

Центр цифрового образования детей «IT- куб»
(структурное подразделение АНО ДТ «Красноярский Кванториум»)

РЕКОМЕНДОВАНО
методическим советом

Протокол № 1
от « 14 » мая 2024 г.

УТВЕРЖДАЮ
Генеральный директор
АНО ДТ «Красноярский
Кванториум»

Кениг С.Р.

Приказ № 177
от « 20 » мая 2024 г.



Дополнительная общеобразовательная общеразвивающая программа
технической направленности

«Программирование на языке Python»

Срок реализации:

1 год

Возраст детей:

11-17 лет

Составитель программы:

Артемьев И.В.

г. Красноярск, 2024 г.

1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Данная программа разработана в соответствии с нормативными правовыми актами в области образования:

Федеральным законом от 29 декабря 2012 № 273-ФЗ «Об образовании в Российской Федерации»;

Указом Президента РФ от 07.05.2018 № 204 «О национальных целях и стратегических задачах развития Российской Федерации на период до 2024 года»;

Распоряжением Правительства РФ от 29.05.2015 № 996-р «Стратегия развития воспитания в Российской Федерации до 2025 года, утвержденная»;

Распоряжением Правительства РФ от 31.03.2022 № 678-р «Концепция развития дополнительного образования детей до 2030»;

Постановлением Правительства Российской Федерации от 26.12.2017 № 1642 «Об утверждении государственной программы Российской Федерации «Развитие образования»;

Приказом Министерства просвещения РФ от 27.07.2022 №629 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;

Постановлением Главного государственного санитарного врача Российской Федерации от 28.09.2020 № 28 «Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;

Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы) (Приложение к письму Министерства образования и науки Российской Федерации от 18.11.2015 г. № 09-3242);

Методическими рекомендациями по реализации дополнительных общеобразовательных программ с применением электронного обучения и дистанционных образовательных технологий (Приложение к письму Министерства просвещения РФ от 31.01.2022 г. № 1ДГ 245/06);

Методическими рекомендациями «Об использовании государственных символов Российской Федерации при обучении и воспитании детей и молодежи в образовательных организациях, а также организациях отдыха детей и их оздоровления» (Письмо Министерства просвещения РФ от 15.04.2022 № СК-295/06);

Методическими рекомендациями по созданию и функционированию центров цифрового образования «IT-куб» (Письмо Министерства просвещения РФ от 10.11.2021 №ТВ-1984/04.

Дополнительная общеобразовательная общеразвивающая программа «Программирование на языке Python» (далее - программа) имеет техническую направленность, базовый уровень сложности и ориентирована

на обучающихся 11-17 лет. Программа в объеме 144 часа рассчитана на один год из расчета 4 часа в неделю.

1.1. Актуальность программы

В настоящее время компьютеры окружают современного человека повсюду, и не многие понимают то, как работают сервисы и программы, которыми мы пользуемся ежедневно. Программирование — это не просто написание программ для компьютера, это целое искусство воплощения идеи в свою программу. Знание основ программирования может помочь не только начинающим программистам продолжить свой путь в этой перспективной области, но и тем, чья жизнь в дальнейшем не будет тесно связана с написанием программ. Имея эти знания человек в дальнейшем, будет способен автоматизировать решение рутинных задач и бытовых проблем, с которыми сталкивается в жизни.

Знания и умения, полученные в результате освоения курса, могут быть использованы при сдаче ЕГЭ, при участии в олимпиадах по программированию, при решении задач по физике, химии, и другим наукам, а также они являются фундаментом для дальнейшего изучения других языков программирования.

1.2. Отличительные особенности

Программа авторская, разработана на основе методического пособия «Реализация дополнительной общеобразовательной программы по тематическому направлению «Программирование на языке Python» с использованием оборудования центра цифрового образования детей «IT-куб»», г. Москва, 2021 г.

Отличительной особенностью данной образовательной программы является то, что процесс обучения выстроен с учетом используемых современных образовательных технологий. Программа предполагает использование и реализацию таких форм организации материала, которые допускают освоение специализированных знаний и языка программирования, обеспечивают трансляцию общей и целостной картины в рамках содержательно-тематического направления программ. Итоговое задание позволяет проявить творческий подход к решению задачи.

1.3. Адресат общеразвивающей программы

Набор на программу осуществляется в соответствии с Порядком организации образовательной деятельности по дополнительным общеобразовательным общеразвивающим программам автономной

некоммерческой организации «Красноярский детский технопарк «Кванториум».

Дополнительная общеобразовательная общеразвивающая программа «Программирование на Python» предназначена для детей в возрасте 11–17 лет, мотивированных к обучению и обладающих системным мышлением.

В связи с ориентированностью программы на практическую индивидуальную (групповую) работу максимальное количество обучающихся в группе не должно превышать 8 человек.

Образовательный процесс в разновозрастных учебных группах выстраивается на идеях педагогики сотрудничества: учение без принуждения, самоанализа, создания благоприятного интеллектуального фона учебной группы, личностного подхода, продвижения в индивидуальном темпе, самоконтроля и взаимоконтроля. Реализация положений педагогики сотрудничества эффективно воплощается в жизнь при применении диалогических форм обучения, которые подразумевают творческое отношение и обмен креативной деятельностью. Осуществление педагогического диалога в учебном процессе позволяет в ходе учебно-познавательной деятельности детей развивать их коллективистские связи.

Возрастные особенности группы:

11-14 лет – подростковый возраст. Характерная особенность - индивидуальное самосознание, сознательное выражение индивидуальности. Основное стремление - самоутверждение. В подростковом возрасте интересы детей стабилизируются. Основное новообразование - формирование взрослости как желания жить во взрослом обществе. На социально-нравственном уровне - потребность иметь и отстаивать собственные мнения и оценки; на интеллектуально-энергетическом уровне - потребность овладеть элементами саморазвития и понять сферу интересов подростка; на культурном уровне - потребность отразить взрослость во внешности и поведении. Основными критериями являются следующие. Роль преподавателей высших учебных заведений, работающих с подростками, заключается в проведении регулярной подготовки к введению их в социально значимые группы.

15-17 лет – подростковый возраст. Завершение физической и психической зрелости. Социальная готовность к общественно полезному производительному труду и гражданской ответственности. В отличие от подросткового возраста, где индивидуальность проявляется через самоопределение - "кто я", подростковая индивидуальность выражается через самовыражение - "как я влияю". Основная задача педагога дополнительного образования, занимающегося с детьми 15-17 лет, - подготовить их к полноценной социальной жизни и разрешить противоречие между подготовкой к полноценной социальной жизни и недопущением задержек в содержании и организации учебной деятельности.

Основной формой обучения являются практические занятия, так как дети этой возрастной группы обладают внутренней уравновешенностью и

готовностью к активному участию в практической деятельности. Они также заинтересованы в общественной и групповой деятельности, так как возрастает значение групп, обмена опытом, взаимоотношений со сверстниками и оценки их действий, слов и поступков сверстниками и старшими. Дети стремятся завоевать авторитет в собственных глазах и занять ценное положение в группе. Поэтому в программе предусмотрены практические упражнения в соревновании личностей, которые позволяют каждому проявить себя и найти свое место в группе.

Следует также отметить, что для детей в этом возрасте характерны такие познавательные процессы, как изменение структуры личности и возникновение интересов к ней, развитие абстрактных форм мышления, формирование более осознанной и целенаправленной деятельности, проявление стремления к независимости и самостоятельности, формирование самооценки. Эти процессы формируют раннее профессиональное самоопределение учащегося.

1.4. Педагогическая целесообразность

Выбор своего первого языка программирования — это очень сложная задача и поэтому курс построен таким образом, чтобы помочь учащимся заняться программированием и понять в каком направлении им хотелось бы развиваться дальше. Изучая программирование, обучающиеся получают глубокое понимание принципов работы компьютера, организации ввода, вывода и хранения информации, принципов построения диалоговых приложений, познают азы профессии программиста.

В рамках образовательной программы «Программирование на языке Python» обучающийся познакомится со средой программирования PyCharm, затем он познакомится с одним из самых универсальных языков программирования — Python.

1.5. Срок реализации программы и объем учебных часов

Программа рассчитана на 1 год обучения. Годовая нагрузка на обучающегося составляет 144 часа.

1.6. Формы обучения, виды занятий и режим занятий

Учебные занятия проходят в очной форме. Режим занятий — 2 раза в неделю по 2 академических часа (1 академический час - 40 минут) с обязательным перерывом, что определяется Санитарно-эпидемиологическими правилами и нормативами СанПиН 2.4.3648-20.

При проведении занятий используются комбинированные занятия — изложение нового материала, проверка пройденного материала, закрепление полученных знаний, самостоятельная работа, занятия по решению кейсов.

При проведении занятий используются три 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.

Повторение и усвоение пройденного материала осуществляется через прикладную работу обучающегося, использующего на практике приобретенные знания.

1.7. Цель и задачи программы

Целью программы «Программирование на языке Python» является изучение основ программирования на языке Python, рассмотрение различных парадигм программирования, предлагаемых этим языком (процедурная, функциональная, объектно-ориентированная); подготовка к использованию как языка программирования, так и методов программирования на Python в учебной и последующей профессиональной деятельности в различных предметных областях.

Задачи:

- сформировать и развить навыки алгоритмического и логического мышления, грамотной разработки программ;
- познакомить с принципами и методами функционального программирования;
- познакомить с принципами и методами объектно-ориентированного программирования;
- сформировать навыки работы в интегрированной среде разработки на языке Python;
- сформировать знания о конструкции языка программирования Python;
- познакомить с основными структурами данных и типовыми методами обработки этих структур;
- сформировать навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;
- сформировать навыки поиска информации в сети интернет, анализа выбранной информации на соответствие запросу, использования информации при решении задач;
- развить у обучающихся интерес к программированию;
- сформировать навыки самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;
- расширить кругозор обучающихся в области программирования.

1.8. Планируемые результаты освоения программы

По результатам обучения, обучающиеся овладевают основами программирования на языке Python.

Личностные результаты:

- сформировано умение самостоятельной деятельности;
- сформировано умение работать в команде;
- сформированы навыки анализа и самоанализа.

Метапредметные результаты:

-сформировано умение самостоятельно планировать пути достижения целей, соотносить свои действия с планируемыми результатами, осуществлять контроль своей деятельности, определять способы действий в рамках предложенных условий, корректировать свои действия в соответствии с изменяющейся ситуацией; оценивать правильность выполнения учебной задачи;

-сформировано умение самоконтроля, самооценки, принятия решений и осуществления осознанного выбора в учебной и познавательной деятельности;

-сформированы навыки самостоятельности и творческого подхода к решению задач с использованием средств вычислительной техники;

-сформировано умение выступления с компьютерным сопровождением.

Предметные результаты:

-сформированы знания основ современных языков программирования, знания о принципах и методах функционального программирования, принципах и методах объектно-ориентированного программирования;

-развиты навыки алгоритмического и логического мышления, грамотной разработки программ;

-расширен кругозор обучающихся в области программирования.

-развит интерес к программированию;

-сформировано умение объяснять и использовать на практике как простые, так и сложные структуры данных, конструкции для работы с ними и типовые методы обработки этих структур;

-сформированы навыки работы в интегрированной среде разработки на языке и знания о конструкции языка программирования Python;

-сформированы навыки разработки эффективных алгоритмов и программ на основе изучения языка программирования Python;

-сформировано умение анализа, поиска и обработки как своего, так и чужого кода;

-сформировано умение работы с информацией: нахождение, оценки и использования информации из различных источников, необходимой для решения профессиональных задач (в том числе на основе системного подхода).

1.9. Формы подведения итогов обучения

Итоговая аттестация проводится в конце года и представляет собой защиту итогового индивидуального задания с использованием знаний, полученных за курс. Ознакомиться с примером тем для итогового кейса можно в Приложении 1. С критериями оценки индивидуального проекта можно ознакомиться в Приложении 2.

Итоговая аттестация учащихся осуществляется по 100-балльной шкале, которая переводится в один из уровней освоения образовательной программы согласно таблице:

Уровень освоения	Баллы
Низкий	0 – 49
Средний	50 – 75
Высокий	76 – 100

2. УЧЕБНО-ТЕМАТИЧЕСКИЙ ПЛАН

№	Наименование разделов	Всего часов	Теория	Практика	Форма контроля
1	Введение в программирование	18	7	11	
1.1	Знакомство со средой	2	1	1	—
1.2	Условный оператор	2	1	1	Выполнение контрольного задания
1.3	Простые встроенные функции	2	1	1	—
1.4	Знакомство с циклом while	4	1	3	Выполнение контрольного задания
1.5	Отладчик	2	1	1	—
1.6	Знакомство с циклом for	4	1	3	Выполнение контрольного задания
1.7	True и False, break и continue	2	1	1	—
2	Базовые конструкции в Python	30	10	20	
2.1	Вложенные циклы	4	1	3	—

2.2	Множества	2	1	1	—
2.3	Строки. Индексация	2	1	1	—
2.4	Строки. Срезы	2	1	1	Выполнение контрольного задания
2.5	Знакомство со списками	2	1	1	—
2.6	Кортежи. Преобразование коллекций	2	1	1	—
2.7	Методы split и join. Списочные выражения	4	1	3	—
2.8	Методы списков и строк	4	1	3	Выполнение контрольного задания
2.9	Вложенные списки	4	1	3	—
2.10	Знакомство со словарями	4	1	3	—
3	Решение прикладных задач в Python	38	19	19	
3.1	Функции	4	2	2	—
3.2	Возвращение значений из функции	2	1	1	—
3.3	Области видимости переменных функции	2	1	1	—
3.4	Передача параметров функции	2	1	1	—
3.5	Функция с переменным числом аргументов	2	1	1	—
3.6	Функции как объект. Лямбда-функции	4	2	2	—
3.7	Рекурсия	4	2	2	Выполнение контрольного задания
3.8	Обработка коллекций. Поточковый ввод sys.stdin	4	2	2	—
3.9	Библиотеки Python. Встроенные модули	2	1	1	—
3.10	Библиотеки Python. Работа с графическими файлами	2	1	1	—
3.11	Библиотеки Python. Работа с графическими файлами и звуком	2	1	1	—
3.12	Библиотеки Python. Морфология	2	1	1	—
3.13	Библиотеки Python. Работа с	2	1	1	—

	документами				
3.14	Библиотеки Python. Numpy	4	2	2	—
4	Объектно-ориентированное программирование	28	10	18	
4.1	Введение в ООП	4	2	2	—
4.2	Полиморфизм	4	2	2	—
4.3	Определение операторов	2	1	1	—
4.4	Наследование	4	2	2	—
4.5	Проектирование и разработка классов	4	0	4	—
4.6	Графический интерфейс пользователя с библиотекой tkinter	6	2	4	—
4.7	Работа с графикой с библиотекой tkinter	4	1	3	—
5	Выполнение индивидуального проекта	30	0	30	
5.1	Выполнение индивидуального проекта	28	0	28	—
5.2	Итоговая аттестация	2	0	2	Защита проекта
Итого		144	46	98	

3. СОДЕРЖАНИЕ ПРОГРАММЫ

1. Введение в программирование

1.1. Тема: Знакомство со средой

Теоретическая работа: программирование и программа, виды языков программирования.

Практическая работа: знакомство с виртуальной средой взаимодействия: регистрация, организация личного кабинета, поиск и выкладывание материалов. Знакомство с системой автоматизированной проверки задач.

1.2. Тема: Условный оператор

Теоретическая работа: конструкция ветвления. Разветвляющиеся программы. Условные выражения. Операции отношения. Логические операции. Полная и сокращенная форма условного оператора. Блоки и составные операторы. Оператор множественного выбора.

Практическая работа: работа с логическим типом данных. Объявление и использование в программе переменных, констант и выражений логического типа. Использование простых и составных условий. Полная и сокращенная

формы условного оператора. Создание разветвляющейся программы с помощью условных операторов. Оператор множественного выбора. Организация множественного разветвления в программе.

1.3. Тема: Простые встроенные функции

Теоретическая работа: простые типы данных: целый, вещественный, символьный, логический и их представление в ПК. Приведение типов. Работа с переменными и константами. Использование математических функций. Организация ввода и вывода данных.

Практическая работа: работа с простыми типами данных: целым, вещественным, символьным и логическим. Организация ввода и вывода данных.

1.4. Тема: Знакомство с циклом while

Теоретическая работа: циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком, Циклы с предусловием,

Практическая работа: повтор команд. Переменные цикла. Операторы while.

1.5. Тема: Отладчик

Теоретическая работа: отладка программы. Шаг вперед и назад. Вход в функции.

Практическая работа: разбор ошибок в программе. Первые шаги к тестированию.

1.6. Тема: Знакомство с циклом for

Теоретическая работа: синтаксис, реализация и способы применения всех видов циклов.

Практическая работа: циклы. Повтор команд. Переменные цикла. Оператор for.

1.7. Тема: True и False, break и continue

Теоретическая работа: операторы принудительного завершения итерации цикла. Оператор завершения цикла. Оператор безусловного перехода. Синтаксис, реализация и правила использования.

Практическая работа: бесконечный цикл. Выход из цикла.

2. Базовые конструкции в Python

2.1. Тема: Вложенные циклы

Теоретическая работа: циклы. Циклические программы. Понятия итерации, счетчика и параметров цикла. Циклов со счетчиком. Циклы с предусловием. Циклы с постусловием. Синтаксис, реализация и способы применения всех видов циклов. Вложенные циклы. Операторы принудительного завершения итерации цикла. Оператор завершения цикла. Оператор безусловного перехода. Синтаксис, реализация и правила использования.

Практическая работа: циклы. Повтор команд. Переменные цикла. Вложенные циклы. Операторы while, for. Бесконечный цикл. Выход из цикла.

2.2. Тема: Множества

Теоретическая работа: основы множеств и алгебры логики. Круги Эйлера.

Практическая работа: создание простейших множеств и основные

встроенные функции. Как объединять множества и делать выборку.

2.3. Тема: Строки. Индексация.

Теоретическая работа: новые типы данных. Для чего они нужны и как с ними работать.

Практическая работа: работа типами данных: Строки, списки, кортежи и словари. Организация ввода и вывода данных.

2.4. Тема: Строки. Срезы

Теоретическая работа: новые типы данных, срезы. Обратная индексация.

Практическая работа: работа типами данных: Строки, срезы. Организация ввода и вывода данных.

2.5. Тема: Знакомство со списками

Теоретическая работа: списки. Индексация и обратная индексация, схожесть со строками.

Практическая работа: работа типами данных: Списки. Организация ввода и вывода данных.

2.6. Тема: Кортежи. Преобразование коллекций

Теоретическая работа: отличия кортежей от списков. Практическое использование списков.

Практическая работа: работа с типами данных. Кортежи, tuple(). Организация ввода и вывода данных.

2.7. Тема: Методы split и join. Списочные выражения

Теоретическая работа: преобразование данных.

Практическая работа: работа со списком, преобразование данных.

2.8. Тема: Методы списков и строк

Теоретическая работа: методы и функции для разных типов данных. Различия между изменяемыми и неизменяемыми типами данных.

Практическая работа: работа со списком, преобразование данных. Сортировка, поиск максимального и минимального элемента. Создание выборок.

2.9. Тема: Вложенные списки

Теоретическая работа: создание структуры в структуре.

Практическая работа: создание списка в списке, кортежа в списке. Правильный вызов и вывод.

2.10. Тема: Знакомство со словарями

Теоретическая работа: отличия словарей от списков, кортежей и множеств.

Практическая работа: создание словарей, использование словарей для создания базы данных.

3. Решение прикладных задач в Python

3.1. Тема: Функции

Теоретическая работа: подходы к программированию, деление программы на функции.

Практическая работа: создание функций. Аргументы. Правильный вывод и ввод данных.

3.2. Тема: Возвращение значений из функции

Теоретическая работа: вывод из функции. Аргументы.

Практическая работа: функция return. Присваивание функций.

3.3. Тема: Области видимости переменных функции

Теоретическая работа: отличия глобальных и локальных данных.

Практическая работа: создание глобальных и локальных переменных и структур данных. Вызов глобальных переменных и ограничения локальных.

3.4. Тема: Передача параметров функции

Теоретическая работа: ввод данных через аргументы и их отработка.

Практическая работа: работа с данным как локальными, так и глобальными. Передача данных между функций.

3.5. Тема: Функция с переменным числом аргументов

Теоретическая работа: создание функций с переменными аргументами.

Практическая работа: разница между статичными и переменными функциями.

3.6. Тема: Функции как объект. Лямбда-функции

Теоретическая работа: лямбда-функции.

Практическая работа: создание и использование функции. Работа с некорректным вызовом.

3.7. Тема: Рекурсия

Теоретическая работа: создание рекурсий. Принцип бесконечности.

Практическая работа: определение условий окончания рекурсий. Контроль бесконечности.

3.8. Тема: Обработка коллекций. Поточковый ввод sys.stdin

Теоретическая работа: потоковый ввод-вывод, классификация, общие подходы и методы, конвертация

Практическая работа: обобщение типов данных. Функция Counter(). Общие подходы к работе с любой коллекцией

3.9. Тема: Библиотеки Python. Встроенные модули

Теоретическая работа: виды библиотек.

Практическая работа: установка и подключение библиотек. Вызов функций из модулей и подмодулей.

3.10. Тема: Библиотеки Python. Работа с графическими файлами

Теоретическая работа: особенности работы с графическими файлами.

Практическая работа: библиотеки по работе с данными и графическими файлами. Обработка потокового изображения.

3.11. Тема: Библиотеки Python. Работа с графическими файлами и звуком

Теоретическая работа: обработка звуковых файлов. Особенности и сложности.

Практическая работа: работа с библиотеками подключения звука. Создание будильника.

3.12. Тема: Библиотеки Python. Морфология

Теоретическая работа: работа с текстовыми данными. Семантика и парсинг

документов.

Практическая работа: считывание данных из файлов и запись. Редактирование и парсинг.

3.13. Тема: Библиотеки Python. Работа с документами

Теоретическая работа: работа с текстовыми данными. Семантика и парсинг документов.

Практическая работа: считывание данных из файлов и запись. Редактирование и парсинг.

3.14. Тема: Библиотеки Python. Numpy

Теоретическая работа: анализ данных и обработка больших данных.

Практическая работа: обработка больших данных. Особенности библиотеки NumPy. Разница между списками и структурами в NumPy.

4. Объектно-ориентированное программирование

4.1. Тема: Введение в ООП.

Теоретическая работа: понятия объекта и класса. Поля и методы класса. Иерархия классов. Принципы инкапсуляции, наследования и полиморфизма. Преимущества и недостатки объектно-ориентированного подхода. Этапы объектно-ориентированного программирования.

Практическая работа: создание классов, объектов, полей и методов.

4.2. Тема: Полиморфизм

Теоретическая работа: первый принцип ООП.

Практическая работа: изучение синтаксиса описания класса, его компонентов и спецификаторов доступа к ним. Изучение правил описания и вызова конструкторов, деструкторов и других методов класса. Объявление объектов и работа с ними. Создание программы, выполняющей работу с классами и объектами.

4.3. Тема: Определение операторов

Теоретическая работа: свертка. Сильные стороны сверточных нейронных сетей в задачах компьютерного зрения.

Практическая работа: нейросетевая библиотека Keras. Классификация изображения.

4.4. Тема: Наследование

Теоретическая работа: второй принцип ООП.

Практическая работа: изучение синтаксиса описания класса, его компонентов и спецификаторов доступа к ним. Изучение правил описания и вызова конструкторов, деструкторов и других методов класса. Объявление объектов и работа с ними. Создание программы, выполняющей работу с классами и объектами.

4.5. Тема: Проектирование и разработка классов

Практическая работа: изучение синтаксиса описания класса, его компонентов и спецификаторов доступа к ним. Изучение правил описания и вызова конструкторов, деструкторов и других методов класса. Объявление объектов и работа с ними. Создание программы, выполняющей работу с

классами и объектами

4.6. Тема: Графический интерфейс пользователя с библиотекой tkinter

Теоретическая работа: понятия GUI. Модуль Tkinter и библиотека PyQt.

Практическая работа: создание своего первого графического интерфейса. Создание виджета Label. Настройка размера и шрифта текста. Настройка размеров окна приложения. Добавление виджета Button.

4.7. Тема: Работа с графикой с библиотекой tkinter

Теоретическая работа: модуль Tkinter и библиотека PyQt. Основы графики.

Практическая работа: создание крестиков-ноликов. Отрисовка изображений.

5. Выполнение индивидуального кейса

5.1 Выполнение индивидуального кейса

Практическая работа: выполнение индивидуального проекта.

5.2 Итоговая аттестация.

Практическая работа: защита решения индивидуального проекта.

4. СПИСОК ЛИТЕРАТУРЫ

1. Босова, Л.Л. Занимательные задачи по информатике [Электронный ресурс] / Л.Л. Босова, А.Ю. Босова, Ю.Г. Коломенская. – 5-е изд. – М.: БИНОМ. Лаборатория знаний, 2013.
2. Гаско, Рик. Простой Python просто с нуля [Электронный ресурс] / Рик Гаско. – М.: СОЛОН-Пресс, 2019.
3. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Д.М. Златопольский. – М.: БИНОМ. Лаборатория знаний, 2007.
4. Златопольский, Д. М. Сборник задач по программированию [Электронный ресурс] / Д.М. Златопольский. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011.
5. Меньшиков, Ф.В. Олимпиадные задачи по программированию [Электронный ресурс] / Ф.В. Меньшиков. – СПб.: Питер, 2006.
6. Мюллер, Джон Пол. Python для чайников [Электронный ресурс] / Джон Пол Мюллер. – 2-е изд.: Пер с англ. – СПб.: ООО «Диалектика», 2019.
7. Окулов, С.М. Основы программирования: учебное пособие [Электронный ресурс] / С.М. Окулов. – 10-е изд., электрон. – Москва : Лаборатория знаний, 2020.
8. Прохоренок, Н.А. Python 3. Самое необходимое [Электронный ресурс] / Н. А. Прохоренок, В. А. Дронов. – 2-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2019.
9. Пупышев, В.В. 128 задач по началам программирования [Электронный ресурс] / В.В. Пупышев. – М.: БИНОМ. Лаборатория знаний, 2009.
10. Пэйн, Брайсон. Python для детей и родителей [Электронный ресурс] / Брайсон Пейн [пер. с англ. М.А. Райтмана]. – М.: Издательство «Э», 2017.
11. Свейгарт, Эл. Учим Python, делая крутые игры [Электронный ресурс] / Эл. Свейгарт. – М: Эксмо, 2018.
12. Седер, Наоми. Python. Экспресс-курс [Электронный ресурс] / Наоми Седер. – 3-е изд. – СПб.: Питер, 2019.
13. Столяров, А.В. Оформление программного кода [Электронный ресурс] / А.В. Столяров. – 2-е изд., испр. и доп. – Москва: МАКС Пресс, 2019.
14. Шень, А. Программирование: теоремы и задачи [Электронный ресурс] / А. Шень. – 6-е изд., дополненное. М.: МЦНМО, 2017.
15. Шуман, Х.Г. Python для детей [Электронный ресурс] / Х.Г. Шуман [пер. с нем. М.А. Райтман]. – М.: ДМК Пресс, 2019.

5. ИНФОРМАЦИОННО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ ПРОГРАММЫ

5.1. Материально-техническое обеспечение программы

-Общее компьютерное оборудование:

1. Ноутбук преподавателя Lenovo ThinkBook 15 G2 15/6 /мышь/ тип1 – в количестве 2 шт.;
2. Ноутбук ученика Lenovo ThinkBook 15 G2 15/6 /мышь/ тип1- в количестве 8 шт.;
3. Проводная компьютерная мышь /ОКЛИК 325M/ - в количестве 10 шт.;
4. Наушники Panasonic RP-НТ161 -в количестве 10 шт.;
5. Монитор Dell E2421 тип 2 тип1 – в количестве 1 шт.;
6. Принтер лазерный HP LaserJet Enterprise M406dn – в количестве 1 шт.;

-Общее презентационное оборудование:

7. Интерактивная панель 65" 4К Lumien MP6502ELRU - в количестве 1 шт.;
8. Стойка СМ-1200У - в количестве 1 шт.;
9. Флипчарт 100*70 магнитно-маркерный Attache - в количестве 1 шт.

5.2. Информационное обеспечение программы

Книги:

- «Изучаем Python», Марк Лутц — книга крайне подробно рассматривает общие концепции языка, всё написано понятно и доступно.
- «Высокопроизводительный Python: практическое пособие для людей», Миша Горелик, Ян Освальд — в книге рассказывается о многопоточности, асинхронном и мультипроцессорном исполнении, о том, как организовать кластерные вычисления и как потреблять меньше оперативной памяти.

Telegram каналы:

- Python etc. Здесь вы найдете практические советы по написанию кода на Python (с отрывками-примерами). Канал англоязычный.
- PythonistPro. Англоязычный канал по Python. Тут вас ждут ссылки на статьи, видео и книги, а также посты с приемами

программирования. Подпишитесь на канал — и будете не только Python учить, но и английский подтягивать.

- Python Academy. Очень популярный канал (почти 49 тысяч человек) с советами по использованию Python. Каждый пост — короткий рассказ о каком-нибудь модуле или функции, или приеме. У канала есть свой чат, но есть и обсуждение в комментариях под постами.

Youtube каналы:

- Хауди Хо™ – Просто о мире IT! – уроки по программированию, в том числе по
- Гоша Дударь – разные тематические видео на IT тематику: Python в
- Иван Викторович – мини-курс Python с нуля и разработка игр на Pygame.

Темы для итогового проекта

Итоговым результатом освоения обучающимися полученных в процессе обучения навыков и компетенций в рамках представленной программы является итоговая защита проекта. Так как от выбора темы проекта зависит качество проделанной самостоятельной работы, а также итоговая защита проекта, зачастую у обучающихся возникает проблема выбора темы итогового проекта.

Поэтому необходимо помочь обучающимся найти все пути, ведущие к достижению цели. В приложении приведены примеры тем, которые может выполнить ученик. В зависимости от знаний и интересов ученика темы могут меняться при согласовании с преподавателем. Работа над персональным проектом — предполагает построение цели, задач для ее достижения, и выполнение по персональному ТЗ.

Предлагаемые темы работ:

- Чат-бот с использованием библиотеки Aiogram;
- 2D игра с использованием библиотеки PyGame;
- Настольное (Desktop) приложение (PyGame, Tkinter и др.);
- Мобильное приложение (PyQt, KivyMD и др.).

Критерии оценки итогового проекта

№ п/п	Критерии	Баллы
1	Новизна и актуальность темы проекта	от 1 до 10
2	Внешний вид разработанного проекта	от 1 до 10
3	Работоспособность разработанного проекта	от 1 до 10
4	Качество выполнения разработанного проекта	от 1 до 10
5	Форматирование и качество кода, наличие комментариев	от 1 до 10
6	Формулировка темы, цели и задач проекта	от 1 до 10
7	Соответствие результата проекта поставленной цели	от 1 до 10
8	Дальнейшее развитие проекта	от 1 до 10
9	Оценка защиты проекта	от 1 до 10
10	Качество презентации и презентационных материалов	от 1 до 10

Контрольные задания

УСЛОВИЯ

1. Пользователь вводит 3 целых числа. Найти сумму этих чисел.

Пример

Ввод:

4

3

5

Вывод:

12

2. Пользователь вводит два числа ширину и длину прямоугольника. Найти площадь этого прямоугольника.

Пример

Ввод:

4

3.5

Вывод:

14

3. Пользователь вводит трёхзначное число. Найдите сумму его цифр.

Пример

Ввод:

241

Вывод:

7

4. Пользователь вводит три целых числа — часы, минуты и секунды, прошедшие с начала суток. Найти сколько времени в секундах прошло с начала суток.

Пример

Ввод:

0

2

7

Вывод:

127

5. Пользователь вводит три целых числа. Два из них равны друг другу. Найти третье число, не равное остальным. Если среди введенных чисел не оказалось двух равных друг другу или все числа равны, выведите строку "Ошибка".

Пример

Ввод:

4

3

4

Вывод:

3

6. Пользователь вводит целое положительное число. Проверьте является ли это число трехзначным, если является, то выведите строку "YES", иначе "NO".

Пример

Ввод:

454

Вывод:

YES

7. Пользователь вводит номер текущего месяца. Программа должна вывести текущее время года.

Пример

Ввод:

12

Вывод:

Зима

8. Пользователь вводит 3 числа. Найти и вывести наименьшее из чисел (не использовать функцию min).

Пример

Ввод:

4

3

6

Вывод:

3

ЦИКЛ WHILE

1. Пользователь вводит число N. Выведите на экран все числа от 1 до N в одну строку через пробел.

Пример

Ввод:

5

Вывод:

1 2 3 4 5

2. По данному числу N выведите на экран все числа от N до 1 в порядке убывания.

Пример

Ввод:

5

Вывод:

5 4 3 2 1

3. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран "СТОП!".

Пример

Ввод:

5

4

0

Вывод:

СТОП!

4. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран сумму всех чётных чисел (0 не считается частью последовательности).

Пример

Ввод:

5

4

6

0

Вывод:

10

5. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран максимальное число кратное трём (0 не считается частью последовательности).

Пример

Ввод:

5

4

6

9

0

Вывод:

9

6. Пользователь последовательно вводит числа. Если пользователь введёт 0, программа должна закончить свою работу и вывести на экран минимальное положительное число (0 не считается частью последовательности).

Пример

Ввод:

5

-4

8

3

-1

0

Вывод:

3

ЦИКЛ FOR

1. Пользователь вводит 2 целых числа А и В. Вывести все числа от А до В.

Пример

Ввод:

3

5

Вывод:

3 4 5

Пример 2

Ввод:

7

4

Вывод:

7 6 5 4

2. Пользователь вводит 2 целых числа А и В. Первое число всегда меньше второго. Найти сумму чисел от А до В.

Пример

Ввод:

3

5

Вывод:

12

3. Пользователь вводит число N, затем N целых чисел. Найти количество отрицательных среди введенных чисел.

Пример

Ввод:

5

-1

6

-4

7

-3

Вывод:

3

4. Пользователь вводит число N , затем N целых чисел. Найти сумму всех положительных чисел.

Пример

Ввод:

3

-1

4

7

Вывод:

11

5. Пользователь вводит число N , затем N положительных целых чисел. Найти сумму всех трёхзначных чисел.

Пример

Ввод:

4

12

433

7122

100

Вывод:

533

6. Пользователь вводит число N , затем N целых чисел. Найти минимальное из чисел.

Пример

Ввод:

3

1

4

7

Вывод:

1

СТРОКИ

1. Пользователь вводит строку. Показать номера символов, совпадающих с последним символом строки.

Пример

Ввод:

Мама папа

Вывод:

1 3 6

2. Пользователь вводит строку. Определите общее количество символов «+», идущих после «0».

Пример

Ввод:

20+140+1+2

Вывод:

2

3. Пользователь вводит строку. Вставить в строке после каждого символа «a» символ «b».

Пример

Ввод:

Abcda acegeds

Вывод:

Abcdab abcegeds

4. Пользователь вводит строку. Найти сумму имеющихся в строке цифр.

Пример

Ввод:

Реши пример: 20+140+1+2

Вывод:

10

5. Пользователь вводит строку. Удалить в строке все цифры.

Пример

Ввод:

У нас было 2 пакетика сахара, 15 пирожков и 9 яблок.

Вывод:

У нас было пакетика сахара, пирожков и яблок.

6. Пользователь вводит строку. Удалить из строки все цифры 9, остальные цифры увеличить на 1.

Пример

Ввод:

У нас было 2 пакетика сахара, 15 пирожков и 9 яблок.

Вывод:

У нас было 3 пакетика сахара, 26 пирожков и 10 яблок.

СПИСКИ

1. Создать список из 8 случайных чисел в диапазоне от -10 до 10 включительно. Найти максимально чётное число.

Пример

Список:

4 8 -3 9 5 -4 -2 1

Вывод:

8

2. Пользователь вводит строку из чисел, указанных через пробел. Найти сумму чисел, кратных 5.

Пример

Ввод:

4 6 3 5 10

Вывод:

15

3. Пользователь вводит число N, затем N целых чисел. Найти количество чисел, заканчивающихся на 4.

Пример

Ввод:

4

34

11

44

73

Вывод:

2

4. Программа получает на вход последовательность, каждое число записано в отдельной строке. Последовательность завершается числом 0, при считывании которого программа должна закончить свою работу. Найти минимальное трёхзначное число.

Пример

Ввод:

64

455

3421

435

0

Вывод:

435

5. Создать список из 8 случайных чисел в диапазоне от -10 до 10 включительно. Удалить из списка все отрицательные элементы.

Пример

Ввод:

-4 6 3 -5 2 7 10 -3

Вывод:

6 3 2 7 10

6. Создать список из 8 случайных чисел в диапазоне от -10 до 10 включительно. Заменить элементы, которые больше обоих своих соседей, на максимальный элемент списка (крайние элементы не учитывать).

Пример

Ввод:

-4 6 3 -5 2 7 10 -3

Вывод:

-4 10 3 -5 2 7 10 -3

ФУНКЦИИ

1. Напишите функцию $F1(n, a, b)$, которая будет возвращать список из n случайных чисел в диапазоне от a до b (используйте эту функцию при решении других задач).
2. Напишите функцию $F2(s)$, которая будет возвращать среднее значение всех чётных чисел из списка s . Если в списке нет чётных чисел, вернуть «error».
3. Напишите функцию $F3(s)$, которая будет удалять из списка все числа, кратные 5. Функция должна вернуть изменённый список и вывести на экран, количество удалённых элементов.
4. Напишите функцию $F4(s, a, b)$, для удаления из списка всех элементов в диапазоне от a до b . Функция должна вернуть изменённый список.
5. Напишите функцию $F5(n)$ для вычисления факториала числа. Если пользователь введёт некорректное значение, вернуть -1.

6. Напишите функцию $F6(s1, s2)$, которая получает на вход два списка и возвращает новый список с элементами, которые содержатся в обоих списках.

