода с тридцатилетней историей.

Что такое Python?

В конце 1989 года Гвидо Ван Россум создал Python – новый интерпретируемый язык программирования, который очень быстро стал популярен и востребован у программистов. В подтверждение этому можно привести компании-гиганты, которые используют Python для реализации глобальных проектов. Это Google, Microsoft, Facebook, Yandex... Список этот просто огромен. Не обошлось без Python при создании сервисов YouTube, Wargaming, Instagram, Mozilla и многих других.

Область применения Python очень обширна. Его используют для создания самых различных типов задач. Это скрипты по автоматизации, системы управления жизнеобеспечением, игры, всевозможные приложения, веб-ресурсы...

За все время существования Python плодотворно использовался и динамично развивался. Создавались стандартные библиотеки для поддержки современных технологий, например, работа с базами данных, протоколами Интернета, электронной почтой и многое другое.

Особенности Python

Python, как и любой другой язык программирования, имеет свои отличительные особенности. Итак, можно выделить следующие:

- Кроссплатформенность. Python – это интерпретируемый язык, его интерпретаторы существуют для многих платформ. Поэтому с запуском его на любой ОС не должно возникнуть проблем.
- С Python доступно огромное количество сервисов, сред разработки, и фреймворков. Легко можно найти подходящий продукт для работы.
- Возможность подключить библиотеки, написанные на C. Это позволяет повысить эффективность, улучшить быстродействие.
- Наличие самых разных источников информации о Python. Не составит труда найти ответ на любой возникший вопрос, так существует много бесплатной литературы, обучающих видеопособий, готовых исходников и шаблонов для работы в открытом доступе.

Преимущества Python относительно других языков

Python легко конкурирует с другими языками программирования, так как имеет множество достоинств. Во-первых, это **понятный и простой язык программирования**. Особенно хорошо он для новичков. Можно создать интересные приложения, и при этом не придется сидеть неделями, изучая сложный синтаксис.

Динамическая типизация – это одно из главных достоинств языка Python. Для новичков это возможность упростить написание кода и избежать множества фатальных ошибок и багов в работе. Также в Python **нет операторных скобок**, с расставлением которых зачастую возникают сложности.

По скорости выполнения программ, когда это касается крупных полномасштабных проектов, Python, конечно же, не лидер. Здесь минусом является и автоматическое управление памятью, и полная динамическая типизация. Python уступает значительно таким языкам как Java, C, C++, но и в то же время с легкостью дает фору JavaScript, Ruby, PHP. Подключение библиотек, написанных на C и возможность предварительной компиляции кода в байт-код – все это позволяет улучшить быстродействие.

В конце хочется отметить, что Python – это язык программирования, востребованный сегодня и с большим потенциалом в будущее. Сегодня рынок труда нуждается в квалифицированных специалистах со знаниями Python.

Новизна

Новизна Программы заключается в том, что Python дает более широкие возможности в области программирования, чем Pascal, который входит в школьный курс информатики. На языке Python

можно легко и быстро создавать простые компьютерные игры, трёхмерные модели и программировать роботов. Этот язык быстрее и легче усваивается, чем Pascal. Многие мировые компании такие, как Intel, Cisco, Hewlett-Packard, используют этот язык при реализации своих проектов. Крупнейшие интернет-ресурсы такие, как Google, YouTube, также разработаны с помощью языка программирования Python.

Особенностью программы

От традиционных машинных Питон отличает целый ряд характеристик. К числу его особенностей можно отнести:

- Полностью автоматическое управление памятью. Данная функция позволяет программистам избежать волнений по поводу необходимости распределять или освобождать память.
- Выполнение операций осуществляется в более высоком уровне абстракций отчасти по причине архитектуры языка, отчасти благодаря расширенной библиотеке кодов, поставляемой вместе с Питон.
- Массив может включать объекты различных типов.
- Значение любого типа может быть назначено переменной.
- Язык легко объединяется с написанными на C и C++ модулями, что позволяет существенно увеличить скорость программ. Благодаря этим особенностям развертка приложений может выполняться очень быстро.

Основным недостатком Python является не полная контролируемость языка, поскольку происходит частичная трансляция к внутренней форме кода байта, выполненного интерпретатором Питон.

Актуальность программы

Данная программа предназначена для реализации в системе дополнительного образования и полностью согласуется со школьным курсом информатики 7 - 11 классов, также данная программа направлена на расширение знаний учащихся в направлении изучения языков программирования, в частности, языка программирования Python. Образовательная программа составлена на основе курса А.А. Чернов «Программирование на языках высокого уровня», программы курса К.Ю. Полякова «Информатика», углубленный уровень, курса «Основы программирования», автор СМ. Окулов, курса М. Лутца «Изучаем Питон», (А.А. Чернов «Программирование на языках высокого уровня». Волгоград: «Учитель», 2012, «Основы программирования». С.М. Окулов и др., М.:

Бином. Лаборатория Знаний, 2012, М. Лутц «Изучаем Питон», СанктПетербург: Символ, 2013г.).

Программа является синтезом основных тем, которые используются при изучении языков программирования, расширяющих и дополняющих знания умения и навыки учащихся 12-16 лет, при работе с программным кодом, решении задач. Программа предназначена для системы дополнительного образования и основана на профильном изучении данной дисциплины.

Цель программы:

- обучение учащихся программированию посредством языка Python, развитие инженерного мышления, воспитание конкурентно способной личности.
- освоение компьютера - современного инструмента познавательной и творческой деятельности.

Задачи:

Обучающие:

- обучить языку программирования Python и созданию программ на его основе;

- научить создавать прикладное программное обеспечение; • расширять кругозор обучающихся в области программирования;

- научить дизайнерскому оформлению созданного ПО.

Развивающие:

- развивать память и внимание, познавательную и творческую активность;
- развивать творческие способности, эстетическое и эргономическое восприятие объектов труда;
- развивать логическое мышление.

Воспитательные:

- прививать интерес к активному творческому самовыражению, культуре труда;
- воспитывать упорство в достижении желаемого результата;
- воспитывать эстетический вкус;
- воспитывать чувство взаимопомощи, доверия, коллективизма.

Данная программа отражает *новый подход к образованию – личностно-ориентированное обучение* через *личностно-образующую деятельность*, в результате которой обучающийся приобретает статус субъекта познания, общения, ценностных ориентаций.

Ожидаемые результаты и способы их проверки:

I Личностные:

- широкие познавательные интересы, инициатива и любознательность, мотивы познания и творчества; готовность и способность учащихся к саморазвитию и реализации творческого потенциала в духовной и предметно-продуктивной деятельности за счет развития их образного, алгоритмического и логического мышления;
- готовность к повышению своего образовательного уровня и продолжению обучения с использованием средств и методов информатики и ИКТ;
- интерес к информатике и ИКТ, стремление использовать полученные знания в процессе обучения другим предметам и в жизни;
- способность связать учебное содержание с собственным жизненным опытом и личными смыслами, понять значимость подготовки в области информатики и ИКТ в условиях развития информационного общества;
- готовность к самостоятельным поступкам и действиям, принятию ответственности за их результаты; готовность к осуществлению индивидуальной и коллективной информационной деятельности;
- способность к избирательному отношению к получаемой информации за счет умений ее анализа и критичного оценивания; ответственное отношение к информации с учетом правовых и этических аспектов ее распространения;
- способность и готовность к принятию ценностей здорового образа жизни за счет знания основных гигиенических, эргономических и технических условий безопасной эксплуатации средств ИКТ.

II Метапредметные: Регулятивные:

- умение организации собственной учебной деятельности, включающими: целеполагание как постановку учебной задачи на основе соотнесения того, что уже известно, и того, что требуется установить;
- планирование – определение последовательности промежуточных целей с учетом конечного результата, разбиение задачи на подзадачи, разработка последовательности и структуры действий, необходимых для достижения цели при помощи фиксированного набора средств;

Познавательные:

- прогнозирование – предвосхищение результата;
- контроль – интерпретация полученного результата, его соотнесение с имеющимися данными с целью установления соответствия или несоответствия (обнаружения ошибки);
- коррекция – внесение необходимых дополнений и корректив в план действий в случае обнаружения ошибки;
- оценка – осознание учащимся того, насколько качественно им решена учебнопознавательная задача;
- владение основными универсальными умениями информационного характера: постановка и формулирование проблемы;
- поиск и выделение необходимой информации, применение методов информационного поиска;
- структурирование и визуализация информации; выбор наиболее эффективных способов решения задач в зависимости от конкретных условий;
 - самостоятельное создание алгоритмов деятельности при решении проблем творческого и поискового характера;

Коммуникативные:

- владение основами продуктивного взаимодействия и сотрудничества со сверстниками и взрослыми: умение правильно, четко и однозначно сформулировать мысль в понятной собеседнику форме;
- умение осуществлять в коллективе совместную информационную деятельность, в частности при выполнении проекта;
- умение выступать перед аудиторией, представляя ей результаты своей работы с помощью средств ИКТ;
 - использование коммуникационных технологий в учебной деятельности и повседневной жизни.

III Предметные результаты:

Обучающиеся научатся:

- использовать термины «информация», «сообщение», «данные», «алгоритм», «программа»; понимание различий между употреблением этих терминов в обыденной речи и в информатике;
- составлять линейные, разветвляющиеся и циклические алгоритмы управления исполнителями на языке программирования python;
 - использовать логические значения, операции и выражения с ними;

Обучающиеся получат возможность научиться:

- владеть понятиями класс, объект, обработка событий;
- формально выполнять алгоритмы, описанные с использованием конструкций ветвления (условные операторы) и повторения (циклы), вспомогательных алгоритмов;
- создавать и выполнять программы для решения несложных алгоритмических задач;
- использовать готовые прикладные компьютерные программы и сервисы;

выбирать способы представления данных в зависимости от поставленной задачи.

Проверка результатов проходит в форме:

- тестирования;
- проведения самостоятельных работ;
- написание индивидуальных программ.

По итогам реализации программы являются:

- написание проекта;
- успешные выступления кружковцев на олимпиадах всех уровней, конкурсах, международной конкурсах;
- участие школьников в вебинарах по избранным вопросам информатики или интернет-олимпиадах и интернет-конкурсах, проводимых интернет-порталом «МетаШкола»;
- участие в школьном (районном) этапе олимпиады по информатике;

В реализации программы принимают участие дети в возрасте от 12-14 лет.

Срок реализации программы - 2 год.

Продолжительность занятия 45 минут, 1 раз в неделю.

Формы организации знаний

В рамках образовательного процесса следует создавать условия для целенаправленного и комфортного воспитания и развития школьников, в этой связи рекомендованная продолжительность учебного занятия - 45 минут. Заниматься развитием творческих способностей учащихся необходимо систематически и целенаправленно через систему занятий, которые должны строиться на междисциплинарной, интегративной основе, способствующей развитию психических свойств личности – памяти, внимания, воображения, мышления. Задачи на занятиях подбираются с учетом рациональной последовательности их предъявления: от репродуктивных, направленных на актуализацию знаний, к частично поисковым, поисковым, исследовательским и проблемным, ориентированным на овладение обобщенными приемами познавательной деятельности. Система занятий должна вести к формированию важных характеристик творческих способностей: беглость мысли, гибкость ума, оригинальность, любознательность, умение выдвигать и разрабатывать гипотезы. Методы и приемы обучения: проблемно-развивающее обучение, знакомство с историческим материалом, иллюстративно-наглядный метод, индивидуальная и дифференцированная работа с учащимися, дидактические игры, диалоговые и дискуссионные технологии, информационные технологии. Кроме того, эффективности организации курса способствует использование различных форм проведения занятий: эвристическая беседа; практикум; интеллектуальная игра; дискуссия; творческая работа. При закреплении материала, совершенствовании знаний, умений и навыков целесообразно практиковать самостоятельную работу школьников. Использование современных образовательных технологий позволяет сочетать все режимы работы: индивидуальный, парный, групповой, коллективный.

Способами проверки степени усвоения материала по каждой теме рекомендуется проводить тематический контроль в форме проверочных самостоятельных работ, тестов, кроссвордов по темам блока занятий, устную олимпиаду и т.п.

Такие проверочные работы должны носить не столько оценивающий, сколько обучающий характер и являться продолжением процесса обучения. Оценки за такие работы можно ставить условно – например, в баллах по числу верно выполненных заданий. Учитывая возраст учащихся, проверочные работы можно проводить в форме игр, викторин, соревнований.

Формы подведения итогов реализации программы:

Основные формы проведения занятий

1. Комбинированное тематическое занятие:

- выступление учителя или кружковца;
- самостоятельное решение задач по избранной теме;
- разбор решения задач (обучение решению задач);
- решение задач занимательного характера; ответы на вопросы учащихся;
- домашнее задание.

2. Конкурсы и соревнования по решению информационных задач, олимпиады, игры, соревнования.

3. Заслушивание рефератов учащихся.

4. Изготовление моделей для уроков информатики.

5. Просмотр презентаций по информатике.

6. Организация учебно-проектной деятельности.

Специфика информатики деятельности такова, что требует системной отработки навыка приобретаемых умений, поэтому поурочные домашние задания в разумных пределах являются обязательными. Домашние задания заключаются не только в повторении темы занятия, решении задач, а также в самостоятельном изучении литературы, рекомендованной учителем.

Теоретические и практические занятия способствуют развитию устной коммуникативной и речевой компетенции учащихся, умениям: вести устный диалог на заданную тему.

методы обучения:

- объяснительно-иллюстративный метод;
- репродуктивный метод;
- метод проблемного обучения;
- метод организации рефлексивной деятельности;
- методы программированного обучения;
- проведение занятий с использованием средств искусства.

Каждое занятие по данной программе отличается своей конкретной целью, логикой и структурой. Применяемые методы обучения обеспечивают взаимосвязь деятельности педагога и обучающегося по обеспечению усвоения учебного содержания в соответствии с задачей: способствовать воспитанию и развитию личности обучающихся, активизировать учебно-познавательную деятельность ребенка.

Материально-техническое обеспечение

Для проведения занятий используется класс математики, выбор оптимальных условий и площадок для проведения различных мероприятий.

Также используется для проведения занятий следующее оборудование и инвентарь:

- выбор оптимальных условий и площадок для проведения различных мероприятий;
- наличие канцелярских принадлежностей;
- аудиоматериалы и видеотехника;
- компьютеры;
- проектор;
- экран и др.

Результаты освоения общеразвивающей программы отслеживаются в процессе:

- овладение системой информатики знаний и умений, необходимых для применения в практической деятельности;
- сформировано умение правильно применять информационную терминологию;
- сформированы качества мышления, характерные для информатики деятельности и необходимые для полноценной жизни в обществе.
- расширен кругозор воспитанников в различных областях элементарной информатики;
- развиты умения делать доступные выводы и обобщения, обосновывать собственные мысли;

- развиты внимание, память, образное и логическое мышление, пространственное воображение;
- выявлены и развиты математические и творческие способности;
- сформирован интерес к языкам программирования;
- расширены коммуникативные способности детей;
- сформированы понятия о культуре труда и совершенствованы трудовые навыки.

Результатом успешной деятельности обучающихся и эффективности данной программы являются творческие достижения обучающихся, а также наличие положительной динамики развития их творческого потенциала. Все результаты заносятся в **накопительное портфолио образовательных результатов и достижений обучающегося.**

Календарный учебный график
реализации дополнительной общеразвивающей программы
«Изучаем PYTHON в 7-8 классах»
на 2021-2022 учебный год

Год обучения	Дата начала занятий	Дата окончания занятий	Количество учебных недель	Количество учебных дней	Количество учебных часов	Режим занятий
1 год	01.09.21	31.05.2022	34	34	34	1 час в неделю

Учебно-тематический план

№ п/п	Тема	Количество часов		
		Всего	Теория	Практика
7 класс				
1.	Язык Python. Среда программирования. Особенности вводавывода	3	1	2
2.	Типы данных, операции. Оператор присваивания	3	1	2
3.	Числа. Стандартные операции	6	2	4
4.	Условный оператор	2	2	-
5.	Цикл while. Теория чисел	10	2	8
6.	Цикл for	10	2	8
	Всего	34	10	24
8 класс				
7.	Литералы строк	3	1	2
8.	Срезы строк	3	1	2
9.	Методы строк	10	3	7
10.	Парадигма и преимущества структурного программирования	2	2	-
11.	Граф и стек вызовов функций. Области видимости переменных	4	2	2
12.	Прямая рекурсия	8	1	7
13.	Косвенная рекурсия	4	2	2

14.	Всего	34	12	22
-----	--------------	----	----	----

Содержание программы

Содержание учебного (тематического) плана

1. Введение в Python

Основные понятия: трансляция, интерпретация, компиляция, синтаксис, семантика, прагматика, переменная, динамическая типизация, служебные слова, идентификаторы, простые типы данных, приоритеты операций, литералы чисел, операция присваивания, PEP 8.

1.1. Язык Python. Среда программирования. Особенности вводавывода

Теория. Язык программирования Python. Достоинства и недостатки. Области применения. Интерактивный режим работы программы.

Практика. Установка языка программирования Python 3.5 и среды программирования WingIDE 100. Регистрация на Интернет-ресурсах.

1.2. Типы данных, операции. Оператор присваивания *Теория.* Ввод и вывод числовой информации.

Практика. Тренировочное задание на ввод и вывод числовой информации.

1.3. Числа. Стандартные операции

Теория. Стандартные операции с целыми и действительными числами. Стиль программирования Python (PEP 8).

Практика. Решение простых задач в интерактивном режиме.

2. Алгоритмические инструкции

Основные понятия: логический тип данных, логические операции (and, or, not, ^A), условный оператор, условное и альтернативное исполнение алгоритма, операторы сравнения, вложенность операторов, оператор цикла, переменная-флаг, генерация псевдослучайной последовательности, инструкции break, continue и pass.

2.1. Условный оператор

Теория. Условная и циклическая инструкции. Каскадность и вложенность алгоритмических инструкций.

2.2. Цикл while *Теория.* Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики.

2.3. Цикл for *Теория.* Теория чисел. Фильтрация потока чисел. НОД и НОК. Проверка числа на простоту. Алгоритм Евклида. Нахождение максимума и минимума.

Практика. Решение задач на анализ чисел потока и целочисленной арифметики повышенной трудности.

3. Строки

Основные понятия: символ, строка, литерал, таблицы кодов ASCII, UTF-8, отладка кода, неизменяемый объект, формат вывода строки, экранированные escape- последовательности, положительная и отрицательная нумерация символов в строке, срез, конкатенация, длина строки.

2.4. Литералы строк

Теория. Понятие «литералы строк».

Практика. Ввод-вывод строки. Решение задач на ввод строки, поиск подстроки.

2.5. Срезы строк

Теория. Форматирование строки.

Практика. Преобразование строки. Решение упражнений.

2.6. Методы строк

Теория. Методы работы со строкой.

Практика. Применение методов строки. Решение задач.

4. Функции

Основные понятия: подпрограмма, функция, процедура, рекурсия, глубина рекурсии, объявление, определение и вызов функции, возврат значений, глобальные и локальные переменные, передача параметров, работа с памятью, граф вызовов, стек вызовов, полиморфизм функций, утиная типизация, lambda-функции.

4.1. Парадигма и преимущества структурного программирования

Теория. Обзор парадигм программирования. Особенности применения языков программирования.

4.2. Граф и стек вызовов функций. Области видимости переменных

Теория. Применение стек и граф вызовов функций.

Практика. Выполнение тренировочных упражнений.

4.3. Прямая рекурсия

Теория. Понятие прямой рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

4.4. Косвенная рекурсия

Теория. Понятие косвенной рекурсии и ее применение.

Практика. Выполнение тренировочных упражнений.

Практические занятия к темам 4.2., 4.3., 4.4.: Нахождение суммы чисел. Числа Фибоначчи.

Вычисление степени. Ханойские башни. Использование библиотеки математических функций. Решение задач повышенной трудности.

5. Списки и кортежи

Основные понятия: список, кортеж, элемент списка и кортежа, индекс, срез списка, матрица, многомерный список, сортировка, сложность алгоритма, устойчивость сортировки, квадратичная, быстрая, синхронная, поразрядная сортировки списка, случайное перемешивание.

5.1. Списки и кортежи в Python. Сходства и различия

Теория. Представление списка и кортежа в памяти компьютера, сходства и различия.

5.2. Операции со списками

Теория. Способы заполнения списка (с клавиатуры, из файла, случайным образом, по формуле).

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка.

5.3. Срезы списков

Теория. Методы работы со списком и кортежем. Методы сортировки списка.

Практика. Решение задач на ввод-вывод элементов одномерного и многомерного списка и кортежа, преобразование, поиск, замену, подсчет.

5.4. Матрицы. Операции над матрицами

Теория. Вычисление сложности алгоритма. Многомерные списки.

Практика. Решение задач повышенной трудности.

6. Словари и множества

Основные понятия: словарь, множество, ключ, кодирование.

6.1. Словари

Теория. Понятие словаря. Способы создания. Словарь, преимущества и недостатки, методы работы со словарем. Словари со смешанными значениями. Кодирование и декодирование текста.

Практика. Решение задач на заполнение, преобразование, поиск, замену, подсчет, вывод элементов словаря.

6.2. Множества

Теория. Понятие множества. Создание множеств. Множество, преимущества и недостатки, методы работы с множеством.

Практика. Решение задач повышенной трудности.

7. Объектно-ориентированное программирование (ООП)

Основные понятия: ООП, класс, метод INIT, экземпляр, наследование, полиморфизм, исключения, виджет, интерфейс, событие, техническое задание, проект, проектная деятельность, виды проектов.

7.1. Классы в Python

Теория. Понятия «класс», «метод INIT», «экземпляр», «наследование», «полиморфизм», «исключения», «виджет», «интерфейс», «интерфейс», «событие».

7.2. Разработка собственного класса

Теория. Принципы разработки собственного класса. Обработка и генерация исключений. Виджет, методы виджета. Графическая библиотека tkinter, класс Tk. Системные методы. События.

Практика. Создание собственного класса.

7.3. Разработка и программирование собственного проекта

Практика. Выбор вида и темы проекта. Составление технического задания. Программирование. Разработка технической документации и презентации проекта.

8. Подведение итогов

8.1. Защита индивидуального проекта

Организационно-педагогические условия реализации программы

Занятия проводятся в следующих формах: лекции, семинары, практические задания.

Дидактический материал, необходимый для проведения занятий:

- краткие конспекты материалов для лекций;

- распечатки заданий для практикумов; - презентационные материалы для объяснения; - карточки

с индивидуальными заданиями.

Техническое оснащение занятий:

- компьютер для демонстрации презентаций;

- проектор;

- рабочие компьютеры учащихся для работы с доступом в

Интернет;

- принтер для распечатки заданий.

Список литературы

1. ФГОС. Примерные программы по информатике для основной и старшей школы. М.: БИНОМ. Лаборатория знаний. 2012.
2. Программа УМК К. Ю. Полякова, М.: Бином, 2012.
3. Информатика. Демонстрационное поурочное планирование. 9-11 классы. Издательство «Учитель». 2007 г.
4. Доусен М. Программируем на Python / М. Доусен - СПб.: Питер, 2016. - 416с.
5. Лутц М. Изучаем Python, 4 издание / М. Лутц - СПб.: СимволПлюс, 2011. - 1280 с.
6. Любанович Б. Простой Python. Современный стиль программирования / Б. Любанович. - СПб.: Питер, 2016. - 480с.
7. Прохоренок Н.А., Дронов В.А. Python 3 и PyQt 5. Разработка приложений / Н.А. Прохоренок, В.А. Дронов - СПб.: «БХВ- Петербург», 2016. - 832с.
8. Саммерфильд М. Python на практике / М. Саммерфильд, пер. А.А. Слинкин – М.: ДМК-Пресс, 2014. - 338с.

Цифровые образовательные ресурсы

1. Россум Г., Дж. Дрейк Ф.Л., Откидач Д.С., Задка М., Левис М., Монтаро С., Реймонд Э.С., Кучлинг А.М., Лембург М.-А., Йи К.-П., Ксиллаг Д., Петрилли Х.Г., Варсав Б.А., Ахлстром Дж.К., Роскинд Дж., Шемениор Н., Мулендер С. Язык программирования Python: [Электронный ресурс]. 2001. URL: <https://goo.gl/8TzY8w>. (Дата обращения: 27.08.2018).

Нормативные документы

1. Федеральный закон Российской Федерации от 29 декабря 2012 г. № 273-ФЗ «Об образовании в Российской Федерации».
2. Приказ Министерства образования и науки РФ от 29 августа 2013 г. № 1008 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам».
3. Приказ Департамента образования города Москвы «О мерах по развитию дополнительного образования детей в 2014-2015 году» от 17.12.2014 г. № 922 (в ред. от 7 августа 2015 года № 1308, от 8 сентября 2015 года № 2074, от 30 августа 2016 года № 1035, от 31 января 2017 года № 30).
4. Письмо Минобрнауки России № 09-3242 от 18.11.2015 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)»)
5. Санитарно-эпидемиологические правила и нормативы СанПиН 2.4.4.3172-14 «Санитарно-эпидемиологические требования к устройству, содержанию и организации режима работы образовательных организаций дополнительного образования детей» (утв. постановлением Главного государственного санитарного врача РФ от 4 июля 2014 г. № 41).

Приложение № 1

1. Запишите значение переменной t , полученное в результате работы следующей программы. Текст программы приведён на трёх языках программирования.

Алгоритмический язык	Бейсик	Паскаль
<pre> алг нач цел t,i t := 2 нц для i от 1 до 3 t := t * i кц вывод t кон </pre>	<pre> DIM i,t AS INTEGER t = 2 FOR i = 1 TO 3 t = t * i NEXT i PRINT t </pre>	<pre> Vart,i: integer; Begin t := 2; For i := 1 to 3 do t :=t * i; Writeln(t); End. </pre>

Ответ:

2. В таблице Dat хранятся данные о количестве краткосрочных командировок, в которые приходилось ездить сотруднику за последний год (Dat[1] — количество командировок в январе, Dat[2] — количество командировок в феврале и т. д.). Определите, что будет напечатано в результате выполнения следующего алгоритма, записанного на трёх алгоритмических языках.

Алгоритмический язык	Бейсик	Паскаль
<pre> алг нач целтаб Dat[1:12] цел k, m month Dat[1] := 5 Dat[2] := 5 Dat[3] := 6 Dat[4] := 8 Dat[5] := 4 Dat[6] := 5 Dat[7] := 4 Dat[8] := 7 Dat[9] := 4 Dat[10] :=4 Dat[11] := 8 Dat[12] := 7 month:=1; m := Dat[1] нц для k от 2 до 12 если Dat[k] <= m то m:= Dat[k]; month := k все кц вывод month кон </pre>	<pre> DIM Dat(12) AS INTEGER Dat[1] = 5 , Dat[2] = 5 Dat[3] = 6 Dat[4] = 8 Dat[5] = 4 Dat[6] = 5 Dat[7] = 4 Dat[8] = 7 Dat[9] := 4 Dat[10] :=4 Dat[11] := 8 Dat[12] := 7 month = 1: m = Dat(1) FORk = 2 TO 12 IF Dat(k) <= m THEN m = Dat(k) month = k END IF NEXT k PRINT month END </pre>	<pre> Var k, m, month: integer; Dat: array[1. .12] of integer; Begin Dat[1] := 5; Dat[2] := 5; Dat[3] := 6; Dat[4] := 8; Dat[5] := 4; Dat[6] := 5; Dat[7] := 4; Dat[8] := 7; Dat[9] := 4; Dat[10] :=4; Dat[11] := 8; Dat[12] := 7; month:= 1; m := Dat[1]; for k := 2 to 12 do if Dat[k] <= m then begin m:= Dat[k]; month := k; end; write(month); End. </pre>

Ответ:

1. У исполнителя Квадратор две команды, которым присвоены номера: 1. возведи в квадрат, 2. прибавь 3.

Первая из них возводит число на экране во вторую степень, вторая — прибавляет к числу 3. Составьте алгоритм получения из числа 5 числа 127, содержащий не более 5 команд. В ответе запишите только номера команд. (Например, 12212 — это алгоритм: возведи в квадрат прибавь 3 прибавь 3 возведи в квадрат прибавь 3, который преобразует число 2 в число 103).

Если таких алгоритмов более одного, то запишите любой из них.

Ответ:

3. Некоторый алгоритм из одной цепочки символов получает новую цепочку следующим образом. Сначала вычисляется длина исходной цепочки символов; если она чётна, то в середину цепочки символов добавляется символ А, а если нечётна, то в начало цепочки добавляется символ Б. В полученной цепочке символов каждая буква заменяется буквой, следующей за ней в русском алфавите (А — на Б, Б — на В и т. д., а Я — на А). Получившаяся таким образом цепочка является результатом работы алгоритма.

Например, если исходной была цепочка **ВРМ**, то результатом работы алгоритма будет цепочка **ВГСН**, а если исходной была цепочка **ПД**, то результатом работы алгоритма будет цепочка **РБЕ**.

Дана цепочка символов **ТОР**. Какая цепочка символов получится, если к данной цепочке применить описанный алгоритм дважды (т. е. применить алгоритм к данной цепочке, а затем к результату вновь применить алгоритм)?

Русский алфавит: АБВГДЕЁЖЗИЙКЛМНОПРСТУ- ФХЦЧШЩЪЫЬЭЮЯ.

Ответ:

4. Напишите программу, которая в последовательности натуральных чисел определяет количество чисел, оканчивающихся на 6. Программа получает на вход количество чисел в последовательности, а затем сами числа. В последовательности всегда имеется число, оканчивающееся на 6. Количество чисел не превышает 1000. Введённые числа не превышают 30

ООО. Программа должна вывести одно число — количество чисел, оканчивающихся на 6.

Пример работы программы:

Входные данные	Выходные данные
3, 16, 26, 24	0, z