



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ ТАТАРСТАН
Государственное автономное профессиональное образовательное учреждение
«Набережночелнинский политехнический колледж»

ПРОГРАММА

УТВЕРЖДАЮ
Директор ГАПОУ
«Набережночелнинский
политехнический колледж»
И.М. Ганиев
от « 18 » 04 2024г



Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«Программирование на языке Python»

Возраст обучающихся 12-16 лет

Срок реализации: 1 год



ЦЕНТР ЦИФРОВОГО ОБРАЗОВАНИЯ ДЕТЕЙ «ИТ-Куб»

СОГЛАСОВАНО

Начальник центра цифрового
образования детей «ИТ-куб»

А.Г. Нургалиева
от «18» 04 2024г

Разработчик:

Слудникова Ирина Валерьевна - педагог дополнительного образования

Возраст обучающихся: 14-17 лет

Сроки реализации: 1 год

Рассмотрено на заседании Педагогического совета ГАПОУ
«Набережночелнинский политехнический колледж» г. Набережные Челны
Протокол № 7 от 18.04. 2024г.

Содержание

1 Цели реализации программы	4
2. Требования к результатам обучения. Планируемые результаты обучения	7
2.1. Формы обучения и виды занятий	7
2.2 Требования к результатам освоения программы	8
3 Содержание программы	9
3.1 Учебный план	10
3.2 Учебная программа	11
4 Календарный учебный график (порядок освоения модулей)	13
5. 15	
5.1 Материально-технические условия реализации программы	15
5.2 Учебно-методическое обеспечение программы	17
5.3 Кадровые условия реализации программы	18
6. Оценка качества освоения программы	18
7. Составители программы	18

**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
(ОБЩЕРАЗВИВАЮЩАЯ) ПРОГРАММА
технической направленности
«Программирование на языке Python»**

»

1 Цели реализации программы

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

В обязательном школьном курсе информатики программирование нередко представлено лишь на элементарном уровне, на это выделяется недостаточное количество часов. Лишь немногие школы могут себе позволить преподавать программирование на достойном уровне. Следствием этого является формальное восприятие учащимися основ современного программирования и неумение применять полученные знания на практике.

Опираясь на уникальный опыт преподавания программирования в Школе анализа данных Яндекса (АНО ДПО «ШАД») и на факультете компьютерных наук НИУ ВШЭ, была подготовлена данная программа. В ней большое внимание уделяется практической работе на компьютере, самостоятельному написанию кода.

Изучение основных принципов программирования невозможно без регулярной практики написания программ на каком-либо языке. Для обучения был выбран язык Python. Данный выбор обусловлен тем, что синтаксис языка достаточно прост и интуитивно понятен, а это понижает порог вхождения и позволяет сосредоточиться на логических и алгоритмических аспектах программирования, а не на выучивании тонкостей синтаксиса. 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.

Научившись программировать на языке Python, обучающиеся получают мощный и удобный инструмент для решения как учебных, так и прикладных задач. Вместе с тем чистота и ясность его конструкций позволит учащимся потом с лёгкостью выучить любой другой язык программирования.

Знания и умения, приобретённые в результате освоения курса, могут быть использованы обучающимися при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, биологии, лингвистике и другим наукам, а также они являются фундаментом для дальнейшего совершенствования мастерства программирования.

Направленность программы:

Программа «Основы программирования на языке Python» имеет техническую направленность, в её 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.

Актуальность программы:

В рамках изучения программы обучающиеся постоянно будут сталкиваться с необходимостью самостоятельной работы над заданиями: школьники учатся решать задачи без помощи преподавателя. Для этого в содержании курса фигурируют задания, в которых для решения задачи необходимо найти какую-то информацию в сети Интернет; может

потребуется устранение ошибки, которую не так просто обнаружить; условие сформулировано недостаточно прозрачно и ученику необходимо самостоятельно формализовать его (или задать правильные вопросы преподавателю). Все эти знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят школьников к самостоятельной проектно-исследовательской деятельности с применением современных технологий.

Новизна программы:

Новизна данной образовательной программы заключается в том, что программа курса предполагает знакомство с фундаментальными понятиями алгоритмизации и программирования на доступном уровне; имеет практическую направленность с ориентацией на реальные потребности, соответствующие возрасту ученика; охватывает как алгоритмическое направление, так и вопросы практического использования полученных знаний при решении задач из различных областей знаний; ориентирована на существующий парк вычислительной техники и дополнительные ограничения; допускает возможность варьирования в зависимости от уровня подготовки и интеллектуального уровня учащихся (как группового, так и индивидуального), а также предусматривает возможность индивидуальной работы с учащимися.

Практическая значимость курса заключается в том, что он способствует более успешному овладению знаниями и умениями по направлению «Программирование» через развитие самостоятельности обучающихся и оптимизацию средств и методов обучения.

Элементы программы курса могут быть рекомендованы для использования учителями информатики при проведении лабораторно-практических и практических занятий.

Педагогическая целесообразность программы:

Программа «Основы программирования на языке Python» является целостной и непрерывной в течение всего процесса обучения, и позволяет учащемуся шаг за шагом раскрывать в себе творческие возможности и самореализоваться в современном мире.

Углубленный уровень предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка, гарантированно обеспечивают трансляцию общей и целостной картины в рамках содержательно-тематического направления программы.

Осваивая данную программу, обучающиеся будут овладевать навыками востребованных уже в ближайшие десятилетия специальностей, многие из которых включены в Атлас профессий будущего. Практически для каждой перспективной профессии будут полезны знания и навыки, рассматриваемые в программе.

Содержание программы учитывает возрастные и психологические особенности детей 16–17 лет, которые определяют выбор форм проведения занятий с обучающимися. Дети этого возраста отличаются внутренней уравновешенностью, стремлением к активной практической деятельности, поэтому основной формой проведения занятий выбраны практические занятия. Ребятам также увлекает совместная, коллективная деятельность, так как резко возрастает значение коллектива, общественного мнения, отношений со сверстниками, оценки поступков и действий ребёнка со стороны не только старших, но и сверстников. Ребёнок стремится завоевать в их глазах авторитет, занять достойное место в коллективе. Поэтому в программу включены практические занятия соревновательного характера, которые позволяют каждому проявить себя и найти своё место в детском коллективе.

Также следует отметить, что дети данной возрастной группы характеризуются такими психическими процессами, как изменение структуры личности и возникновение интереса к ней, развитие абстрактных форм мышления, становление более осознанного и целенаправленного характера деятельности, проявление стремления к самостоятельности и независимости, формирование самооценки. Эти процессы позволяют положить начало созданию начального профессионального самоопределения обучающихся.

Отличительная особенность программы:

Дополнительная общеразвивающая программа «Основы программирования на языке Python» в отличие от других подобных программ включает изучение языка программирования Python, начиная со стартового уровня. Также включает раздел изучения алгоритмов и знакомит обучающихся с азами программирования.

Программа состоит из трех модулей: «Введение в программирование», «Базовые конструкции языка Python», «Решение прикладных задач в Python» и организована по принципу дифференциации по уровням сложности. Модуль – структурная единица образовательной программы, имеющая логическую завершенность по отношению к результатам обучения (Словарь рабочих терминов по предпрофильной подготовке). Каждый модуль состоит из кейсов (не менее двух), направленных на формирование определенных компетенций (hard и soft). Результатом каждого кейса является «продукт» (групповой, индивидуальный), демонстрирующий сформированность компетенций.

Кейс – история, описывающая реальную ситуацию, которая требует проведения анализа, выработки и принятия обоснованных решений (Высшая школа экономики). Кейс включает набор специально разработанных учебно-методических материалов. Кейсовые «продукты» могут быть самостоятельным проектом по результатам освоения модуля или общего проекта по результатам всей образовательной программы.

Модули и кейсы различаются по сложности и реализуются по принципу «от простого к сложному». По содержанию модули делятся на предметные, непосредственно связанные с областью знаний, и общеразвивающие, направленные на формирование познавательных и коммуникативных компетенций.

Модуль 1 позволяет обеспечить начальную подготовку детей в области программирования и формирует положительную мотивацию к языкам программирования. Предполагает использование и реализацию общедоступных и универсальных форм организации учебного материала, минимальную сложность предлагаемого для освоения содержания программы.

Модуль 2, Модуль 3 – предполагает освоение специализированных знаний в языке программирования Python.

Для возрастной категории 12–16 лет при решении кейсов ставятся задания повышенного уровня сложности и применяется оборудование соответствующей возрастной категории.

Цель программы:

Способствовать формированию творческой личности, обладающей информационными компетенциями, владеющей базовыми понятиями теории алгоритмов, умеющей разрабатывать эффективные алгоритмы и реализовывать их в виде программы, написанной на языке программирования Python. Подготовка к использованию методов программирования на Python в учебной и последующей профессиональной деятельности, расширение кругозора обучающихся в различных предметных областях.

Задачи программы:

Образовательные:

- изучение конструкций языка программирования Python;
- знакомство с принципами и методами функционального и объектно-ориентированного программирования; основными структурами данных и типовыми методами обработки этих структур;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- формирование навыков разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

Развивающие:

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- приобретение навыков поиска информации в сети Интернет, анализ выбранной информации на соответствие запросу, использование информации при решении задач;
- развитие у обучающихся интереса к программированию, самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- формирование и развитие навыков работы с различными источниками информации, необходимой для решения учебных задач; умения планировать свои действия с учётом фактора времени, в обстановке с элементами конкуренции, предвидеть результат и достигать его, при необходимости вносить коррективы в первоначальный замысел.

Воспитательные:

- воспитание этики групповой работы;
- воспитание отношений делового сотрудничества, взаимоуважения;
- развитие основ коммуникативных отношений внутри проектных групп и в коллективе в целом;
- воспитание упорства в достижении результата.

2. Требования к результатам обучения. Планируемые результаты обучения

2.1. Формы обучения и виды занятий

Основной тип занятий – комбинированный, сочетающий в себе элементы теории и практики. Большинство заданий курса выполняется самостоятельно с помощью персонального компьютера и необходимых программных средств.

Единицей учебного процесса является блок уроков (модуль). Каждый такой блок охватывает отдельную информационную технологию или её часть. Внутри блоков разбивка по времени изучения производится учителем самостоятельно, но с учётом рекомендованного календарно-тематического плана. С учётом регулярного повторения ранее изученных тем темп изучения отдельных разделов блока определяется субъективными и объективными факторами.

Каждая тема курса начинается с постановки задачи – характеристики предметной области или конкретной программы на языке Python, которую предстоит изучить. С этой целью учитель проводит демонстрацию презентации или показывает саму программу, а также готовые работы, выполненные в ней. Закрепление знаний проводится с помощью практики отработки умений самостоятельно решать поставленные задачи, соответствующие минимальному уровню планируемых результатов обучения.

Основные задания являются обязательными для выполнения всеми обучающимися в классе. Задания выполняются на компьютере с использованием интегрированной среды разработки. При этом обучающиеся не только формируют новые теоретические и практические знания, но и приобретают новые технологические навыки.

Методика обучения ориентирована на индивидуальный подход. Для того чтобы каждый ученик получил наилучший результат обучения, программой предусмотрены индивидуальные задания для самостоятельной работы на домашнем компьютере. Такая форма организации обучения стимулирует интерес ученика к предмету, активность и самостоятельность учащихся, способствует объективному контролю глубины и широты знаний, повышению качества усвоения материала обучающимися, позволяет педагогу получить объективную оценку выбранной им тактики и стратегии работы, методики индивидуального обучения и обучения в группе, выбора предметного содержания.

Для самостоятельной работы используются разные по уровню сложности задания, которые носят репродуктивный и творческий характер. Количество таких заданий в работе может варьироваться.

В ходе обучения проводится промежуточное тестирование по темам для определения уровня знаний учеников. Выполнение контрольных заданий способствует активизации учебно-познавательной деятельности и ведёт к закреплению знаний, а также служит индикатором успешности образовательного процесса.

Срок освоения общеразвивающей программы определяется содержанием программы и составляет 1 год.

Режим занятий, объём общеразвивающей программы: Занятия проводятся 2 раза в неделю по 2 урока. Продолжительность занятия - 40 минут. После 40 минут занятий организовывается перерыв длительностью 10 минут для проветривания помещения и отдыха учащихся.

2.2 Требования к результатам освоения программы

Предметные результаты:

– умение определять результат выполнения алгоритма при заданных исходных данных, узнавать изученные алгоритмы обработки чисел и числовых последовательностей, создавать на их основе несложные программы анализа данных, читать и понимать несложные программы, написанные на выбранном для изучения универсальном алгоритмическом языке высокого уровня;

– понимание основных предметных понятий («информация», «алгоритм», «исполнитель», «модель») и их свойств;

– развитие логических способностей и алгоритмического мышления, умения составить и записать алгоритм для конкретного исполнителя, знакомство с основными алгоритмическими структурами – линейной, условной и циклической;

– умение выполнять пошагово (с использованием компьютера или вручную) несложные алгоритмы управления исполнителями и анализа числовых и текстовых данных;

– навыки и опыт разработки программ в выбранной среде программирования, включая тестирование и отладку программ; умение использовать основные управляющие конструкции объектно-ориентированного программирования и библиотеки прикладных программ, выполнять созданные программы;

– умение разрабатывать и использовать компьютерно-математические модели, оценивать числовые параметры моделируемых объектов и процессов, интерпретировать результаты, получаемые в ходе моделирования реальных процессов, анализировать готовые модели на предмет соответствия реальному объекту или процессу;

– умение соблюдать нормы информационной этики и права.

Личностные результаты:

– формирование ответственного отношения к учению, способности довести до конца начатое дело аналогично завершённым творческим учебным проектам;

– формирование способности к саморазвитию и самообразованию средствами информационных технологий на основе приобретённой благодаря иллюстрированной среде программирования мотивации к обучению и познанию;

– развитие опыта участия в социально значимых проектах, повышение уровня самооценки благодаря реализованным проектам;

– формирование коммуникативной компетентности в общении и сотрудничестве со сверстниками в процессе образовательной, учебно-исследовательской и проектной деятельности, участия в конкурсах и конференциях различного уровня;

– формирование целостного мировоззрения, соответствующего современному уровню развития информационных технологий;

– формирование осознанного позитивного отношения к другому человеку, его мнению, результату его деятельности;

– формирование ценности здорового и безопасного образа жизни; усвоение правил индивидуального и коллективного безопасного поведения при работе с компьютерной техникой.

Метапредметные результаты:

- умение самостоятельно ставить и формулировать для себя новые задачи, развивать мотивы своей познавательной деятельности;
- умение самостоятельно планировать пути решения поставленной проблемы для получения эффективного результата; понимание, что в программировании длинная программа не всегда лучшая;
- умение критически оценивать правильность решения учебно-исследовательской задачи;
- умение корректировать свои действия, вносить изменения в программу и отлаживать её в соответствии с изменяющимися условиями;
- владение основами самоконтроля, способность к принятию решений;
- умение создавать, применять и преобразовывать знаки и символы, модели и схемы для решения учебно-исследовательских и проектных работ;
- формирование и развитие компетентности в области использования информационно-коммуникационных технологий (ИКТ-компетенция);
- умение организовывать учебное сотрудничество и совместную деятельность с учителем и сверстниками в процессе проектной и учебно-исследовательской деятельности.

3 Содержание программы

Категория слушателей: учащиеся в возрасте от 12 до 16 лет.

Трудоемкость обучения: 144 академических часа.

Форма обучения: очная.

3.1 Учебный план

№ п/п	Название раздела, темы	Количество часов				Формы аттестации / контроля
		Всего	Теория	Практика	Резервное время	
Углубленный уровень						
Модуль 1.						
	Введение в программирование	21	8	11	2	
1.1	Понятия кода, интерпретатора, программы	2	1	1	-	Знакомство. Опрос.
1.2	Знакомство с языком Python и средой разработки. Исполнение кода и отладка	2	1	1	-	Опрос, беседа, выполнение мини-проекта
1.3	Базовые типы данных. Переменные	4	2	2	-	Опрос, беседа, выполнение мини-проекта
1.4	Именованное. Работа со строками	3	1	2	-	Опрос, беседа, выполнение мини-проекта
1.5	Арифметические операции.	2	1	1	-	Опрос, беседа, выполнение

№ п/п	Название раздела, темы	Количество часов				Формы аттестации / контроля
		Всего	Теория	Практика	Резервное время	
	Выражение в определениях					мини-проекта
1.6	Простые встроенные функции	3	1	2	-	Опрос, беседа, выполнение мини-проекта
1.7	Ветвление. Условные операторы	3	1	2	-	Опрос, беседа, выполнение мини-проекта
	Резервное время	2	-	-	2	
Модуль 2. Базовые конструкции в Python		38	15	19	4	
2.1	Знакомство с циклом While	5	2	3	-	Опрос, беседа, выполнение мини-проекта
2.2	Знакомство с циклом For	5	2	3	-	Опрос, беседа, выполнение мини-проекта
2.3	True, False, Break, Continue. Вложенные циклы	4	2	2	-	Опрос, беседа, выполнение мини-проекта
2.4	Срезы, списки. Индексация	5	2	3	-	Опрос, беседа, выполнение мини-проекта
2.5	Методы split и join. Списочные выражения	4	2	2	-	Опрос, беседа, выполнение мини-проекта
2.6	Методы списков и строк	4	2	2	-	Практическое задание
2.7	Вложенные списки	3	1	2	-	Практическое задание
2.8	Знакомство со словарями	4	2	2	-	Практическое задание
	Резервное время	4	-	-	4	
Модуль 3. Работа с функциями. Введение в объектно-ориентированное программирование		47	22	21	4	
3.1	Функции, функциональный стиль программирования	12	6	6	-	Опрос, беседа, выполнение мини-проекта
3.2	Лямбда функции. Модуль functools	5	2	3	-	Практическое задание

№ п/п	Название раздела, темы	Количество часов				Формы аттестации / контроля
		Всего	Теория	Практика	Резервное время	
3.3	Модуль Collections	4	2	2	-	Практическое задание
3.4	Обработка исключений	4	2	2	-	Практическое задание
3.5	Принципы устройства и механика создания модулей и пакетов	6	4	2	-	Беседа, выполнение мини-проекта
3.6	ООП: инкапсуляция, наследование, полиморфизм	8	4	4	-	Практическое задание
3.7	Итераторы и генераторы	4	2	2	-	Практическое задание
	Резервное время	4	-	-	4	
Модуль 4. Разработка проектов		38	14	22	2	
4.1	Разработка проектов на основе возрастной категории	32	12	20	-	Беседа, практическое задание
4.2	Защита проектов	4	2	2	-	Практическое задание
	Резервное время	2	-	-	2	
Итого		144	59	73	12	

3.2 Учебная программа

Структура программы основана на модульном принципе.

Модуль 1. Введение в программирование

Тема 1.1 Понятие кода, интерпретатора, программы.

Теория: Введение в программу «Программирование на Python»

Основные понятия программирования: исполнитель, система команд, алгоритм, программа, среда разработки. Понятия кода, интерпретатора, программы, интегрированные среды, исполнение кода.

Практика: Проверка полученных знаний.

Тема 1.2 Знакомство с языком Python и средой разработки. Исполнение кода и отладка

Теория: Общие сведения о языке Python. Установка Python на компьютер. Режимы работы Python. Что такое программа. Простейшие программы с выводом на экран. Структура программы на языке Python. Комментарии.

Практика: Установка среды разработки, решение задач.

Тема 1.3 Базовые типы данных. Переменные

Теория: Основные типы данных, понятие переменной. Простейшие программы с использованием ввод-вывода. Технология разработки программы, ключевые слова

Практика: Решение задач с использованием памяти. Составление линейных алгоритмов.

Тема 1.4 Именованное. Работа со строками

Теория: Методы и правила именования переменных, знакомство со строками.

Практика: Решение задач со строками.

Тема 1.5 Арифметические операции. Выражение в определениях

Теория: Знакомство с основными арифметическими операциями. Упрощение сложных вычислений с помощью переменных. Операции. Порядок выполнения операций. Математические функции. Композиция. Задачи на элементарные действия с числами.

Практика: Решение задач на элементарные действия с числами.

Тема 1.6 Простые встроенные функции

Теория: Знакомство с самыми часто используемыми встроенными функциями Python

Практика: Решение задач с использованием встроенных функций.

Тема 1.7 Ветвление. Условные операторы

Теория: Логический тип данных. Логические выражения и операторы. Сложные условные выражения (логические операции and, or, not). Условный оператор. Альтернативное выполнение. Примеры решения задач с условным оператором. Множественное ветвление. Реализация ветвления в языке Python.

Практика: Логические выражения. Условные операторы. Множественное ветвление. Составление условных алгоритмов и программ с ветвлением.

Модуль 2. Базовые конструкции в Python

Тема 2.1 Знакомство с циклом While

Теория: Понятие цикла. Тело цикла. Условия выполнения тела цикла. Оператор цикла с условием. Оператор цикла while. Бесконечные циклы. Альтернативная ветка цикла while. Обновление переменной. Краткая форма записи обновления. Примеры использования циклов. "Числа Фибоначчи".

Практика: Решение задач. Составление циклических алгоритмов.

Тема 2.2 Знакомство с циклом For

Теория: Оператор цикла с параметром for. Операторы управления циклом. Пример задачи с использованием цикла for. Вложенные циклы. Циклы в циклах. Случайные числа. Функция randrange. Функция random. Примеры решения задач с циклом.

Практика: Решение задач.

Тема 2.3 True, False, Break, Continue. Вложенные циклы

Теория: Ключевые слова для работы с вложенными циклами.

Практика: Решение задач.

Тема 2.4 Срезы, списки. Индексация

Теория: Списки. Тип список (list). Индексы. Индексация строк. Обход списка. Проверка вхождения в список. Добавление в список. Суммирование или изменение списка. Операторы для списков. Срезы списков. Удаление списка. Клонирование списков. Списочные параметры. Функция range. Кортежи. Присваивание кортежей. Кортежи как возвращаемые значения.

Практика: Решение задач.

Тема 2.5 Методы split и join. Списочные выражения

Теория: Строковые методы split() и join() в Python. Списочные выражения. Создание списков без явного использования циклов и вызова списочного метода append().

Практика: Решение задач.

Тема 2.6 Методы списков и строк

Теория: Синтаксис методов для работы со списками и строками.

Практика: Решение задач.

Тема 2.7 Вложенные списки

Теория: Матрицы второго порядка.

Практика: Решение задач.

Тема 2.8 Знакомство со словарями

Теория: Словари и множества. Хэш-таблицы.

Практика: Решение задач.

Модуль 3. Решение прикладных задач в Python

Тема 3.1 Функции, функциональный стиль программирования

Теория: Функции, функциональная парадигма программирования., создание функций. Параметры и аргументы. Локальные и глобальные переменные. Поток выполнения. Функции, возвращающие результат. Анонимные функции. Примеры решения задач с использованием функций. Рекурсивные функции. Вычисление факториала.

Практика: Решение задач.

Тема 3.2 Лямбда функции. Модуль functools

Теория: Инструкция lambda. Работа с модулем functools.

Практика: Решение задач.

Тема 3.3 Модуль Collections

Теория: Реализация специализированных контейнеров типов данных с помощью модуля.

Практика: Решение задач.

Тема 3.4 Обработка исключений

Теория: Случаи возникновения ошибок, методы их обработки. Конструкция try - except - else - finally.

Практика: Решение задач.

Тема 3.5 Принципы устройства и механика создания модулей и пакетов

Теория: Понятие модуля и пакета, размещение и импорт модуля.

Практика: Решение задач.

Тема 3.6 Основы объектно-ориентированного анализа и дизайна, шаблоны проектирования, инкапсуляция, наследование, полиморфизм

Теория: Шаблоны проектирования. Причины появления, принципы и основные сущности объектно-ориентированного подхода к разработке ПО. Инкапсуляция, полиморфизм, наследование, композиция. Восходящий и нисходящий способы разработки. Методика объектно-ориентированного дизайна приложения. Проектирование и разработка классов.

Практика: Решение задач.

Тема 3.7 Итераторы и генераторы

Теория: Понятие и реализация итераторов и генераторов. Предназначение, особенности устройства и работы, типовые сферы применения.

Практика: Решение задач.

Модуль 4. Решение прикладных задач в Python

Тема 4.1 Разработка проектов на основе возрастной категории

Теория: Знакомство и работа с фреймворками и библиотеками для написания финального проекта.

Практика: Прототипирование и написание проекта.

Тема 4.2 Защита проектов

Практика: Защита реализованного проекта.

4 Календарный учебный график (порядок освоения модулей)

График разработан в соответствии с Санитарными правилами СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи», Положением о деятельности центра цифрового образования «IT-куб» ГАПОУ «Набережночелнинский политехнический колледж».

График учитывает возрастные психофизические особенности учащихся и отвечает требованиям охраны их жизни и здоровья.

Содержание Графика включает в себя следующее:

- продолжительность учебного года;

- количество учебных групп по годам обучения и направленностям;
- регламент образовательного процесса;
- продолжительность занятий;
- аттестация учащихся;
- режим работы учреждения;
- периодичность проведения родительских собраний.

Государственное автономное профессиональное образовательное учреждение «Набережночелнинский политехнический колледж» в установленном законодательством Российской Федерации порядке несет ответственность за реализацию в полном объеме дополнительных общеобразовательных (общеразвивающих) программ в соответствии с календарным учебным графиком.

1. Продолжительность учебного года в Центре:

Начало учебного года – 01.09.2024 года

Окончание учебного года – 31.05.2025 года.

Начало учебных занятий:

1 год обучения – не позднее 05.09.2024 года;

Комплектование групп 1 года обучения – с 01 по 03.09.2024 года.

Продолжительность учебного года – 36 недель.

2. Количество учебных групп по годам обучения и направленностям:

Направленность программы	1 год обучения	2 год обучения
техническая	1	-
Итого:	1	-

3. Регламент образовательного процесса:

1 год обучения – 4 часа в неделю (144 часа в год);

Занятия организованы в Центре в отдельных группах.

4. Продолжительность занятий.

Занятия проводятся по расписанию, утвержденному начальником центра цифрового образования IT-cube ГАПОУ «Набережночелнинский политехнический колледж» в свободное от занятий в общеобразовательных учреждениях время, включая учебные занятия в субботу и воскресенье с учетом пожеланий родителей (законных представителей) несовершеннолетних учащихся с целью создания наиболее благоприятного режима занятий и отдыха детей.

Занятия начинаются не ранее 8.30 часов утра и заканчиваются не позднее 20.00 часов.

Продолжительность занятия - 80 минут.

После 40 минут занятий организовывается перерыв длительностью 10 минут для проветривания помещения и отдыха учащихся.

5. Аттестация учащихся: промежуточная (итоговая) – декабрь, май.

5. Организационно-педагогические условия реализации программы

5.1 Материально-технические условия реализации программы

№ п/п	Наименование оборудования	Технические характеристики	Модель	Ед. изм.	Кол-во
1. Компьютерное оборудование					
1.1	Системный блок для преподавателя и обучающихся	<p>Модель процессора: Core i5 7400</p> <p>Количество ядер процессора: 4</p> <p>Частота процессора: 3500 МГц</p> <p>Модель дискретной видеокарты: Intel HD Graphics 630</p> <p>Объем видеопамати: 512 МБ</p> <p>Размер оперативной памяти: 8 ГБ</p> <p>Объем твердотельного накопителя (SSD): 240 ГБ</p> <p>В комплекте клавиатура: В комплекте мышь: В комплекте коврик для мыши и клавиатуры</p>	<p>HP Pavilion TG01-0027ur, Microsoft Wired Keyboard 600, Defender GM-090L</p>	шт	13
1.2	Монитор	<p>Диагональ экрана: 23,8"</p> <p>Максимальное разрешение: 1920x1080</p>	BenQ BL2405PT	шт	26
1.3	Многофункциональное устройство	лазерное ч/б А4	HP LaserJet 135a	шт	1
2. Презентационное оборудование					
2.1	моноблочное интерактивное устройство	<p>Диагональ экрана: 65"</p> <p>Разрешение экрана: 3840x2160</p> <p>Процессор Intel® Core™ i5</p>	TEACHTOUCH 3.5 65", UHD, ПК CORE I5	шт	1
2.2	напольная стойка для интерактивных досок	совместимость с моноблочным интерактивным устройством	0	шт	1
3. Мебель					
3.1	Стол преподавателя письменный с тумбами	<p>с боковой панелью: ширина 1600, приставка 1000</p> <p>В комплекте 1 тумба, экран крепится к ножкам стола</p>		шт	1
3.2	Стул преподавателя	материал спинки: сетчатый функция регулировки и фиксации наклона		шт	1
3.3	Стол ученический	2-местный, ширина 1200,		шт	12

№ п/п	Наименование оборудования	Технические характеристики	Модель	Ед. изм.	Кол-во
		экран крепится к ножкам стола			
3.4	Стул ученический	эргономичной формы с зонами антискольжения		шт	12
3.5	Шкаф платяной комбинированный	800x1900x400		шт	1
3.6	шкаф для документов	92x85x38,5 см, закрывается на ключ, для хранения ценного оборудования		шт	1

5.2 Учебно-методическое обеспечение программы

Методические материалы

Образовательный процесс осуществляется в очной форме.

В образовательном процессе используются следующие **методы**:

1. словесные (беседа, опрос, дискуссия и т.д.);
2. игровые;
3. метод проблемного изложения (постановка проблемы и решение её самостоятельно или группой);
4. метод проектов;
5. наглядные:
 - демонстрация плакатов, схем, таблиц, диаграмм;
 - использование технических средств;
 - просмотр кино- и телепрограмм;
6. практические:
 - практические задания;
 - анализ и решение проблемных ситуаций и т.д.

Методическое обеспечение

Методические пособия, разработанные преподавателем с учётом 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.

Электронное приложение к учебникам К.Ю. Полякова Информатика и ИКТ. Набор цифровых образовательных ресурсов – дидактические материалы, интерактивные тесты, анимационные плакаты.

5.3 Кадровые условия реализации программы

Количество педагогов дополнительного образования, привлеченных для реализации программы 3 чел.

Данные ПДО, привлеченных для реализации программы

№ п/п	ФИО	Должность, наименование организации
1	Слудникова Ирина Валерьевна	Педагог дополнительного образования ЦЦОД «IT-куб»

2	Елькин Сергей Иванович	Педагог дополнительного образования ЦЦОД «IT-куб»
3	Слудникова Ирина Валерьевна	Педагог дополнительного образования ЦЦОД «IT-куб»

5.4 Оценка качества освоения программы

Система контроля знаний и умений обучающихся представляется в виде учета результатов по итогам выполнения заданий итогового контроля и посредством наблюдения, отслеживания динамики развития обучающегося.

Итоговая аттестация обучающихся осуществляется по 50-балльной шкале, которая переводится в форму, согласно таблице:

Баллы, набранные обучающимся	Уровень освоения
0-20 баллов	Не зачтено
21-50 баллов	Зачтено

Критерии оценивания заданий итогового контроля:

Максимальный балл	Критерии оценки
20 баллов	Качество выполненного задания
20 баллов	Самостоятельность выполнения
10 баллов	Оригинальность задания

Формы проведения итогового контроля соответствуют разделам дополнительной общеобразовательной общеразвивающей программы, целям и задачам Центра цифрового образования детей «IT-Куб».

Список литературы:

1. Сысоева М. В., Сысоев И. В. Программирование для «нормальных» с нуля на языке Python: Учебник. В двух частях. Часть 1 / Сысоева М. В., Сысоев И. В. – Москва: Базальт СПОМАКС Пресс, 2018. – 180с.

2. Мэтиз. Изучаем Python. Программирование игр, визуализация данных, веб-приложения: Пособие / Мэтиз Эрик. - СПб.: Питер, 2017. - 496с.: ил. — (Серия «Библиотека программиста»).

3. Гуриков, С. Р. Основы алгоритмизации и программирования на Python: учеб. пособие / С.Р. Гуриков. — Москва: ФОРУМ: ИНФРА-М, 2020. — 343с. — (Среднее профессиональное образование).

4. Жуков, Р. А. Язык программирования Python: практикум: учебное пособие / Р.А. Жуков. — Москва: ИНФРА-М, 2020. — 216с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование).

Электронные ресурсы:

1. Материалы и презентации к урокам в LMS Яндекс.Лицея.
2. Сайт pythonworld.ru – «Python 3 для начинающих».
3. Сайт pythontutor.ru – «Питонтьютор».
4. <https://www.youtube.com/playlist?list=PLJOzdkh8T5kpIBTG9mM2wVBjh-5OpdwBl> – Лекции А.В. Умнова, прочитанные в Школе анализа данных Яндекса.