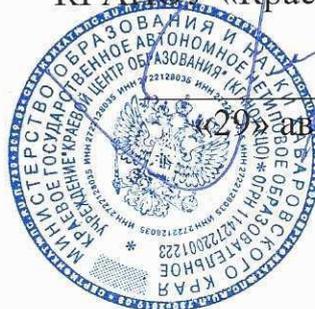


Министерство образования и науки Хабаровского края  
Краевое государственное автономное нетиповое  
общеобразовательное учреждение  
«Краевой центр образования»

РАССМОТРЕНО  
на заседании Педагогического  
совета КГАНОУ «Краевой центр  
образования»  
Протокол №1 от  
«29» августа 2023 года

УТВЕРЖДАЮ  
Генеральный директор  
КГАНОУ «Краевой центр образования»



/Черёмухин П.С.  
«29» августа 2023 года

ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ  
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА ТЕХНИЧЕСКОЙ  
НАПРАВЛЕННОСТИ

«Основы программирования на языке Python»

Уровень освоения: базовый  
Возраст учащихся: 11-18 лет  
Общий объем программы в часах: 144 часа

Составители программы:  
Громогласов В.В., ПДО  
Валетова М.А., методист

Хабаровск  
2023 г.

## 1. Комплекс основных характеристик ДООП

### Пояснительная записка

**Дополнительная общеобразовательная общеразвивающая программа «Основы программирования на языке Python.» имеет техническую направленность (IT – технологии), базовый уровень.**

Программа разработана с учетом следующих нормативно-правовых документов:

- Федерального закона РФ от 29.12.2012г. № 273-ФЗ «Об образовании в Российской Федерации» (с изменениями и дополнениями);
- Приказа Министерства Просвещения РФ от 27.07.2022 г. № 629 «Об утверждении порядка организации и осуществления образовательной деятельности по дополнительным общеобразовательным программам»;
- Приказа Министерства труда и социальной защиты РФ от 22.09.2021г. № 652н «Об утверждении профессионального стандарта «Педагог дополнительного образования детей и взрослых»;
- Постановления Главного государственного санитарного врача РФ от 28.09.2020г. № 28 "Об утверждении санитарных правил СП 2.4.3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи»;
- Распоряжения Правительства РФ от 31.03.2022г. № 678-р «Концепция развития дополнительного образования детей до 2030 года»;
- Положения о дополнительной общеобразовательной программе, реализуемой в Хабаровском крае, утвержденного приказом КГАОУ ДО РМЦ от 26.09.2019 г. № 383П;
- Устав краевого государственного автономного нетипового образовательного учреждения «Краевой центр образования».

**Актуальность** программы обусловлена широким распространением информационно-коммуникационных технологий в обществе и необходимостью обеспечивать связанную с этим инфраструктуру специалистами. При этом язык программирования Python является достаточно эффективным и доступным инструментом достижения задач в области создания программных продуктов.

**Педагогическая целесообразность.** В связи с простотой синтаксиса в сравнении с другими языками программирования (ясность кода, быстрота реализации) при изучении Python есть возможность сформировать у обучающихся представление о базовых понятиях структурного программирования (данных, переменных, ветвлениях, циклах и функциях). В

то же время Python является востребованным языком, он отлично подходит для знакомства с концепцией объектно-ориентированного программирования и активно применяется в различных областях от разработки веб-приложений до машинного обучения. Научившись программировать на языке Python, обучающиеся получают мощный и удобный в использовании инструмент для решения учебных задач и для создания собственных проектов. Вместе с тем чистота и ясность его конструкций позволит в дальнейшем с лёгкостью выучить любой другой язык программирования.

**Адресат программы:** Программа ориентирована на обучающихся возрастной категории от 11 до 18 лет. Состав группы постоянный, количество обучающихся 12 человек.

**Форма обучения:** очная

**Объём реализации программы:**

Период	Продолжительность занятия	Кол-во занятий в неделю	Кол-во часов в неделю	Кол-во недель	Кол-во часов в год
1 год	2 часа	2	4 ч	36	144 ч
Итого по программе:					144 ч

**Режим организации занятий:** Занятия в объединении рекомендуется проводить по 2 часа 2 раза в неделю. Продолжительности занятия - 45 минут. После 45 минут занятий организовывается перерыв длительностью 10 минут для проветривания помещения и отдыха обучающихся

### Цель и задачи программы

**Цель программы:** Развитие познавательной активности, интеллектуальных и творческих способностей учащихся, в процессе обучения программированию на языке Python.

**Задачи программы:**

- формирование и развитие навыков алгоритмического и логического мышления, грамотной разработки программ;
- знакомство с принципами и методами функционального программирования;
- знакомство с принципами и методами объектно-ориентированного программирования;
- приобретение навыков работы в интегрированной среде разработки на языке Python;
- знакомство с основными структурами данных и типовыми методами обработки этих структур;
- формировать профессиональный интерес в целом к программированию и в частности к профессии программиста.

## Учебный план

№ п/п	Название разделов, тем	Количество часов			Формы аттестации/контроля
		Всего	Теория	Практика	
<b>1.</b>	<b>Основы программирования на языке Python</b>	<b>50</b>	<b>16</b>	<b>34</b>	Тестирование, решение практических задач
1.1	Вводное занятие. ТБ. Знакомство с программой.	2	2	-	
1.2	История языков программирования. Введение в язык программирования Python. Среда программирования Python.	2	1	1	
1.3	Типы данных и функции вывода. Определение переменной	2	1	1	
1.4	Переменные и арифметические выражения	2	1	1	
1.5	Чтение данных. Операции над строками	2	1	1	
1.6	Примеры решения задач.	2	-	2	
1.7	Отработка навыков решения простейших задач.	2	-	2	
1.8	Логический тип данных и операции. Примеры использования логических выражений.	2	1	1	
1.9	Условный оператор. Вложенный условный оператор.	4	2	2	
1.10	Примеры решения задач.	4	-	4	
1.11	Отработка навыков решения простейших задач.	4	-	4	
1.12	Цикл WHILE. Примеры решения задач.	4	2	2	
1.13	Подсчет суммы и оператор CONTINUE. Примеры решения задач.	4	2	2	
1.14	Отработка навыков решения простейших задач.	4	-	4	

1.15	Вещественные числа. Основы работы с вещественными числами.	4	2	2
1.16	Округление вещественных чисел. Примеры решения задач.	4	1	3
1.17	Отработка навыков решения простейших задач.	2	-	2
<b>2.</b>	<b>Методы программирования на языке Python. Введение в олимпиадное программирование</b>	<b>50</b>	<b>9</b>	<b>41</b>
2.1	Срезы строк. Использование срезов.	2	1	1
2.2	Метод FIND. Примеры решения задач.	2	1	1
2.3	Отработка навыков решения простейших задач.	2	-	2
2.4	Методы RFIND, REPLACE и COUNT. Примеры решения задач.	2	1	1
2.5	Отработка навыков решения простейших задач.	2	-	2
2.6	Решение несложных олимпиадных задач.	2	-	2
2.7	Функции. Использование функций.	2	1	1
2.8	Примеры решения задач.	2	-	2
2.9	Возврат значений. Локальные и глобальные переменные.	2	1	1
2.10	Примеры решения задач.	2	-	2
2.11	Отработка навыков решения задач.	2	-	2
2.12	Рекурсия. Использование рекурсии.	2	1	1
2.13	Примеры решения задач.	2	-	2
2.14	Отработка навыков решения задач.	4	-	4
2.15	Кортежи. Функция RANGE, цикл FOR.	4	2	2
2.16	Примеры решения задач.	2	-	2
2.17	Отработка навыков решения задач.	4	-	4
2.18	Списки. Метод SPLIT и JOIN.	2	1	1
2.19	Примеры решения задач.	2	-	2

2.20	Отработка навыков решения задач	4	-	4		
2.21	Решение олимпиадных задач.	2	-	2		
2.22	Отработка навыков решения сложных задач.	2	-	2		
<b>3.</b>	<b>Объектно-ориентированное программирование. Решение задач повышенной сложности.</b>	<b>44</b>	<b>12</b>	<b>32</b>		
3.1	Парадигмы программирования и функциональное программирование.	2	1	1		
3.2	Встроенные функции для работы с последовательностями.	4	2	2		
3.3	Примеры решения задач в функциональном стиле.	4	1	3		
3.4	Итераторы и генераторы. Генерация комбинаторных объектов ITERTOOLS.	4	2	3		
3.5	Функции PARTIAL, REDUCE, ACCUMULATE.	4	2	3		
3.6	Примеры решения задач.	4	-	4	Тестирование , решение практических задач	
3.7	Отработка навыков решения сложных задач.	4	-	4		
3.8	Объектно-ориентированное программирование.	4	2	4		
3.9	Комплексные числа. Инкапсуляция и конструкторы.	2	1	1		
3.10	Примеры решения задач.	2	-	2		
3.11	Классы объектов.	2	1	1		
3.12	Обработка ошибок. Наследование и полиморфизм.	2	-	2		
3.13	Решение задач повышенной сложности.	2	-	2		
3.14	Итоговая аттестация.	2	-	2		Решение практической задач
	<b>Итого:</b>	<b>144</b>	<b>37</b>	<b>107</b>		

### Содержание учебного плана

#### Раздел 1. Основы программирования на языке Python.

*Теория:* Инструктаж по технике безопасности. Введение в математическое программирование. История языков программирования. Введение в язык программирования Python. Типы данных и функции вывода.

Определение переменной. Переменные и арифметические выражения. Чтение данных. Операции над строками. Логический тип данных и операции. Примеры использования логических выражений. Условный оператор. Вложенный условный оператор. Цикл WHILE. Подсчет суммы и оператор CONTINUE. Вещественные числа. Основы работы с вещественными числами. Округление вещественных чисел.

*Практика:* Установка программы. Примеры решения задач. Отработка навыков решения простейших задач.

## **Раздел 2. Методы программирования на языке Python. Введение в олимпиадное программирование.**

*Теория:* Срезы строк. Использование срезов. Метод FIND. Методы RFIND, REPLACE и COUNT. Функции. Использование функций. Возврат значений. Локальные и глобальные переменные. Рекурсия. Использование рекурсии. Кортежи. Функция RANGE, цикл FOR. Списки. Метод SPLIT и JOIN.

*Практика:* Примеры решения задач. Отработка навыков решения простейших задач. Решение несложных олимпиадных задач. Отработка навыков решения задач. Решение олимпиадных задач.

## **Раздел 3. Объектно-ориентированное программирование. Решение задач повышенной сложности.**

*Теория:* Парадигмы программирования и функциональное программирование. Встроенные функции для работы с последовательностями. Итераторы и генераторы. Генерация комбинаторных объектов ITERTOOLS. Функции PARTIAL, REDUCE, ACCUMULATE. Объектно-ориентированное программирование. Комплексные числа. Инкапсуляция и конструкторы. Классы объектов. Обработка ошибок. Наследование и полиморфизм.

*Практика:* Отработка навыков решения задач. Решение олимпиадных задач. Примеры решения задач в функциональном стиле. Примеры решения задач. Отработка навыков решения сложных задач.

### **Планируемые результаты**

*Предметные:*

- применяют цифровых технологий в процессе образовательной деятельности; умеют использовать на практике как простые, так и сложные структуры данных и конструкции для работы с ними;
- умеют разбивать решение задачи на подзадачи;
- смогут писать грамотный, красивый код;
- смогут отлаживать программный код средствами среды разработки.

*Метапредметные:*

- смогут организовывать сотрудничество и совместную деятельность с учителем и сверстниками;
- смогут самостоятельно разрабатывать алгоритмы достижения целей (решения задач, выполнения проектов и т.д.).

*Личностные:*

- приобщение учащихся к информационной культуре, раскрытие творческих способностей детей;
- приобретение начальной профессиональной подготовки по данному направлению, что поспособствует повышению социальной адаптации учащихся после окончания школы;
- совершенствование навыков работы с компьютером и умение применять изученные алгоритмы для решения актуальных задач.

## 2. Комплекс организационно – педагогических условий

### Календарный учебный график программы

Год обучения	Дата начала занятий	Дата окончания занятий	Кол-во учебных недель	Кол-во учебных дней	Кол-во учебных часов	Режим занятий
1 год обучения	01.09.2023 г.	29.05.2024 г.	36	72	144	2 раза в нед. по 2 часа

### Условия реализации программы

#### Материально-техническое обеспечение:

Персональный компьютер – по 1 шт. на одного обучающегося:

- Не менее 4 ГБ оперативной памяти;
- Процессор с тактовой частотой не менее 1,2 ГГц;
- Диагональ мониторов не менее 12 дюймов;
- Наличие колонок или наушников;
- Свободные 50 ГБ на накопителях;
- компьютеры (ноутбуки) должны быть подключены к единой сети с доступом в Интернет не медленнее 10 Мбит/с;
- презентационное оборудование (проектор с экраном/телевизор с большим экраном) с возможностью подключения к компьютеру (ноутбуку) – 1 комплект;
- флипчарт с комплектом листов / маркерная доска, соответствующий набор письменных принадлежностей – 1 шт.

#### Программное обеспечение:

- Операционная система — любая, желательно Windows;
- Любой современный браузер (например, Яндекс.Браузер, Google Chrome, Mozilla Firefox, Safari);
- Python версии не ниже 3.6;

- среда разработки. PyCharm Community Edition версии не ниже 2020.1.

#### **Информационное обеспечение:**

- Документация Python - <https://docs.python.org/3/>
- "Поколение Python": курс для продвинутых - <https://stepik.org/course/68343/promo>;
- Python для продвинутых - [https://ru.hexlet.io/courses/advanced\\_python](https://ru.hexlet.io/courses/advanced_python);
- Информатика. Алгоритмы и структуры данных на Python. Лекции Хирьянова Т. - [http://judge.mipt.ru/mipt\\_cs\\_on\\_python3](http://judge.mipt.ru/mipt_cs_on_python3).

#### **Кадровое обеспечение:**

Программа реализуется педагогом дополнительного образования.

#### **Формы аттестации**

Для отслеживания результативности и динамики освоения ребенком образовательной программы педагогу необходимо понимать, какой уровень развития тех или иных навыков имеет учащийся при зачислении на обучение. Поэтому проводится входящая диагностика по диагностической карте промежуточной аттестации, чтобы определить уровень развития ребенка по определенным параметрам, заложенным картой. Также используются такие формы и методы контроля, как:

- наблюдение;
- устный опрос;
- устный анализ самостоятельных работ;
- тестирование, самостоятельные и контрольные работы;
- промежуточная аттестация - диагностика по параметрам – в сентябре, декабре и мае;
- обучение заканчивается написанием программы для решения одной из задач.

*Наблюдение* – необходимый педагогу метод для осуществления промежуточной и текущей аттестации, применяется педагогом постоянно. *Устный опрос* – метод при котором педагог может оценить теоретически знания учащихся. Проводится в конце пройденной темы.

*Устный анализ самостоятельных работ* – дает возможность учащимся научиться логически мыслить и уметь высказать собственное суждение, поможет педагогу оценить логическое мышление учащихся. Проводится в конце пройденной темы.

*Тестирование, самостоятельные и контрольные работы* – помогут учащимся проверить свои знания по пройденным темам и оценить приобретенные умения, проводятся в конце каждого раздела. Участие в проектной деятельности – это условие, необходимое для становления

самостоятельности учащихся, которые сформируют установку на самостоятельность, индивидуальные цели и возможности их реализации; *Промежуточная аттестация* проводится три раза за учебный год: в начале учебного года – в сентябре, в середине учебного года – в декабре и в конце учебного года – в мае. Задачей промежуточной аттестации является определение эффективности педагогического воздействия и полноты освоения образовательной программы.

Формы предъявления и демонстрации образовательных результатов: участие в предметных олимпиадах и конкурсах.

### Оценочный материал

Мониторинг результатов обучения ребёнка по дополнительной образовательной программе: промежуточная аттестация (Приложение 1), итоговая аттестация (Приложение 2).

#### Диагностическая карта промежуточной аттестации

№	Ф.И. ребенка	Параметры						
		Работа в команде	Самостоятельность в выполнении задач	Участие в конкурсах	Специальные знания	Алгоритмическое логическое мышление	Поиск и обработка информации в сети Интернет	ИТОГ

1 балл – низкий уровень; 2 балла – средний уровень; 3 балла – высокий уровень.

При подсчете баллов по каждому учащемуся можно определить уровень освоения им программы, также по каждой группе и по объединению в целом.

Определение уровня каждого учащегося выводится после подсчета баллов по всем параметрам:

- от 1 до 8 баллов – низкий уровень освоения программы учащимся;
- от 9 до 21 баллов – средний уровень освоения программы учащимся;
- от 22 до 27 баллов – высокий уровень освоения программы учащимся.

После этого подсчитывается процентное соотношение уровней освоения по группам и по объединению и результат заносится в сводную таблицу.

#### Сводная таблица

	Высокий уровень %	Средний уровень %	Низкий уровень %
На начало года реализации образовательной программы			
На середину года реализации			

образовательной программы			
На конец года реализации образовательной программы			

По этой таблице можно провести анализ результативности данной программы.

### **Методическое обеспечение**

Методика преподавания включает разнообразные формы, методы и приемы обучения и воспитания. Обоснованность применения различных методов обусловлена тем, что нет ни одного универсального метода для решения разнообразных творческих задач

#### *Методы обучения*

Методы обучения, применяемые в реализации программы, можно систематизировать на основе источника получения знания:

- словесные: рассказ, объяснение, беседа, дискуссия;
- наглядные: демонстрация дидактических материалов, видеофильмов; компьютерные игры.
- практические: работа с аудио- и видеоматериалами, тематические экскурсии, интернет-экскурсии, тренинги, участие в мероприятиях.

Вместе с традиционными методами на занятиях спешно используются активные методы обучения: мозговой штурм, моделирование, метод проектов, метод эвристических вопросов, игровые ситуации, анализ конкретных ситуаций (case-study) и др.

Выбор методов обучения зависит от дидактических целей, от характера содержания занятия, от уровня развития обучающихся.

#### *Формы организации образовательного процесса*

Занятия проводятся с использованием различных форм организации учебной деятельности (групповая, фронтальная, индивидуальная, индивидуальная дистанционная, групповая дистанционная). Разнообразные формы обучения и типы занятий создают условия для развития познавательной активности, повышения интереса детей к обучению.

#### *Формы организации учебного занятия*

Основной формой проведение учебных занятий является практическое занятие. Однако в ходе реализации программы, педагог вправе применять любую из доступных форм организации учебного занятия: беседа, встреча с интересными людьми, выставка, диспут, защита проектов, игра, конкурс, конференция, круглый стол, лекция, мастеркласс, «мозговой штурм», наблюдение, олимпиада, презентация, семинар, соревнование, чемпионат, экскурсия.

Типы занятий: изучение новой информации, занятия по формированию

новых умений, обобщение и систематизация изученного, практическое применение знаний, умений, комбинированные занятия, контрольно-проверочные занятия.

*Педагогические технологии, используемые в образовательном процессе*

1. Проектная технология, учебно-исследовательская деятельность. На протяжении всего курса обучения учащиеся вовлечены в учебно-исследовательскую деятельность, которая позволяет им находить, обрабатывать, сравнивать и систематизировать информацию, полученную из встреч с интересными людьми, публикаций в сети Интернет. В ходе образовательного процесса учащиеся создают и защищают собственные проекты, учатся методам поиска информации, самопрезентации, которые необходимы им в дальнейшей жизни и профессиональной карьере, на практических занятиях учащиеся выполняют исследовательские проекты. Проектная деятельность позволяет учащимся принять активную гражданскую позицию, сформировать потребность в участии в общественно полезной деятельности, необходимость быть нужным обществу. На занятиях создаются и реализуются учебные мини-проекты, в которых учащиеся решают учебные задачи на основе построения последовательности этапов от цели к конкретному результату. В процессе обучения осуществляется знакомство учащихся с информационно-коммуникационными технологиями, достижениями науки техники в области инженерной мысли. Современные педагогические технологии в сочетании с современными информационными технологиями могут существенно повысить эффективность образовательного процесса, решить стоящие перед педагогом задачи воспитания всесторонне развитой, творчески свободной личности.

2. Технология развития критического мышления помогает учащимся определять приоритеты, анализировать, оценивать, выявлять ошибки, повысить мотивацию. Осуществляется при совместной работе в группах, при взаимодействии во время выполнения заданий, при диалоге обучающихся между собой и с педагогом. Обязательным условием является сбор данных о динамике обучающегося и анализ его достижений и трудностей.

Алгоритм формирования критического мышления, предполагающий ответы на следующие вопросы:

1. Какова цель данной познавательной деятельности?
2. Что известно?
3. Что делать?
4. Достигнута ли поставленная цель?

Таким образом, критическое мышление - значит «искусство суждения, основанное на критериях». Результат - владение стратегиями критического мышления.

3. Технология имитационной игры – это моделирование реальной деятельности в специально созданных условиях, а её элементы включают в себя взаимосвязанные знаниевые и деятельностные компоненты обучения. Особенности:

- не моделируется труд конкретных работников;
- имитируются лишь некоторые хозяйственные, правовые, экономические, экологические, социально-психологические принципы, определяющие поведение людей и механизмы их действий (в экстремальных ситуациях);
- моделирование только среды, особенности среды знакомы играющим в основном понаслышке, что делает анализ информации более сложным и субъективным;
- общая цель всего игрового коллектива изначально не задана, и для ее достижения самими игроками может быть найден определенный механизм взаимодействия;
- отсутствуют альтернативы, участники должны действовать лишь в предложенных вариантах;
- не программируется конфликтная ситуация (как, например, в деловых играх), а представлены только различные личные (субъективные) интересы участников игры;
- описанные сценарии игр не включают технологии и механизмы специального обучения общению и коллективному принятию решений.

4. Технология проблемного обучения способствует развитию проблемного мышления учащихся и педагога.

Результаты:

- усвоение учащимися системы знаний и способов умственной деятельности; – развитие интеллектуальных умений и навыков учащихся;
- усвоение способов организации познавательной деятельности и формирования познавательной самостоятельности;
- развитие интеллектуальных возможностей, включающих творческие способности и прошлый опыт учащихся.

Проблемный вопрос - это входящий в состав проблемной задачи или отдельно взятый учебный вопрос (вопрос-проблема), требующий ответа на него посредством мышления. Вопрос же, требующий воспроизведения по памяти, не является проблемным. Вопросы, стимулирующие мышление, начинаются с таких вопросительных слов и словосочетаний, как «почему», «отчего», «как (чем) это объяснить», «как это понимать», «как доказать (обосновать)», «что из этого следует (какой вывод)» и т.п. А вопросительные слова «кто», «что», «когда», «где», «сколько», «какой» всегда требуют ответа на основе памяти.

Проблемная задача – учебная проблема с четкими условиями, задаваемыми преподавателем (лектором) или выявленными и сформулированными кем-либо из обучаемых (студентов), и в силу этого получившую ограниченное поле поиска (в отличие от объективно возникающей перед человеком жизненной проблемы) и ставшую доступной для решения всеми обучаемыми (студентами).

Проблемная ситуация – это ситуация познавательного затруднения, вовлекающая учащихся в самостоятельное познание элементов новой темы.

5. Интерактивные технологии направлены на развитие готовности к

организации группового общения.

Результаты:

- готовность воспринимать многомерность информацию, работать в режиме полилога;
- способность выбирать и обосновывать выбор методов, форм и техник организации коммуникационного процесса;
- владение психологическими техниками и методами организации коммуникационного процесса.

6. Технология дискуссионного общения включает в себя взаимосвязанные компоненты:

- мотивационный (готовность, желание принять участие в дискуссии);
- познавательный (знание о предмете спора, проблемная ситуация);
- операционно-коммуникативный (умение вести спор, отстаивать свою точку зрения, владеть способами осуществления логических операций);
- эмоционально-оценочный (эмоциональные переживания, потребности, отношения, мотивы, оценки, личностный смысл).

### Календарный план воспитательной работы

№ п/п	Наименование мероприятия	Форма проведения	Срок исполнения
1.	Проведение инструктажей с обучающимися по порядку действия в случае возникновения пожара, правилам поведения на воде и дорогах.	Час общения	сентябрь
2.	Единый урок по безопасности: «Безопасность на дорогах глазами детей»; «Безопасность в сети Интернет»; «Осторожно! Тонкий лёд. Падение снежных масс и наледи»; «К нам приходит Новый год!»; «Что мы знаем о терроризме»; «Я и мои виртуальные друзья»; «Действия при пожаре – правила пожарной безопасности»; «Осторожно! Загрязнение пластмассовыми материалами»; «Безопасное лето».	Дискуссия, устный журнал, встречи с сотрудниками МЧС, составление памяток и рекомендаций	ежемесячно
3.	Уроки истории нашей страны: «Моя малая родина»; «Край родной – навек любимый»; «Интересные и знаменитые люди	Фотовыставка, устный журнал, поисково-исследовательская работа, встреча с	ноябрь январь апрель

	нашего края».	интересными людьми	
4.	День воинской славы России: День героев Отечества России; День защитника Отечества; День Победы.	Военно- патриотическая беседа, встречи с участниками боевых действий, экскурсия в музей	декабрь февраль май
5.	«Как у наших у ворот...» – Масленица; – Пасха. Пасхальные торжества.	Игровая программа	март апрель
	Мы разные, но у нас равные права!	Занятие- обсуждение	ноябрь
	Честность прежде всего	Устный журнал	январь
	Что такое «хорошо» и что такое «плохо»?	Дискуссия	март
	«Моя семья - моя крепость»	Детско-родительская встреча	май
	«Что значит быть ответственным»	Занятие- обсуждение	май
6.	«Речная лента», «Покормите птиц зимой» и др.	акции	В течение года
7.	Подготовка и участие в конкурсах, олимпиадах, хакатонах и других мероприятиях	Образовательное мероприятие	В течение года
	Цикл мероприятий «Участуй в НТО»	Образовательное мероприятие	Сентябрь – октябрь
	Экскурсии в IT-компаниях	экскурсия	По согласованию
	Ярмарка проектов и достижений	День открытых дверей	Декабрь, май

## Список литературы

### *Список литературы для педагога:*

1. Гаско, Рик. Простой Python просто с нуля [Электронный ресурс] / Рик Гаско. – М.: СОЛОН-Пресс, 2019.
2. Златопольский, Д.М. Программирование: типовые задачи, алгоритмы, методы [Электронный ресурс] / Д.М. Златопольский. – М.: БИНОМ. Лаборатория знаний, 2007.
3. Мюллер, Джон Пол. Python для чайников [Электронный ресурс] / Джон Пол Мюллер. - 2-е изд.: Пер с англ. – СПб.: ООО «Диалектика», 2019.
4. Окулов, С.М. Основы программирования: учебное пособие [Электронный ресурс] / С.М. Окулов. – 10-е изд., электрон. – Москва : Лаборатория знаний, 2020.
5. Пупышев, В.В. 128 задач по началам программирования [Электронный ресурс] / В.В. Пупышев. – М.: БИНОМ. Лаборатория знаний, 2009.
6. Свейгарт, Эл. Учим Python, делая крутые игры [Электронный ресурс] / Эл. Свейгарт. - М: Эксмо, 2018.
7. Седер, Наоми. Python. Экспресс-курс [Электронный ресурс] / Наоми Седер. - 3-е изд. – СПб.: Питер, 2019.

### *Список литературы для обучающихся:*

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. Златопольский, Д. М. Сборник задач по программированию [Электронный ресурс] / Д.М. Златопольский. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011.
3. 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.
4. Пэйн, Брайсон. Python для детей и родителей [Электронный ресурс] / Брайсон Пейн [пер. с англ. М.А. Райтмана]. - М.: Издательство «Э», 2017.
5. Столяров, А.В. Оформление программного кода [Электронный ресурс] / А.В. Столяров. – 2-е изд., испр. и доп. – Москва: МАКС Пресс, 2019.
6. Шень, А. Программирование: теоремы и задачи [Электронный ресурс] / А. Шень. - 6-е изд., дополненное. М.: МЦНМО, 2017.
7. Шуман, Х.Г. Python для детей [Электронный ресурс] / Х.Г. Шуман [пер. с нем. М.А. Райтман]. – М.: ДМК Пресс, 2019.

**Промежуточная аттестация.**

Промежуточная аттестация проходит в форме компьютерного тестирования. Выберите БУКВУ, соответствующую верному ответу:

- 1) Что представляют собой словари в Python?
  - a) массивы
  - b) отсортированные списки
  - c) json-объекты
  - d) неупорядоченные коллекции
  
- 2) Что возвращает метод словаря items()?
  - a) ключи
  - b) типы ключей и значений
  - c) значения
  - d) пары ключ/значение
  
- 3) С чем напрямую связан метод?
  - a) с функцией
  - b) с процедурой
  - c) с классом
  - d) с интерфейсом
  
- 4) Какая структура хранения данных лишняя (отличается от остальных по своему внутреннему устройству/порядку хранения значений)
  - a) Список
  - b) Кортеж
  - c) Массив
  - d) Множество
  
- 5) Какая структура данных в Python является ассоциативным массивом
  - a) Список
  - b) Строка
  - c) Словарь
  - d) Множество
  
- 6) Что делает данная строка кода: `arr[arr[:pos]]+[num]+arr[pos:]`
  - a) Заменяет элемент в позиции pos на новый элемент из переменной num
  - b) Вставляет значение переменной num в позицию pos
  - c) Эта строка вызовет ошибку. Складывать списки и числа нельзя
  - d) Формирует список из трех элементов: pos, num, pos
  
- 7) Какое зарезервированное слово применяется для определения функции в Python?
  - a) func
  - b) def
  - c) define

- d) procedure
- 8) Чем являются функции в Python?
- a) объектами
  - b) методами
  - c) экземплярами
  - d) классами
- 9) Какая конструкция служит для создания анонимной функции?
- a) lambda func
  - b) lambda
  - c) anonymous
  - d) omega
- 10) Что такое рекурсия?
- a) подбор параметров функции
  - b) описание какого-либо объекта внутри этого самого объекта
  - c) возвращение результатов функции
  - d) вызов функции в обратном порядке ее алгоритма

11) Что выведет этот код

```
def get_summa(a, b):
```

```
    pass
```

```
print(get_summa(4,2))
```

- a) 4
- b) 2
- c) 6
- d) None

12) Что выведется в результате выполнения кода `def get_summa{a=2, b=3};`

```
print(a+b)
```

```
get_summa {4}
```

- a) 6
  - b) 7
  - c) 5
  - d) 4
- 13) Что относится к основным принципам ООП?
- a) Инкапсуляция, полиморфизм, наследование, абстракция
  - b) Инкапсуляция, полиморфизм, делегирование, абстракция
  - c) Полиморфизм, разделение интерфейса, наследование, абстракция
  - d) Инкапсуляция, наследование, абстракция, открытость/закрытость
- 14) Какой принцип ООП описывает следующее предложение: «Этот принцип является способностью использовать общий интерфейс для нескольких форм (типов данных)».

- a) Инкапсуляция

- b) Полиморфизм
  - c) Абстракция
  - d) Наследование
- 15) Как создать конструктор класса А?
- a) А(параметры конструктора)
  - b) def \_\_init\_\_(параметры конструктора)
  - c) def \_\_A\_\_(параметры конструктора)
  - d) def init(параметры конструктора)
- 16) Какой параметр обязательно принимает в себя метод экземпляра?
- a) тип объекта
  - b) сам экземпляр объекта
  - c) название класса
  - d) таких аргументов нет

17) Укажите результат выполнения скрипта:  
x=0

```
class Foo:
    count=x
    def init(self) :
```

```
    self.count+=1
```

```
obj = Foo()
print (obj.count)
```

- a) 0
  - b) 1
  - c) 2
  - d) пустая строка
  - e) скрипт не будет выполнен, т.к. код содержит ошибки
- 18) Как нужно задавать начальные значения полей?

- a) их вообще не задают
  - b) в конструкторе класса
  - c) в специальном методе
  - d) для каждого поля нужен свой метод
  - e) их задают после создания объекта
- 19) Отметьте методы приведённого класса объектов: class TCat:

```
def init ( self ) : self.name = ""
self.age = 5
```

```
def say(self): print C'M.qy! ")
```

```
def run(self): print (self . name, "нобеа;aji! ")
```

- a) say, name
- b) say, name, age
- c) TCat, name, say
- d) TCat, name, say, age

20) Что будет выведено на экран? class Test:

```
def print_text(self) :
```

```
print {'3to ponjiTejibciaiii KJiacc Test")
```

```
class Test2(Test):
```

```
def print_text(self) :
```

```
print {"3to nonepHuft KJiacc Test2")
```

```
tst = Te3t2 0
```

```
tst.print_text()
```

- a) Это родительский класс Test
- b) Это дочерний класс Test2
- c) SyntaxError: invalid syntax
- d) Это родительский класс Test
- e) Это дочерний класс Test2

### **Критерии оценки:**

Правильные ответы выделены в тексте заливкой. Каждый правильный ответ оценивается в один балл. Максимальное количество баллов – 20. Набранные баллы переводятся в уровень освоения по следующей шкале:

1. 5 – 9 баллов: низкий уровень;
2. 10 – 14 баллов: средний уровень;
3. 15 – 20 баллов: высокий уровень.

## Образец итоговой аттестации

### *Краткое описание математической задачи:*

Дан список целых чисел. Посчитать, сколько раз в нем встречается каждое число. Например, если дан список [1,1,3,2,1,3,4], то в нем число 1 встречается три раза, число 3 – два раза, числа 2 и 4 – по одному разу.

Для хранения количества каждого встречающегося в списке значения создадим словарь. В нем ключами будут числа, которые встречаются в списке, а значениями количества этих чисел в списке. Для примера, приведенного выше, в итоге должен был бы получиться такой словарь: {1:3, 3:2, 2:1, 4:1}.

Пусть в программе будет функция, которая заполняет список случайными числами в диапазоне и количестве, указанными пользователем.

Другая функция будет считать количество каждого значения и заносить данные в словарь. Алгоритм подсчета заключается в следующем. Если очередной элемент списка уже есть в качестве ключа словаря, то следует увеличить значение этого ключа на единицу. Если очередного элемента списка нет в качестве ключа в словаре, то такой ключ следует добавить и присвоить ему значение, равное единице.

Для того, чтобы вывести содержимое словаря в отсортированном по возрастанию ключей виде, используя функции sorted(). Она сортирует ключи Словаря и помещает их в список.

### *Решение:*

```
def fill_list(min_bound, max_bound, amount_items,
             filling_list):
    from random import randint
    for i in range(amount_items):
        filling_list.append(randint(min_bound, max_bound))

def analysis(your_list, your_dict):
    for i in your_list:
        if i in your_dict:
            your_dict[i] += 1
        else:
            your_dict[i] = 1

lst = []
dct = {}

mn = int(input('Минимум: '))
mx = int(input('Максимум: '))
qty = int(input('Количество элементов: '))

fill_list(mn, mx, qty, lst)
analysis(lst, dct)

for item in sorted(dct):
    print("%d':%d" % (item, dct[item]))
```

### *Пример выполнения:*

```
Минимум: 1
Максимум: 11
Количество элементов: 20
'1':4
'3':7
'4':2
'6':6
'11':1
```