

Министерство образования и науки Пермского края  
Государственное бюджетное образовательное учреждение  
«Академия первых»

ПРИНЯТА  
педагогическим советом  
ГБОУ «Академия первых»  
Протокол от 26.07.2023 № 5

СОГЛАСОВАНО  
на заседании экспертного совета  
ГБОУ «Академия первых»  
Протокол от 20.12.2022 № 6

УТВЕРЖДЕНА  
приказом директора  
ГБОУ «Академия первых»  
от 27.07.2023 № 201



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА  
«Программирование на Python. Решение нестандартных задач»**

Возраст обучающихся: 10-15 лет (5-7 класс)

Срок реализации программы: 72 часа

Составитель программы:  
Перескокова Ольга Ивановна,  
кандидат технических наук,  
педагог дополнительного  
образования

## 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

**Направленность (профиль) программы:** настоящая программа «Программирование на Python. Решение нестандартных задач» является дополнительной общеразвивающей программой технической направленности. Программа относится к базовому уровню реализации, и предназначена для школьников 5-7 классов. Она позволяет познакомить учащихся с языком программирования Python, научить решать задачи по программированию в условиях автоматической проверки решений, и ввести их в мир олимпиадного программирования на задачах начального уровня сложности.

**Актуальность программы:** в современном IT-мире умение программировать становится базовым инструментом при решении многих задач. Среди школьников 5-7 классов имеются одаренные ученики, которые способны изучать программирование раньше, чем это запланировано школьной программой. Более того, есть олимпиады по программированию, ориентированные на учащихся этой возрастной категории. Ярким примером такой олимпиады является Всероссийская олимпиада по информатике им. Мстислава Келдыша для учащихся 5-8 классов. Данная программа предоставляет для школьников ранний старт в области программирования и позволяет войти в среду олимпиадного программирования уже в средней школе.

**Педагогическая целесообразность:** программирование на языках программирования высокого уровня в школе начинают изучать с 8 класса. Однако, имеются одаренные школьники, способные освоить этот материал раньше. Для учащихся 5-8 классов есть олимпиады по информатике, однако, дополнительных программ, позволяющих учащимся среднего школьного возраста изучать программирование и знакомиться с подходами к решению олимпиадных задач, катастрофически не хватает. Данный курс имеет целью восполнить существующий вакуум в образовании по программированию для учащихся 5-7 классов.

**Отличительные особенности программы:** дополнительная общеразвивающая программа «Программирование на Python. Решение нестандартных задач» рассчитана на интенсивный краткосрочный курс обучения. Она преследует две цели: знакомство с основами программирования на языке Python и изучение подходов к решению олимпиадных задач начального уровня.

Программа состоит из цикла очных занятий с разбором теоретического материала и решением задач. В курсе есть домашние задания и контрольные работы.

Отличительной особенностью программы является использование системы автоматической проверки решений задач по программированию. Она позволяет учащимся:

- 1) получать ответ о правильности решения задачи в любое время без участия преподавателя, что делает процесс решения задачи увлекательным и динамичным;
- 2) видеть свое место в автоматически обновляемом рейтинге участников курса, что вносит элемент соревновательности в учебный процесс.

Другой отличительной особенностью программы является адаптивность к уровню учащихся, сильным ученикам предлагаются для решения более сложные задачи, а учащимся, только начинающим изучать программирование, задачи начального уровня.

**Новизна программы:** новизной программы является с одной стороны возрастная категория учащихся, в другой стороны содержание практических заданий курса. Как правило курсы по началам программирования адресуются учащимся 8-10 классов. В данном курсе материал адаптирован для учащихся 5-7 классов, для этого разработаны разноуровневые практические задачи и детализирован теоретический материал. Задачи, рассматриваемые в курсе, имеют условие в традиционном формате олимпиадных задач и проверяются системой автоматической проверки решений. Задачи ранжированы по сложности. Задачи среднего и высокого уровня сложности являются некоей

головоломкой, решение таких задач «в лоб» не приводит к результату, требуется анализ различных подходов к решению и выбор наиболее эффективного из них. Все это позволит сделать «ранний старт» в мире олимпиадного программирования.

**Целью реализации** настоящей дополнительной общеразвивающей программы является изучение основ программирования на языке Python и развитие учащихся в области олимпиадного программирования (начальный уровень подготовки).

**Задачи реализации** программы разделяются в соответствии с кругом решаемых вопросов.

Обучающие задачи:

- изучение синтаксиса языка программирования Python;
- развитие навыков использования управляющих инструкций и структур данных языка Python при решении задач;
- изучение технологии решения задачи по программированию, проверяемой системой автоматически;
- изучение подходов к решению олимпиадных задач начального уровня.

Развивающие задачи:

- умение анализировать программу на языке программирования Python;
- развитие готовности к решению нестандартных задач;
- повышение мотивации учащихся заниматься саморазвитием в области олимпиадного программирования;
- развитие психологической готовности к участию в олимпиадах по программированию начального уровня.

Воспитательные задачи:

- обеспечение самостоятельности при решении задач учащимися и формирование негативного отношения ко всем вариантам заимствования чужих решений;

- формирование позитивного отношения к любым специализациям IT-отрасли и нацеленности на дальнейшее развитие в этой области.

**Адресат программы:** программа «Программирование на Python. Решение нестандартных задач» предназначена для детей, обучающихся в 5-7 классах общеобразовательных организаций, в том числе и для тех, кто ранее не умел программировать. Освоение программы научит программировать на языке Python и позволит уверенно чувствовать себя на олимпиадах по программированию.

**Срок реализации программы:** программа реализуется в течение 72 часов и сочетает очный и дистанционный форматы обучения.

**Форма обучения:** очно-заочная.

**Формы обучения:** настоящая программа предполагает, что занятия (2 часа в неделю) проводятся в формате очного обучения. Каждую неделю учащиеся получают задачи по пройденной теме в виде конкурса, которые решают в режиме самостоятельной работы (по 2 часа в неделю). При возникновении вопросов по решению задач, их можно задавать в чате группы или индивидуально преподавателю, в том числе при необходимости можно договариваться об индивидуальных онлайн-консультациях с преподавателем.

**Ожидаемые результаты обучения и способы определения их результативности:** в результате освоения дополнительной общеразвивающей программы «Программирование на Python. Решение нестандартных задач» учащийся должен знать: 1) синтаксис языка программирования Python; 2) способы тестирования своих решений, в том числе с использованием средств отладки программ; 3) причины, по которым решение не засчитывается системой автоматической проверки решений и способы устранения этих причин.

Ожидаемый результат по обучающему компоненту программы:

- овладеет навыками использования циклов и ветвлений при создании программ на языке программирования Python;

- умеет пользоваться структурами данных и методами работы с ними, предоставляемыми языком программирования Python.

Ожидаемый результат по развивающему компоненту программы.

- умеет анализировать программу на языке программирования Python, может предложить ее изменение с целью повышения ее эффективности;

- умеет довести решение задачи до полного цикла, для этого анализирует условие задачи, составляет тесты для ее проверки и пробует другие подходы к решению.

Ожидаемый результат по воспитательному компоненту программы:

- осознает, что самостоятельность в решении задач является основным фактором развития и относится негативно ко всем вариантам заимствования решений;

- понимает, что программирование является важным знанием в современном IT-мире, готов развиваться дальше в области программирования, в том числе олимпиадного.

**Способы определения результативности:**

- педагогическое наблюдение;
- изучение активности обучающихся на занятиях;
- ведение рейтинга обучающихся;
- ведение журнала учета.

**Формы подведения итогов реализации программы.** Промежуточный мониторинг осуществляется в виде решения конкурса. Будет вестись индивидуальный рейтинг каждого обучающегося.

## 2. УЧЕБНЫЙ ПЛАН

дополнительной общеразвивающей программы

«Программирование на Python. Решение нестандартных задач»

№	Наименование раздела (модуля)/ темы	Количество часов					Формы аттестации/ контроля
		Всего	Теория	Практика	Консультация	Самостоятельная работа	
1.	Программирование исполнителей.	5	1	1	1	2	Практическая работа
2.	Организация ввода-вывода. Простейшие вычисления по формуле.	4	1	1	0	2	Практическая работа
3.	Ветвление. Сложные условия. Вложенные ветвления.	5	1	1	1	2	Практическая работа
4.	Решение задач с использованием ветвлений.	4	1	1	0	2	Практическая работа
5.	Цикл for: использование с диапазонами range.	5	1	1	1	2	Практическая работа
6.	Решение задач с использованием циклов.	4	1	1	0	2	Практическая работа
7.	Операторы break и continue.	5	1	1	1	2	Практическая работа
8.	Цикл while.	4	1	1	0	2	Практическая работа
9.	Промежуточный контроль	5	0	2	1	2	Самостоятельная работа
10.	Вложенные циклы. Логические переменные.	4	1	1	0	2	Практическая работа
11.	Строки. Посимвольная обработка строк.	5	1	1	1	2	Практическая работа

12.	Срезы строк.	4	1	1	0	2	Практическая работа
13.	Методы работы со строками.	5	1	1	1	2	Практическая работа
14.	Списки. Индексация элементов списка.	4	1	1	0	2	Практическая работа
15.	Методы работы со списками.	5	1	1	1	2	Практическая работа
16.	Итоговый контроль	4	0	2	0	2	Самостоятельная работа
	ИТОГО	72	14	18	8	32	



### 3. СОДЕРЖАНИЕ ПРОГРАММЫ

Тема 1. Программирование исполнителей (5 часов).

*Теория:* Интерфейс исполнителя. Система команд исполнителя.

*Практика:* Решение задач на программирование конкретного исполнителя.

Тема 2. Организация ввода-вывода. Простейшие вычисления по формуле (4 часов).

*Теория:* Оператор вывода `print()`. Оператор ввода `input()`. Арифметические операции. Порядок выполнения арифметических операций в выражении. Типы данных: `int`, `str`, `float`. Преобразование типов данных.

*Практика:* Знакомство с системой программирования. Решение задач: вывод формулы по условию задачи.

Тема 3. Ветвление. Сложные условия. Вложенные ветвления (5 часов).

*Теория:* Оператор `if`. Логические связки `and`, `or`, `not` для формирования сложных условий. Полное и неполное ветвление. Использование `elif`. Вложенные ветвления. Эквивалентность операторов ветвления.

*Практика:* Решение задач: анализ ветвящихся алгоритмов. Формирование сложных условий.

Тема 4. Решение задач с использованием ветвлений (4 часов).

*Теория:* Приведение вложенного ветвления к «плоскому» виду и случаи, когда это возможно. Формула «округления вверх» с использованием ветвления и без него.

*Практика:* Решение задач: разбор случаев, вывод формулы для каждого из случаев.

Тема 5. Цикл `for`: использование с диапазонами `range` (5 часов).

*Теория:* Диапазоны `range()`: правила генерации последовательности. Использование цикла `for` совместно с диапазонами `range()`.

*Практика:* Решение задач: использование `for` для прямого и обратного счета, а также для счета с заданным шагом.

Тема 6. Решение задач с использованием циклов (4 часов).

*Теория:* Использование средств отладки программ. Трассировочная таблица.

*Практика:* Решение задач: решение содержательных задач на вывод закономерности и ее программирование с использованием цикла. Замена цикла формулой: для случаев арифметической и геометрической прогрессии.

Тема 7. Операторы break и continue (5 часов).

*Теория:* Досрочное прерывание цикла с помощью оператора break. Досрочный переход к следующей итерации цикла с помощью continue. Эквивалентность циклов с использованием break/continue и без него.

*Практика:* Решение задач: вычисление закономерностей с окончанием цикла по условию, пропуск некоторых элементов последовательностей.

Тема 8. Цикл while (4 часов).

*Теория:* Выполнение цикла while. Условие продолжение цикла и условие окончания цикла: трансформация одного в другое. Взаимозаменяемость циклов while и for.

*Практика:* Решение содержательных задач на использование цикла while.

Тема 9. Промежуточный контроль (5 часов)

*Практика:* Решение задач по пройденным темам в виде двухчасового контекста.

Тема 10. Вложенные циклы. Логические переменные (4 часов).

*Теория:* Взаимозаменяемость вложенных циклов одним циклом. Использование оператора break во вложенных циклах. Логические переменные и их использование в программах.

*Практика:* Решение задач на использование вложенных циклов. Избавление от вложенных циклов в программах.

Тема 11. Строки. Посимвольная обработка строк (5 часов).

*Теория:* Строковый тип данных. Индексация в строках. Посимвольная обработка строки с помощью цикла for.

*Практика:* Решение задач обработки строковых данных.

Тема 12. Срезы строк (4 часов).

*Теория:* Правила формирования среза строки. Конкатенация строк.

*Практика:* Решение задач на использование срезов строк.

Тема 13. Методы работы со строками (5 часов).

*Теория:* Поиск подстроки в строке. Определение количества вхождений подстроки в строку. Замена подстроки в строке. Использование стандартной справки Python.

*Практика:* Решение задач на использование стандартных функций обработки строк.

Тема 14. Списки. Индексация элементов списка (4 часов).

*Теория:* Списки (list) как тип данных. Обращение к элементу списка по номеру. Преобразование строки в список символов. Ввод списка из одной строки. Использование split() и join() для преобразования строки в список и обратно. Возможности вывода списка: использование распаковки, а также параметров sep и end оператора print().

*Практика:* Решение задач на использование списков. Сортировка подсчетом.

Тема 15. Методы работы со списками (5 часов).

*Теория:* Сортировка списка. Использование срезов в списках. Стандартные методы обработки списков. Вложенные списки, в том числе работа с двумерными массивами. Разница между строкой и списком в возможностях изменения отдельного элемента и последствия.

*Практика:* Решение задач обработки списков, в том числе работа с двумерными массивами.

Тема 16. Итоговый контроль (4 часов).

*Практика:* Итоговое соревнование с задачами по материалам всего курса.

#### 4. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

дополнительной общеразвивающей программы

«Программирование на Python. Решение нестандартных задач»

№ п/п	Месяц	Число	Время проведения занятия	Форма занятия	Кол-во часов	Тема занятия	Место проведения	Форма контроля
1.	Конкретная дата и время указываются преподавателем в журнале				<b>5, в том числе:</b>	<b>Входное тестирование</b>		Практическая работа
1.1				Лекция	1	Система команд исполнителя	Аудитория	
1.2				Практикум	1	Решение задач	Аудитория	
1.3				Консультация	1	Разбор задач	Дистанционно	
1.4				Самостоятельная работа	2	Дорешивание задач конкурса	Дистанционно	
2.	Конкретная дата и время указываются преподавателем в журнале				<b>4, в том числе:</b>	<b>Организация ввода-вывода. Простейшие вычисления по формуле.</b>		Практическая работа в форме конкурса
2.1				Лекция	1	Ввод-вывод. Арифметические выражения.	Аудитория	
2.2				Практикум	1	Решение задач	Аудитория	
2.3				Самостоятельная работа	2	Дорешивание задач конкурса	Дистанционно	
3.	Конкретная дата и время указываются преподавателем в журнале				<b>5, в том числе:</b>	<b>Ветвление. Сложные условия. Вложенные ветвления.</b>		Практическая работа в форме конкурса
3.1				Лекция	1	Ветвление в Python	Аудитория	
3.2				Практикум	1	Решение задач	Аудитория	
3.3				Консультация	1	Разбор задач		

3.4		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
4.	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Решение задач с использованием ветвлений.</b>		Практическая работа в форме контеста
4.1		Лекция	1	Анализ ветвлений	Аудитория	
4.2		Практикум	1	Решение задач	Аудитория	
4.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
5	Конкретная дата и время указываются преподавателем в журнале		<b>5, в том числе:</b>	<b>Цикл for: использование с диапазонами range.</b>		Практическая работа в форме контеста
5.1		Лекция	1	Цикл for	Аудитория	
5.2		Практикум	1	Решение задач	Аудитория	
5.3		Консультация	1	Разбор задач	Дистанционно	
5.4		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
6.	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Решение задач с использованием циклов.</b>		Практическая работа в форме контеста
6.1		Лекция	1	Методы отладки программ	Аудитория	
6.2		Практикум	1	Решение задач	Аудитория	
6.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
7.	Конкретная дата и время указываются преподавателем в журнале		<b>5, в том числе:</b>	<b>Операторы break и continue.</b>		Практическая работа в форме контеста
7.1		Лекция	1	Операторы break и continue	Аудитория	

7.2		Практикум	1	Решение задач	Аудитория	
7.3		Консультация	1	Разбор задач	Дистанционно	
7.4		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
8.	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Цикл while.</b>		Практическая работа в форме контеста
8.1.		Лекция	1	Цикл while	Аудитория	
8.2.		Практикум	1	Решение задач	Аудитория	
8.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
9.	Конкретная дата и время указываются преподавателем в журнале		<b>5, в том числе:</b>	<b>Промежуточный контроль</b>		Самостоятельная работа в форме контеста
9.1		Практикум	2	Решение задач	Аудитория	
9.2		Консультация	1	Разбор задач	Дистанционно	
9.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
10	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Вложенные циклы. Логические переменные.</b>		Практическая работа в форме контеста
10.1		Лекция	1	Вложенные циклы. Логические переменные.	Аудитория	
10.2		Практикум	1	Решение задач	Аудитория	
10.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
11	Конкретная дата и время указываются преподавателем в журнале		<b>5, в том числе:</b>	<b>Строки. Посимвольная обработка строк.</b>		Практическая работа в форме контеста
11.1		Лекция	1	Посимвольная обработка строк.	Аудитория	

11.2		Практикум	1	Решение задач	Аудитория	
11.3		Консультация	1	Разбор задач	Дистанционно	
11.4		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
12	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Срезы строк.</b>		Практическая работа в форме контеста
12.1		Лекция	1	Срезы строк	Аудитория	
12.2		Практикум	1	Решение задач	Аудитория	
12.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
13	Конкретная дата и время указываются преподавателем в журнале		<b>5, в том числе:</b>	<b>Методы работы со строками.</b>		Практическая работа в форме контеста
13.1		Лекция	1	Стандартные методы работы со строками	Аудитория	
13.2		Практикум	1	Решение задач	Аудитория	
13.3		Консультация	1	Разбор задач	Дистанционно	
13.4		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
14	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Списки. Индексация элементов списка.</b>		Практическая работа в форме контеста
14.1		Лекция	1	Списки в Python	Аудитория	
14.2		Практикум	1	Решение задач	Аудитория	
14.3		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
15	Конкретная дата и время указываются преподавателем в журнале		<b>5, в том числе:</b>	<b>Методы работы со списками.</b>		Практическая работа в форме контеста

15.1		Лекция	1	Методы работы со списками. Вложенные списки.	Аудитория	
15.2		Практикум	1	Решение задач	Аудитория	
15.3		Консультация	1	Разбор задач	Дистанционно	
15.4		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	
16	Конкретная дата и время указываются преподавателем в журнале		<b>4, в том числе:</b>	<b>Итоговый контроль</b>		Самостоятельная работа в форме контеста
16.1		Практикум	2	Решение задач	Аудитория	
16.2		Самостоятельная работа	2	Дорешивание задач контеста	Дистанционно	



## 5. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ

### 5.1. Материально-технические условия реализации программы

Наименование специализированных помещений, площадок, аудиторий, кабинетов, лабораторий	Форма (вид) занятий	Оборудование, программное обеспечение
Дистанционное обучение	Онлайн-консультация, самостоятельная работа (практикум)	Персональный компьютер с выходом в Интернет (желательно наушники с микрофоном)
Учебный класс (аудитория) при проведении очного обучения.	Лекция, практикум.	Компьютер, мультимедийный проектор, экран, доска, принтер. Каждому обучающемуся предоставляется компьютер или ноутбук для выполнения практических заданий. Для всех компьютеров должен быть доступ в Интернет и предустановленные системы программирования.

Для доступа в информационно- телекоммуникационную сеть Интернет рекомендуется использовать скорость подключения не менее 10 Мбит/сек.

Рабочее место преподавателя и обучающегося для очных занятий оборудуется персональным компьютером или ноутбуком. Компьютеры участников должны обладать следующими характеристиками:

- процессор с частотой не менее 1,5 ГГц;
- не менее 2 Гб оперативной памяти;
- не менее 1 Гб пространства на диске, доступных участнику для сохранения его файлов.

- монитор размером не менее 13 дюймов, разрешение экрана должно составлять не менее 1024\*768 пикселей.

Список компиляторов и сред разработки, которые должны быть предустановлены на компьютерах преподавателя и обучающихся.

Язык программирования	Компилятор / интерпретатор	Среда разработки
Python 3	Python 3.6 или более новая	IDLE, Wing IDE, PyCharm 2013.1 Community Edition или более новая

## 5.2. Учебно-методическое обеспечение программы

Образовательная программа содержит лекционную, методическую и практическую подготовку, большее количество времени уделяется выработке прикладных умений, лекции проводятся в интерактивном формате.

В основу программы положен комплекс педагогических технологий:

- системно-деятельностный подход, обеспечивающий развитие компетенций обучающихся, акцентирующий внимание на вовлечённости и самостоятельной работе слушателей;

- технология развивающего обучения, где слушателям предлагаются задачи, выходящие за пределы их зоны комфорта, ставящие школьников перед необходимостью проявления эвристики, а не повторения рутинизированных действий;

- технология формативного оценивания – на основании обратной связи в процессе научения педагог фиксирует наличие отстающих и общее понимание группой пройденного содержательного блока, в случае неудовлетворительной обратной связи, материал подаётся заново в адаптированном виде.

Программа реализуется с учётом ряда педагогических принципов:

- субъект-субъектные отношения, предполагающие соучастие слушателей и диалог, отсутствие жёстких сценариев большей части занятий;
- дифференцированное обучение, строящееся на концепции минимакса – минимальный уровень должны освоить все, но для лидеров, желающих взять больше знаний и навыков, не ставится верхний

(«достаточный») предел роста, сохраняется индивидуальный характер обучения;

- смена видов и форматов деятельности: для сохранения концентрации и интереса практикуется чередование периодов практической деятельности и усвоения информации, периоды интеллектуальной мобилизации и разрядки, периоды поточной, групповой и индивидуальной работы, также разнообразятся сами занятия (лекция, анализ материалов, беседа, дискуссия, тренинг, игра и т.д.).

Занятия проводит педагог, имеющий высшее педагогическое образование.

Программа составлена с учетом санитарно-гигиенических требований к порядку проведения занятий и адаптирована к возрастным особенностям обучающихся.

### 5.3. Список источников и литературы

#### Основная литература:

1. Поляков К.Ю. Программирование. Python. C++. Учебное пособие. В 4-х частях. – М.: Бином. Лаборатория знаний. 2020.
2. Чан Джейми. Python: быстрый старт. – СПб.: Питер, 2021.
3. A Byte of Python по-русски. [Электронный ресурс] URL: <http://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf>. (Дата обращения 08.07.2022)
4. Бриггс Джейсон. Python для детей. – М.: Манн, Иванов и Фербер, 2017.
5. Пейн Брайсон. Python для детей и родителей – М.: Издательство «Э», 2017.
6. Густокашин М. Курс лекций по олимпиадной информатике. [Электронный ресурс] URL: <https://informatics.msk.ru/mod/resource/view.php?id=1381> (Дата обращения: 21.03.2022)
7. А. Лааксонен, Олимпиадное программирование. Изучение и улучшение алгоритмов на соревнованиях. – ДМК Пресс. 2020.

8. Великович Л.С., Цветкова М.С. Программирование для начинающих. – М. Бином. Лаборатория знаний, 2007. – 287 с.
9. Волченков С.Г., Корнилов П.А., Белов Ю.А. и др. Ярославские олимпиады по информатике. Сборник задач с решениями. – М. Бином. Лаборатория знаний, 2010. – 405 с.
10. Златопольский Д.М. Программирование: типовые задачи, алгоритмы, методы. – М.: Бином. Лаборатория знаний, 2007. – 223 с.
11. Кирюхин В.М., Цветкова М.С. Информатика. Программы внеурочной деятельности учащихся по подготовке к Всероссийской олимпиаде школьников: 5 – 11 классы М.: Бином. Лаборатория знаний, 2014.

#### **Дополнительная литература:**

1. Лутц М. Изучаем Python. Учебное пособие. – СПб.: Символ-Плюс, 2009.
2. Жуков Р.А. Язык программирования Python. Практикум. Учебное пособие. – М.: ИНФРА-М, 2019.
3. Златопольский Д. Основы программирования на языке Python. – М.: ДМК Пресс, 2017.
4. Седжвик Роберт, Уэйн Кевин, Дондеро Роберт. Программирование на языке Python. Учебный курс. – М.: Диалектика, 2017.
5. Доусон М. Программируем на Python. – СПб.: Питер, 2015.
6. Программирование на Python: иллюстрированное руководство для детей / К. Квигли, М. Гудфеллоу, Д. Маккафферти, Дж. Вудкок. – М.: Манн, Иванов и Фербер, 2018.
7. МакГрат Майк. Программирование на Python для начинающих. – М.: Эксмо, 2015
8. Мюллер Джон Пол Python для чайников. – СПб: ООО «Диалектика», 2019.
9. Гэддис Т. Начинаем программировать на Python. – СПб: БХВ-Петербург, 2019.
10. Васильев А. Программирование на Python в примерах и задачах. – М.: Эксмо, 2021.

11. Федоров Д.Ю. Основы программирования на примере языка Python. – СПб: СПбГЭУ, 2018.
12. Фримен Эрик. Учимся программировать с примерами на Python. – СПб.: Диалектика, 2020.
13. Щерба А.В. Программирование на Python. Первые шаги. – М.: Лаборатория знаний, 2022.
14. Информатика. 8-9 классы. Начала программирования на языке Python / Босова Л.Л., Аквилянов Н.А., Кочергин И.О., Штепа Ю.Л., Бурцева Т.А.. Дополнительные главы к учебникам. – М.: БИНОМ. Лаборатория знаний, 2020.
15. Саммерфилд М. Программирование на Python 3. Подробное руководство. – СПб.: Символ-Плюс, 2009.

**Приложение 1.****Критерии отбора обучающихся на программу  
«Программирование на Python. Решение нестандартных задач»**

Отборочное задание для открытого набора обучающихся

1. Исполнитель Кузнечик «живет» на числовой прямой. В начальный момент времени он находится в точке 0. Кузнечик умеет совершать прыжки вперед на 3 деления вперед и на 2 деления вперед. В какую точку числовой оси никогда не сможет попасть Кузнечик?

1. 1. 1

2. 2. 2

3. 3. 4

4 4. 11

2. Вася, Денис, Коля и Толя увлечены спиннерами. Один из них умеет делать 8 разных трюков со спиннерами, другие по 5, 1 и 2 трюков. Каждый из них имеет только один любимый спиннер, с которым не расстанется. А вот количество лепестков у их спиннеров разное: есть спиннеры с двумя, тремя, четырьмя и шестью лепестками. У большинства из них спиннеры черного цвета и только один мальчик имеет спиннер другого цвета. Известно так же, что:

а) Толя имеет спиннер с количеством лепестков в 2 раза меньше, чем у того, кто умеет делать на 3 трюка больше, чем он.

б) Вася умеет делать в 2 раза меньше трюков, чем количество лепестков его спиннера, а Коляно количество трюков, напротив, в 2 раза больше числа лепестков его спиннера.

в) Денис имеет спиннер с числом лепестков в 2 раза больше, чем у Толи и на два лепестка больше, чем у самого «крутого» мастера трюков.

г) У всех черных спиннеров четное число лепестков.

Кто из ребят умеет делать ровно пять трюков со спиннерами?

1. 1. Толя

2. 2. Коля

3. 3. Денис

4. 4. Вася

3. Кот Матроскин и пес Шарик нашли клад, который состоял из 9 одинаковых монет. В коробке, в которой лежали монеты, друзья обнаружили записку: «При помощи чашечных весов без гирь найдите среди этих 9 монет одну золотую и купите почтальону Печкину велосипед. Сделайте это при помощи двух взвешиваний. Золотая монета более тяжелая.» Дядя Федор помог своим друзьям справиться с этим заданием. По сколько монет дядя Федор положил на чаши весов для первого взвешивания?

1. 2 и 2

2. 3 и 3

3. 3 и 4

4. 4 и 4

4. Двое играют в такую игру: первый называет однозначное число (т.е. целое число от 1 до 9 включительно), второй прибавляет к нему еще какое-нибудь однозначное число и называет сумму, к этой сумме первый прибавляет еще какое-нибудь однозначное число и опять называет сумму и так далее. Проигрывает тот, кто первым назовет число больше либо равное 66. Раскроем маленькую тайну: при правильной игре всегда выигрывает первый игрок. Он для того, чтобы выиграть, в каждый свой ход должен называть числа из некоторой последовательности. Определите эти числа.

1. 1, 10, 19, 28, 37, 46, 55, 65

2. 5, 15, 25, 35, 45, 55, 65

3. 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65

4. 6, 16, 26, 36, 46, 56, 66

5. Двое программистов играют в игру. У них имеется куча из  $N$  компьютерных мышей. Каждый игрок в свой ход может взять одну, две, три

или четыре мыши. Пропускать ход нельзя. Проигрывает игрок, который не сможет сделать очередной ход. При каком значении  $N$  имеется выигрышная стратегия для игрока, который начинает игру вторым.

1.  $N = 15$
2.  $N = 16$
3.  $N = 17$
4.  $N = 18$

6. В распоряжении продавца имеются четыре гири весом 1, 3, 9 и 27 кг. Определите, какой вес НЕ сможет отмерить продавец с помощью данного набора гирь при помощи рычажных весов, если гири можно ставить как на левую, так и на правую чашку весов, а также на обе чашки одновременно.

1. 16
2. 34
3. 42
4. 24

7. Последовательность чисел «Посмотри-и-скажи»

1, 11, 12, 1121, 1321, 122131, 132231, ... строится следующим образом.

Первый член последовательности  $A_1$  состоит из одной единицы.

$A_1: 1$

На каждом очередном шаге считаем сколько каких цифр встречается в предыдущем элементе последовательности. Цифры перебираем в порядке возрастания. Для изначальной последовательности имеем: цифра 1 встретила 1 раз. Очередной член последовательности строится так: для каждой из встретившихся цифр записываем цифру и, сразу же после, число её повторений. Таким образом, второй член последовательности  $A_2$  будет состоять из двух единиц.

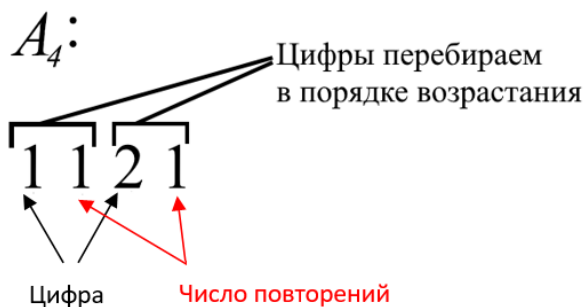
$A_2: 11$  (единица – один раз)

Теперь имеем, цифра 1 встретила 2 раза.

$A_3: 12$  (единица – два раза)



Подробная схема формирования очередного члена последовательности на примере  $A_4$  (рис.):



$A_4$ : 1121 (единица – 1 раз, двойка – 1 раз)

$A_5$ : 1321 (единица – 3 раза, двойка – 1 раз)

$A_6$ : 122131 (единица – 2 раза, двойка – 1 раз, тройка – 1 раз)

$A_7$ : 132231 (единица – 3 раза, двойка – 2 раза, тройка – 1 раз)

Продолжите данную последовательность самостоятельно.

Найдите элемент последовательности, в котором впервые встретится цифра 4.

8. Последовательность чисел «Посмотри-и-скажи»

1, 11, 12, 1121, 1321, 122131, 132231, ...

строится следующим образом. Первый член последовательности  $A_1$  состоит из одной единицы.

$A_1$ : 1

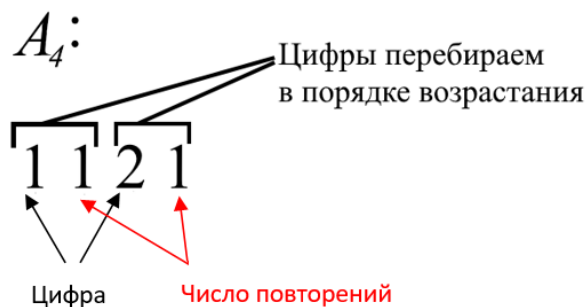
На каждом очередном шаге считаем сколько каких цифр встречается в предыдущем элементе последовательности. Цифры перебираем в порядке возрастания. Для изначальной последовательности имеем: цифра 1 встретила 1 раз. Очередной член последовательности строится так: для каждой из встретившихся цифр записываем цифру и, сразу же после, число её повторений. Таким образом, второй член последовательности  $A_2$  будет состоять из двух единиц.

$A_2$ : 11 (единица – один раз)

Теперь имеем, цифра 1 встретила 2 раза.

$A_3$ : 12 (единица – два раза)

Подробная схема формирования очередного члена последовательности на примере  $A_4$  (рис.5):



$A_4$ : 1121 (единица – 1 раз, двойка – 1 раз)

$A_5$ : 1321 (единица – 3 раза, двойка – 1 раз)

$A_6$ : 122131 (единица – 2 раза, двойка – 1 раз, тройка – 1 раз)

$A_7$ : 132231 (единица – 3 раза, двойка – 2 раза, тройка – 1 раз)

Продолжите данную последовательность самостоятельно.

Определите номер элемента последовательности, после которого уже больше не будут появляться новые члены последовательности, а будут повторяться ранее встретившиеся.

9. Среди пассажиров, едущих в вагоне, шло оживленное обсуждение пяти журналов. Оказалось, что каждый выписывает два журнала, причем каждая из возможных комбинаций двух журналов выписывается одним человеком. Сколько человек было в этой группе?

10. Ваня в первый день работы курьером заработал 4 рубля. Но он не отчаялся и продолжил работать. Во второй день он заработал в 4 раза больше, а именно – 16 рублей. Потом он подумал: «А что будет, если в третий день я заработаю в 16 раз больше, чем в предыдущий. И дальше каждый день буду зарабатывать в  $X$  раз больше, чем в предыдущий, где  $X$  – это заработок предыдущего дня.» Предположим, что Ванина стратегия осуществилась. Сколько Ваня заработает в 5 день работы?

В ответе укажите только число.

12. В параллели 9-х классов обучается 190 человек. Из них 24 человека приняли участие в олимпиаде по английскому языку, 37 – в

олимпиаде по русскому языку, 76 – в олимпиаде по математике. При этом в олимпиаде по русскому языку и в олимпиаде по математике одновременно участвовало 13 человек, в олимпиаде по русскому и английскому – 8 человек, по английскому и по математике – 12 человек. Пятеро учеников умудрились поучаствовать во всех трех олимпиадах. Сколько учеников не участвовало ни в одной олимпиаде?

1

2

.

В пароле, являющимся семизначным натуральным числом, пропали («стерлись») некоторые цифры. Известно, какие цифры стояли в утраченных позициях. Нужно определить, сколько вариантов пароля можно получить, подставляя эти цифры в утраченные позиции. Петя Васечкин написал программу, определения количества возможных паролей по утраченным цифрам. Его друг Вася Петечкин, помогал Пете тестировать эту программу. Ниже дана таблица, в которой находится протокол тестирования программы.

Вася утверждает, что программа выдает верный ответ (величину количества вариантов пароля) на один тест (и он прав!). Найдите и запишите в качестве ответа номер теста, для которого программа выдает неверный ответ.

Номер теста	Программа выдает верный ответ (или нет)	Количество вариантов пароля
1	нет	2
2	да	313
3	да	24
4	да	10
5	да	10

13. Исполнитель Кот, задан описанием:

Среда исполнителя – бесконечное множество пещер подземного царства, в котором каждая пещера связана тоннелями с 4-мя соседними пещерами, находящимися строго на севере, востоке, юге и западе, тоннели прямолинейны и длина каждого тоннеля ровно 10 метров. Кот находится внутри одной из пещер и может переходить из одной пещеры в другую. В

пещерах хранятся сокровища. Кот может собирать сокровища в свой рюкзак и обладает волшебством, которое помогает ему поднять, какой бы то ни было, самый тяжелый рюкзак.

### Система команд исполнителя:

**ВПЕРЕД** – перемещение исполнителя на 10 метров вперед в том направлении, куда он смотрит (переход в соседнюю пещеру для текущего направления);

**ПОВОРОТ** – поворот исполнителя на  $90^0$  по часовой стрелке;

**ВЗЯТЬ** – сбор всех драгоценностей в пещере в свой рюкзак.

**ПОВТОРИ** **<k>** {тело цикла} – конструкция цикла с параметром, тело цикла выполняется **k** раз;

**ЕСЛИ** **<условие>** **ТО** {команды1} **ИНАЧЕ** {команды2} – конструкция ветвления, ветвь **ИНАЧЕ** может отсутствовать;

Фигурные скобки { } обязательны и означают начало и конец серии команд, выполняемых в цикле или в данной ветви оператора ветвления.

**<условие>** может принимать значения:

***ПУСТО*** (условие будет истинным, если в пещере, куда переместился Кот, не было драгоценностей, и ложным в противном случае);

***НЕ ПУСТО*** (противоположно условию ***ПУСТО***).

На рисунке показано местоположение Кота, он смотрит на юг. «Звездочками» отмечены пещеры с сокровищами, в каждой из них по одному сокровищу.



Сколько сокровищ сможет собрать Кот после выполнения программы:

**ПОВТОРИ 5**

```

{ ПОВОРОТ
  ВПЕРЕД
  ЕСЛИ НЕ ПУСТО ТО { ВЗЯТЬ ПОВОРОТ ВПЕРЕД }
}

```

В ответе укажите только число.

14. Исполнитель Кот, задан описанием:

Среда исполнителя – бесконечное множество пещер подземного царства, в котором каждая пещера связана тоннелями с 4-мя соседними пещерами, находящимися строго на севере, востоке, юге и западе, тоннели прямолинейны и длина каждого тоннеля ровно 10 метров. Кот находится внутри одной из пещер и может переходить из одной пещеры в другую. В пещерах хранятся сокровища. Кот может собирать сокровища в свой рюкзак и обладает волшебством, которое помогает ему поднять, какой бы то ни было, самый тяжелый рюкзак.

Система команд исполнителя:

**ВПЕРЕД** – перемещение исполнителя на 10 метров вперед в том направлении, куда он смотрит (переход в соседнюю пещеру для текущего направления);

**ПОВОРОТ** – поворот исполнителя на  $90^0$  по часовой стрелке;

**ВЗЯТЬ** – сбор всех драгоценностей в пещере в свой рюкзак.

**ПОВТОРИ**  $\langle k \rangle$  {тело цикла} – конструкция цикла с параметром, тело цикла выполняется  $k$  раз;

**ЕСЛИ**  $\langle \text{условие} \rangle$  **ТО** {команды1} **ИНАЧЕ** {команды2} – конструкция ветвления, ветвь **ИНАЧЕ** может отсутствовать;

Фигурные скобки { } обязательны и означают начало и конец серии команд, выполняемых в цикле или в данной ветви оператора ветвления.

$\langle \text{условие} \rangle$  может принимать значения:

**ПУСТО** (условие будет истинным, если в пещере, куда переместился Кот, не было драгоценностей, и ложным в противном случае);

**НЕ ПУСТО** (противоположно условию **ПУСТО**).

ПРОЦ <имя> – заголовок описания вспомогательного алгоритма.

КОНЕЦ ПРОЦ – конец описания вспомогательного алгоритма.

*Обращение к вспомогательному алгоритму* происходит путем указания его имени в основной программе.

В скольких пещерах побывает Кот после выполнения команд. Та пещера, в которой он находится в начальный момент тоже считается.

ПРОЦ Работа

ПОВТОРИ 2

```
{ ПОВТОРИ 2 { ВПЕРЕД }
  ПОВТОРИ 2 { ПОВОРОТ }
}
```

КОНЕЦ ПРОЦ

ПОВТОРИ 4 РАЗ

```
{ Работа
  ПОВОРОТ
}
```

В ответе укажите только число.

15. Исполнитель Кот, задан описанием:

Среда исполнителя – бесконечное множество пещер подземного царства, в котором каждая пещера связана тоннелями с 4-мя соседними пещерами, находящимися строго на севере, востоке, юге и западе, тоннели прямолинейны и длина каждого тоннеля ровно 10 метров. Кот находится внутри одной из пещер и может переходить из одной пещеры в другую. В пещерах хранятся сокровища. Кот может собирать сокровища в свой рюкзак и обладает волшебством, которое помогает ему поднять, какой бы то ни было, самый тяжелый рюкзак.

Система команд исполнителя:

ВПЕРЕД – перемещение исполнителя на 10 метров вперед в том направлении, куда он смотрит (переход в соседнюю пещеру для текущего направления);

**ПОВОРОТ** – поворот исполнителя на  $90^0$  по часовой стрелке;

**ВЗЯТЬ** – сбор всех драгоценностей в пещере в свой рюкзак, если драгоценностей в пещере не окажется, ошибки не возникнет

**ПОВТОРИ**  $\langle k \rangle$  {тело цикла} – конструкция цикла с параметром, тело цикла выполняется  $k$  раз;

**ЕСЛИ**  $\langle \text{условие} \rangle$  **ТО** {команды1} **ИНАЧЕ** {команды2} – конструкция ветвления, ветвь **ИНАЧЕ** может отсутствовать;

Фигурные скобки { } обязательны и означают начало и конец серии команд, выполняемых в цикле или в данной ветви оператора ветвления.

$\langle \text{условие} \rangle$  может принимать значения:

**ПУСТО** (условие будет истинным, если в пещере, куда переместился Кот, не было драгоценностей, и ложным в противном случае);

**НЕ ПУСТО** (противоположно условию **ПУСТО**).

**ПРОЦ**  $\langle \text{имя} \rangle$  – заголовок описания вспомогательного алгоритма.

**КОНЕЦ ПРОЦ** – конец описания вспомогательного алгоритма.

*Обращение к вспомогательному алгоритму* происходит путем указания его имени в основной программе.

Сколько сокровищ соберет Кот при выполнении алгоритма Икс, если во всех пещерах подземного царства есть по одному сокровищу.

**ПРОЦ** Икс

**ПОВОРОТ**

**ВПЕРЕД**

**ВЗЯТЬ**

**ВПЕРЕД**

**ЕСЛИ НЕ ПУСТО ТО** { **ВЗЯТЬ** Икс }

**КОНЕЦ ПРОЦ**

В ответе укажите только число.

16. В семье на компьютере работают 4 человека: папа, мама, сын и дочь. Папа заполняет на диске 1 Гб в месяц, мама 1 Гб в два месяца, сын 1 Гб в четыре месяца, дочь – 1 Гб в восемь месяцев. Каждый из членов семьи

работает за компьютером ежедневно и тратит при этом одинаковое количество памяти жесткого диска каждый день. Через сколько месяцев жесткий диск размером 30 Гб в этой семье заполнится полностью?

В ответе укажите только число.

17. Курьер, работающий в некотором банке, находясь в его центральном офисе, должен на велосипеде объехать все офисы банка, побывав в каждом офисе ровно один раз. Между любыми двумя офисами существует путь проезда. Курьер хочет выбрать самый экономный и быстрый маршрут. Сколько вариантов маршрутов он вынужден будет проанализировать, чтобы сделать абсолютно правильный выбор. Общее число офисов банка равно 5.

18. Ваня и Яна закончили первый класс и очень любят математику, поэтому на каникулах играют в игры с четырехзначными числами. Сегодня их интересуют четырехзначные числа, состоящие только из четных цифр. Дополнительное условие: сумма цифр числа должна быть кратна 10. Ваня ищет такие числа, которые как можно больше. А Яна все такие числа, которые как можно меньше. А вы сможете найти самое большое такое число? А самое маленькое? В ответе запишите разность между самым большим таким числом и самым маленьким.

19. Имеется 3 ведра воды (A, B, C) следующей емкости: ведро A – 10 литров, ведро B – 5 литров, ведро C – 3 литра. В начальный момент времени ведро A полностью заполнено водой, а ведра B и C пусты. Можно переливать воду из одного ведра в другое, но в результате переливания должно выполняться хотя бы одно из двух условий:

- 1) одно из ведер, участвовавших в переливании, становится пустым;
- 2) одно из ведер, участвовавших в переливании, становится полным.

Сколько переливаний минимально необходимо, чтобы получить в ведре A ровно 6 литров воды?

20. Настя очень любит печь печенье в виде геометрических фигур. Сегодня она выбрала форму шестиугольника. Своему брату Васе Настя поручила складывать из готового печенья пирамиду, чтобы порадовать



родителей не только вкусным печеньем, но и сервировкой. Вася решил, что в его пирамиде из печенья будет  $n$  слоев (уровней). Для самого верхнего слоя (уровня) нужна ровно одна печенье. Если слоев два, то нижний слой составляется так, чтобы центр в виде одной печеньки был окружен печеньками по кругу которые полностью соприкасаются боковыми гранями. Аналогично выкладывается третий слой – его «радиус» на 1 печенье больше. Настя заводит тесто и просит брата сказать ей, сколько печенья ему понадобится для его пирамиды. Помогите Васе посчитать, сколько печенья будет в пирамиде из 5 слоев (уровней).

