

# Вопросы использования возможностей искусственного интеллекта в работе преподавателя информатики

---

ФИЛИППОВ ВЛАДИМИР ИЛЬИЧ,

К.П.Н., ДОЦЕНТ КАФЕДРЫ ОБЩЕОБРАЗОВАТЕЛЬНЫХ ДИСЦИПЛИН

ГАОУ ДПО МО «КОРПОРАТИВНЫЙ УНИВЕРСИТЕТ РАЗВИТИЯ ОБРАЗОВАНИЯ»,

СМОЛЬНЯКОВ ВЯЧЕСЛАВ ГЕННАДЬЕВИЧ,

МАОУ СОШ №17 С УИОП ГОЩ

# Понятие искусственного интеллекта

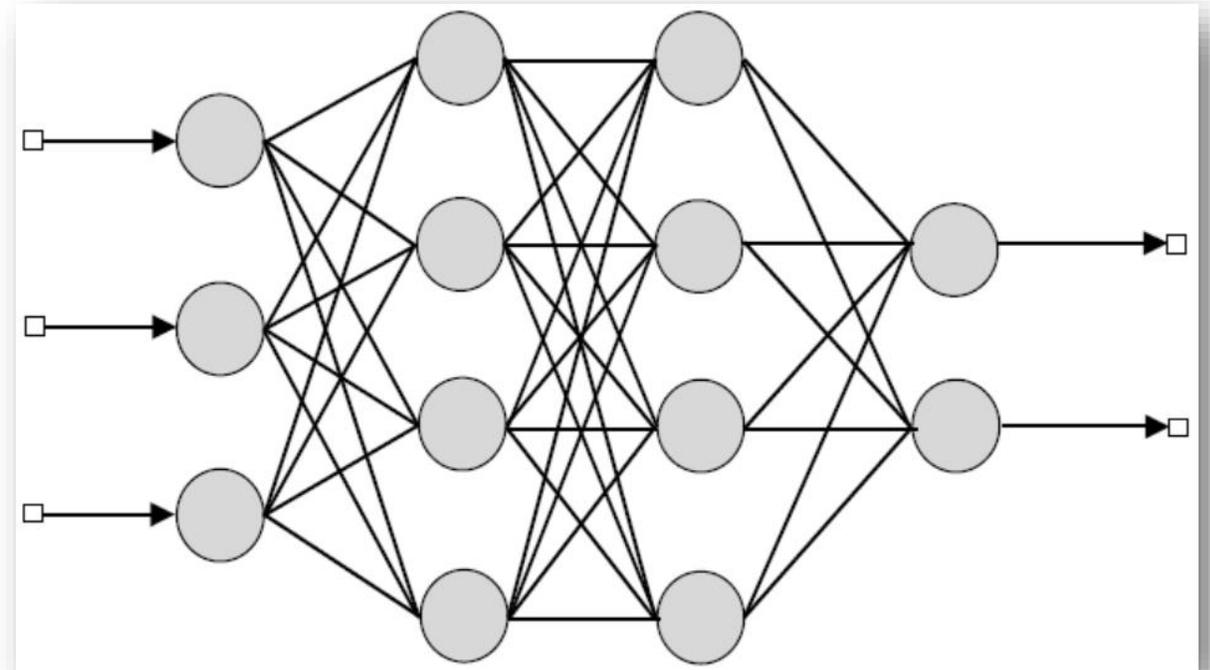
**Искусственный интеллект (ИИ)** - это область компьютерных наук, которая занимается разработкой и созданием систем и программ, способных выполнять задачи, требующие интеллектуальных способностей человека.

ИИ стремится моделировать и эмулировать различные аспекты человеческого мышления, такие как распознавание образов, обучение, планирование, принятие решений, общение и многое другое.



# Нейросети

**Нейросеть** - это компьютерная модель, которая имитирует работу нервной системы человека. Она состоит из множества взаимосвязанных искусственных нейронов, которые обрабатывают информацию и передают ее по сети.



# История возникновения ИИ

## 1940-1950 гг.

Исследования Алана Тьюринга, Джона фон Неймана, Норберта Винера. Первые модели вычислительных машин и идеи о том, как машины могут имитировать человеческое мышление.

## 1960-1970 гг.

Разработаны символьные методы ИИ. Появились компьютерные программы, способные решать задачи с помощью логики и экспертных знаний.

## 2010-н.в.

Нейронные сети и глубокое обучение стали основными технологиями ИИ. Они обрабатывают сложные данные, распознают и генерируют изображения, общаются на естественном языке.



В 1956 году на конференции Дартмутского колледжа впервые использовали термин «искусственный интеллект».

## 1980-1990 гг.

Разработаны алгоритмы обучения с подкреплением. ИИ может самостоятельно изучать и улучшать свои навыки, опираясь на обратную связь от окружающей среды.

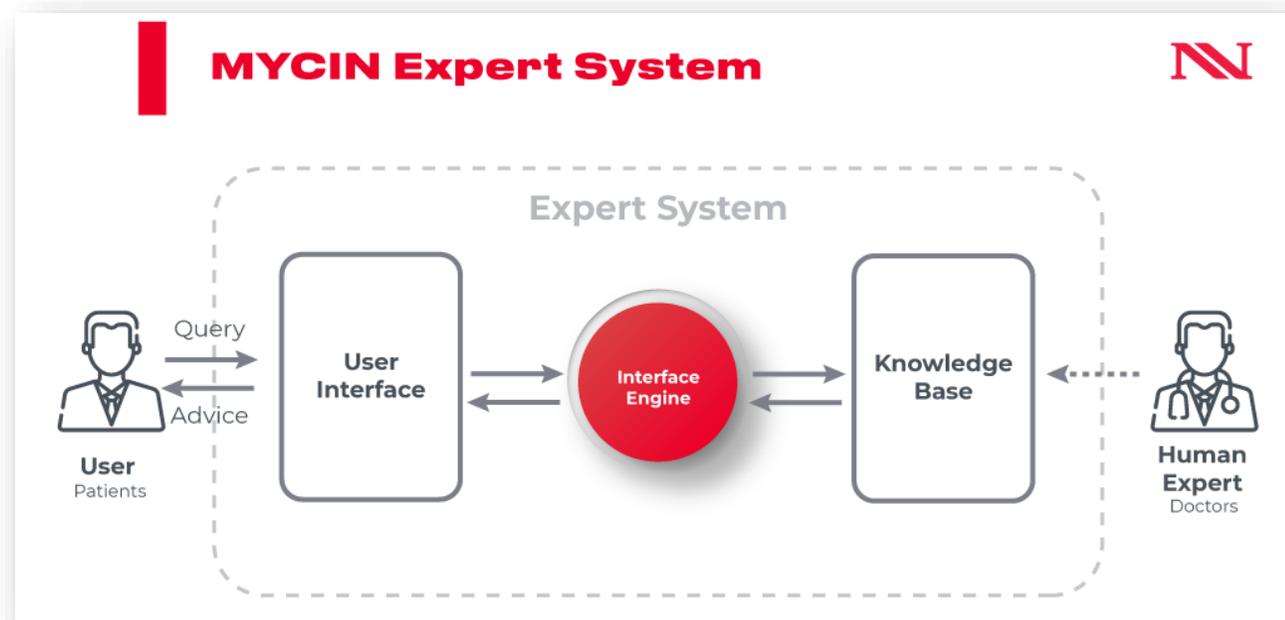
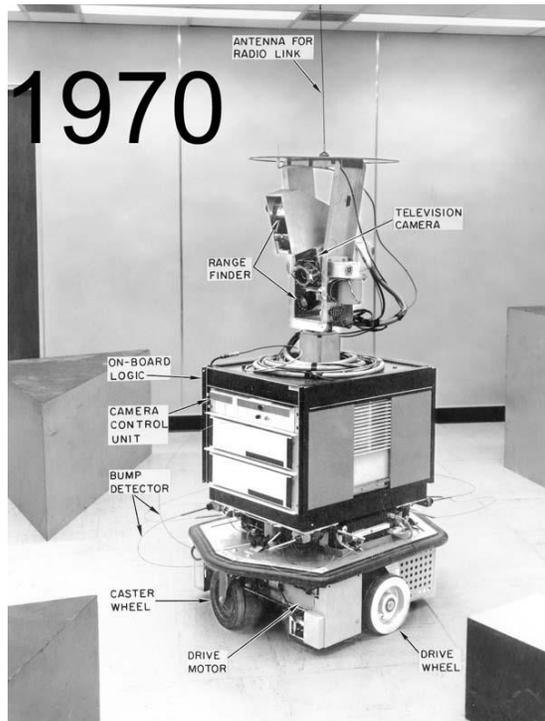
# История возникновения ИИ

В середине 1960-х **Джозеф Вайценбаум** создал **ELIZA** — первого чат-бота, который имитировал работу психотерапевта и мог общаться с человеком на естественном языке

```
=====
EEEEEEEE L      IIIIII ZZZZZZZ      AAA
E         L      I          Z
E         L      I          Z
EEEEEE   L      I          Z
E         L      I          Z
E         L      I          Z
EEEEEEEE LLLLLLL IIIIII ZZZZZZ      A
                                     A
                                     A
                                     A
                                     A
                                     A
                                     A
=====
ELIZA > (PLEASE TYPE IN ALL CAPS) WHAT'S YOUR NAME DEAR ?
AMIT
ELIZA > HI, I'M ELIZA. WHAT DO YOU WANT TO TALK ABOUT ?
AMIT > CAN I TALK ABOUT MY PROBLEM ?
ELIZA > SURE... !
AMIT > DUE TO ON GOING EXAMS IN COLLEGE, I HAVE A LOT OF STRESS.
ELIZA > PLEASE RELAX A LITTLE. SLEEP WELL.
AMIT > THANKS FOR YOUR ADVICE.
ELIZA > NO MENTION.
AMIT > BYE
ELIZA > BYE AND KEEP IN TOUCH...
=====
```

# История возникновения ИИ

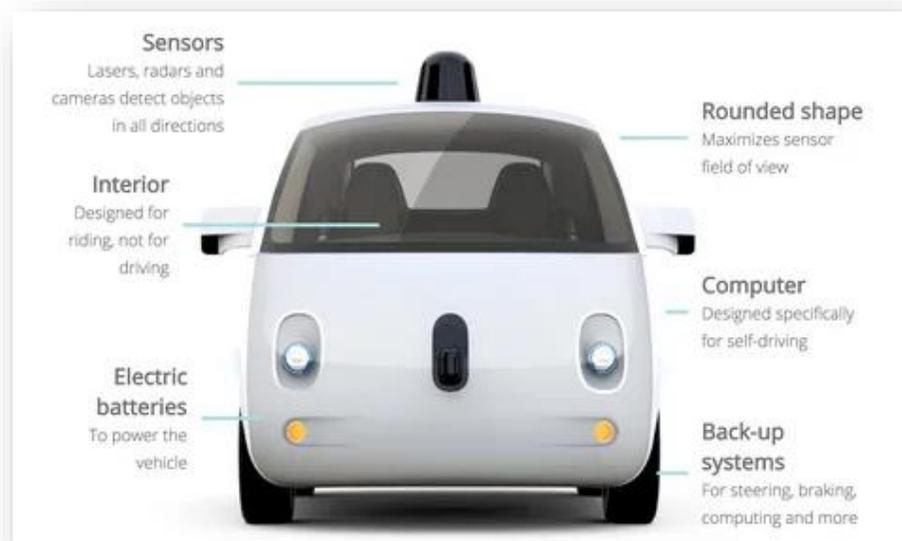
WABOT-1 — интеллектуального человекоподобного робота. Более продвинутые экспертные системы. Например, MYCIN могла диагностировать менингит и рассчитывать дозировку антибиотика для его лечения



# История возникновения ИИ

В 1997 году Deep Blue от IBM победила гроссмейстера Гарри Каспарова, действующего чемпиона мира по шахматам.

В 2009 году Google начала разрабатывать технологию самоуправляемых автомобилей.



# История возникновения ИИ

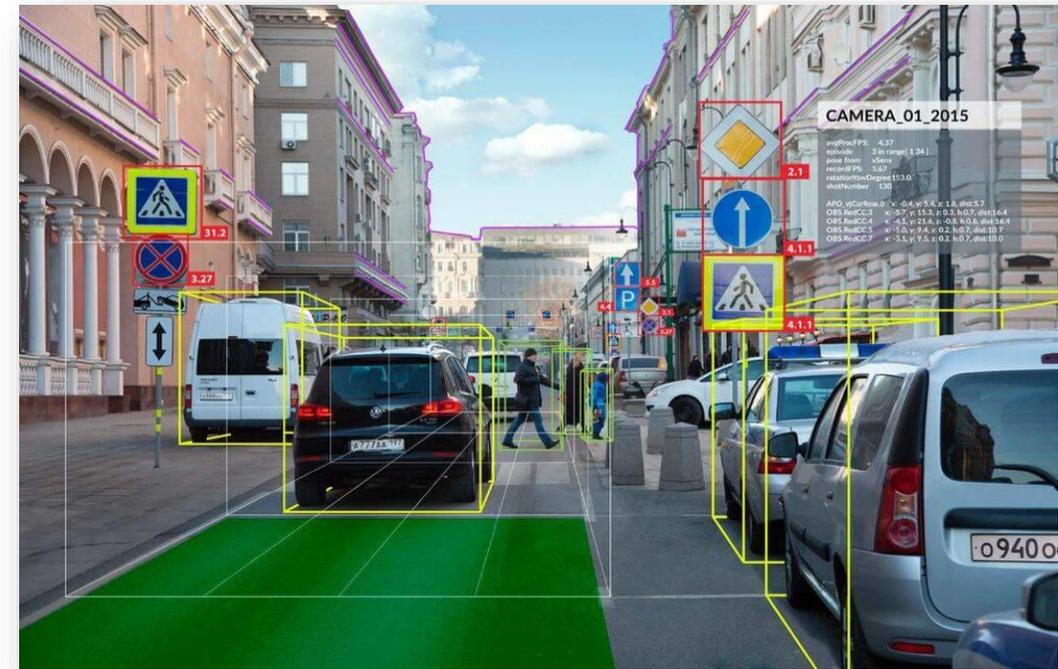
2016 год - появилась София — первый робот, который может менять выражение лица, видеть и разговаривать с помощью искусственного интеллекта.

2023 год — год прогресса для генеративных сетей (GAN), которые создают реалистичные изображения и видео, и больших языковых моделей (LLM), например ChatGPT.



# Сферы применения ИИ в современном мире

1. Голосовые помощники.
2. Рекомендательные системы.
3. Распознавание образов.
4. Автопилоты и автономные транспортные системы.



# Сферы применения ИИ в современном мире

1. Финансовые аналитические системы.
2. Языковые переводчики.
3. Игровая индустрия.
4. Медицинская диагностика.
5. Робототехника



## Перспективы развития

1. Генеративный ИИ может ежегодно приносить мировой экономике от **2,6 до 4,4 триллионов долларов** (от 2 до 4% совокупного мирового ВВП)
2. Повсеместное внедрение генеративного ИИ увеличит стоимость сектора IT от 4,8 до 9,3%.
3. Объём рынка банковского дела, образования, фармацевтики и телекоммуникаций может вырасти от 2 до 5%.

## Перспективы развития

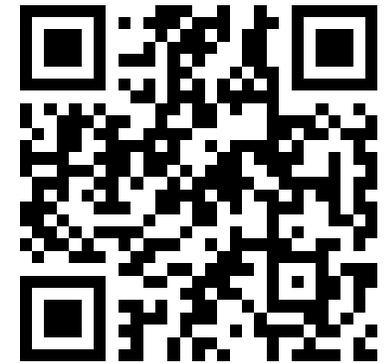
1. Генеративный ИИ, скорее всего, автоматизирует задачи на относительно высокооплачиваемых работах. Например, в разработке ПО и продуктов.
2. Генеративный ИИ может автоматизировать **50% всех рабочих задач в период с 2030 по 2060 год.**

**Неконтролируемое развитие ИИ может быть рискованным, но современные исследования и разработки направлены на создание безопасных и этичных систем искусственного интеллекта.**

# Использование возможностей ИИ педагогами

1. Применение ИИ при реализации проектов.
2. Использование возможностей ИИ при создании заданий и кейсов.

<https://t.me/GPT4Telegrambot>



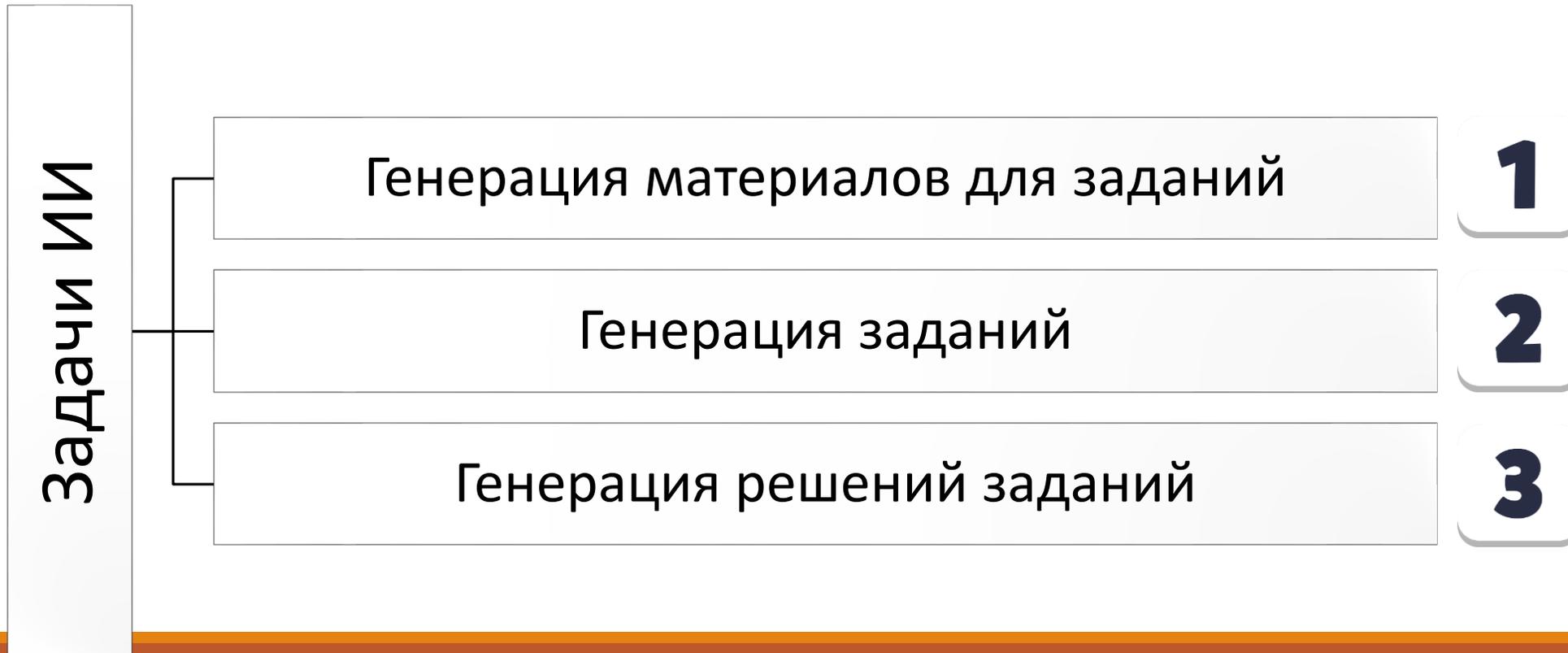
<https://www.sberbank.com/promo/kandinsky/>

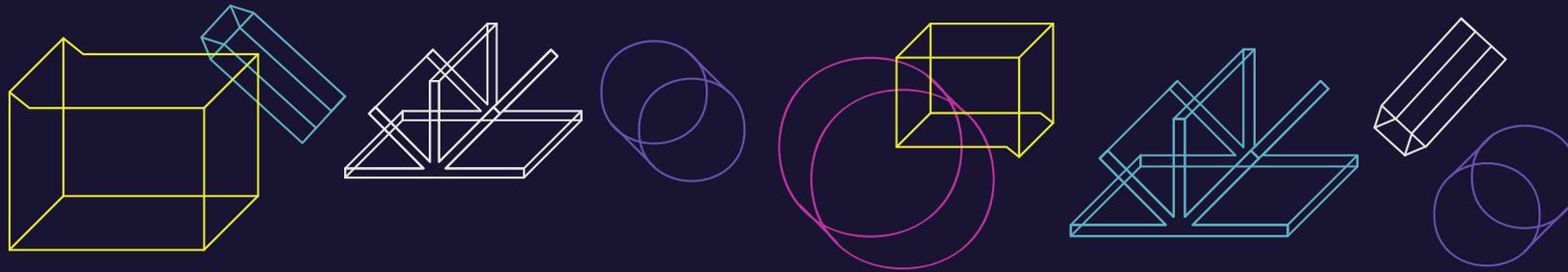


# ОПРЕДЕЛИМСЯ С ЦЕЛЬЮ ПРИМЕНЕНИЯ

**Использование ИИ целесообразно, если:**

- экономятся ресурсы при выполнении рутины (время педагога, труд)
- создаётся новый контент (когда нет или его недостаточно/лицензирован)
- контент актуализируется (когда имеющийся общедоступен)





# ГЕНЕРАЦИЯ МАТЕРИАЛОВ ДЛЯ ЗАДАНИЙ

Где взять уникальные данные для заданий?

# ГЕНЕРАЦИЯ МАТЕРИАЛОВ

1

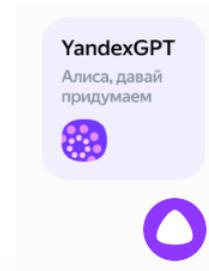


# ГЕНЕРАЦИЯ ТЕКСТОВ



## Помощник «Алиса» от Яндекс

- отечественный (не подвержен ограничениям)
- простая регистрация (нужен аккаунт в Яндексе)
- бесплатный



# ЯНДЕКС

## Какие задачи можно решить с YandexGPT

На этапе закрытого тестирования мы вместе с вами хотим понять, какие задачи можно решать наиболее эффективно и приоритезировать сценарии развития сервиса.



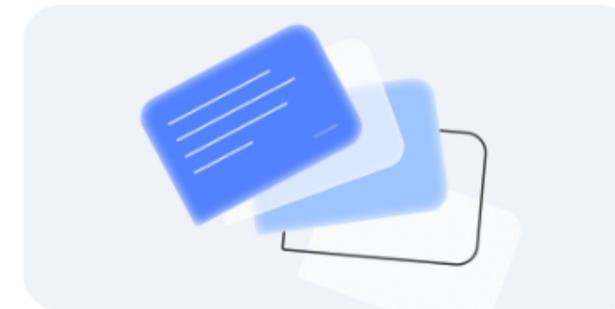
### Анализ текстовой информации

Получите агрегированный результат по накопленным текстовым данным: информации в рабочих чатах, отзывам пользователей и другим типам данных. YandexGPT поможет как обобщить, так и интерпретировать информацию.



### Создание контента

Ускорьте создание текстов, улучшайте их качество и стиль. Создавайте тексты по шаблону для информационных рассылок, описание товаров для интернет-магазина и других приложений.



### Чат-боты

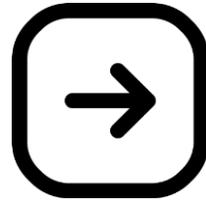
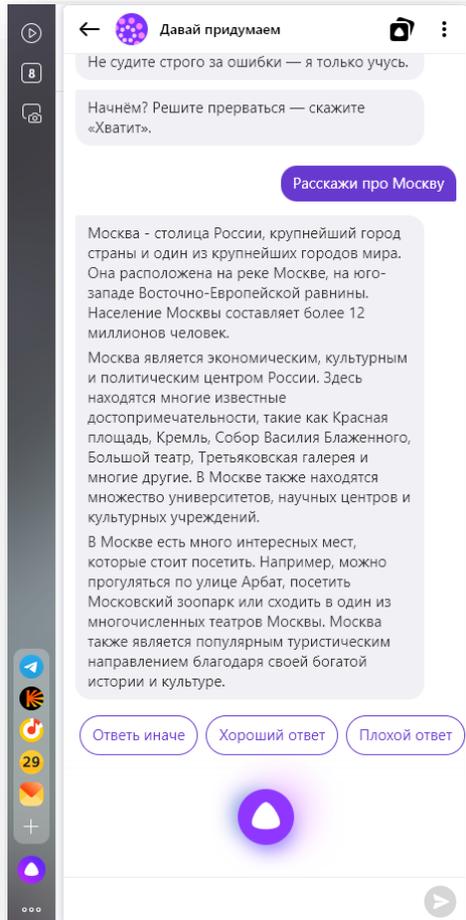
Развивайте чат-бота для службы поддержки: научите бота отвечать на разные вопросы пользователей как типовые, так и более сложные.

<https://browser.yandex.ru/alice/>

# ГЕНЕРАЦИЯ ТЕКСТОВ

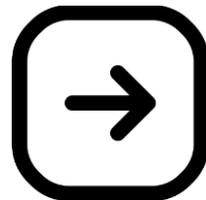


## Пример использования помощника «Алиса»



### 7 класс

Создание и форматирование текстовых документов, создание презентаций



### Подготовка к ОГЭ

Задание 13.1 (текстовая основа)

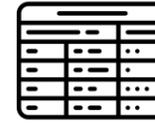
Задание 13.2



- Нет вопросов по авторскому праву
- Текст сгенерирован – его нет в поисковых выдачах, его трудно скопировать откуда-то и легко создать новый («Ответить иначе»)

## Навык «Давай придумаем»

# ГЕНЕРАЦИЯ ТАБЛИЦ



**К сожалению, готовых решений нет. Но...**  
существуют онлайн-генераторы таблиц (бесплатные)

<https://codezi.pro/excel-data-generator>

Enter the number of records you want to create and define the data fields

How many records do you want to generate?

50

Define columns, Key

Data Type	Property Name	Blank (%)	
Company name, (eg: Ortiz Inc) <input type="text"/>	Компания <input type="text"/>	0 <input type="text"/>	
Country, (eg: Madagascar) <input type="text"/>	Страна <input type="text"/>	0 <input type="text"/>	<input type="button" value="🗑️"/>
Integer, (eg: 46126543) <input type="text"/>	Запас ресурса <input type="text"/>	0 <input type="text"/>	<input type="button" value="🗑️"/>
Date (y-m-d), (eg: 2007-07-11) <input type="text"/>	Дата основания <input type="text"/>	0 <input type="text"/>	<input type="button" value="🗑️"/>

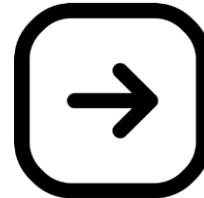
[+ Add More Field](#)

Generate and Preview as Json

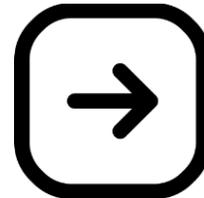
# ГЕНЕРАЦИЯ ТАБЛИЦ



Компания	Страна	Запас ресурса	Дата основания
Rohan-Reichert	Holy See (Vatican City State)	752349	1972-09-08
Becker, Crooks and Turner	Mayotte	1	1973-06-12
Schmeler and Sons	Syrian Arab Republic	86491	1984-03-06
Kshlerin-Beier	Puerto Rico	580	1979-02-28
Corwin-Kohler	Morocco	163	1990-03-25
Bosco Inc	Saint Kitts and Nevis	232097	1994-02-16
Ernser, Hoppe and Strosin	Croatia	625892	1994-10-14
Heathcote, Cremin and Mertz	Lesotho	76	2020-12-05
Predovic PLC	Korea	1521	1993-07-10
Cronin and Sons	Guatemala	426756	1974-03-06
Gleichner-Walsh	China	94178	1990-07-26
Gutkowski Inc	Samoa	392	1979-05-18
Jacobi-Botsford	Cote d'Ivoire	61	1986-03-20
Dach, Tremblay and Gleichner	Tanzania	788	1983-06-14
Hoeger LLC	French Southern Territories	8	1974-11-04
Heaney, Ankunding and Weber	South Africa	2611984	2013-02-06
Gerlach Group	Iceland	675	2010-10-05
Okuneva, Leffler and Brekke	United States Virgin Islands	16357	1996-03-12
Lowe-Dietrich	Luxembourg	41787	2009-02-03
Bogan-Brakus	Saint Pierre and Miquelon	812124	2004-05-05
Homenick-Nolan	Philippines	381544530	2017-01-26
McLaughlin and Sons	Cook Islands	8374175	1989-10-20
Fay-Ledner	United States Minor Outlying Islands	49818765	1999-11-02
Maggio, Langosh and Pollich	American Samoa	46588592	1983-03-14
Lebsack and Sons	Timor-Leste	92	2019-10-08
Schoen, Rempel and Hirthe	Malawi	9766823	1999-11-15
Kunde-Metz	Indonesia	71468803	1972-06-30
Donnelly, Conroy and Schamberger	Brazil	75897009	1992-02-11
Hermiston, O'Keefe and Brekke	Marshall Islands	94511	1981-10-04
Hand PLC	Guadeloupe	60	1978-02-14
West-Nikolaus	Latvia	8717284	1979-04-15
Braun-Runolfsson	Andorra	195009479	2011-11-12
Heller Group	Oman	11	1990-06-06
Schuppe LLC	Ireland	2066271	2001-08-18
Hirsh, Luskwitz and Muller	Guinea	69518596	2006-06-27



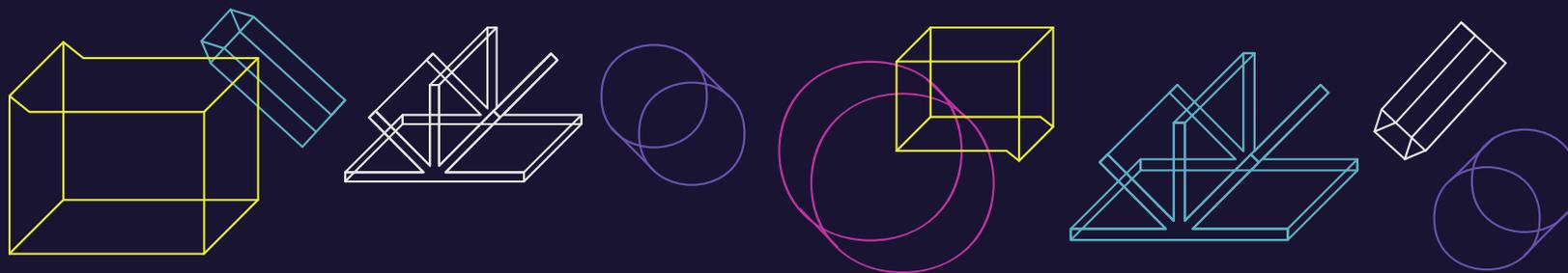
**9 класс**  
Обработка таблиц



**Подготовка к ОГЭ**  
Задание 14



- Нет вопросов по авторскому праву
- Таблица сгенерирована (можно сгенерировать под любое задание)



# ГЕНЕРАЦИЯ ЗАДАНИЙ

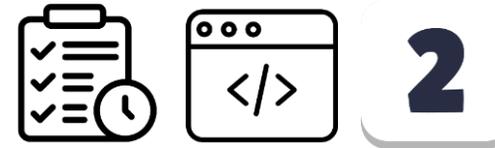
Как составить уникальное задание?

# ГЕНЕРАЦИЯ ЗАДАНИЙ

2

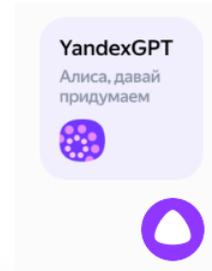


# ГЕНЕРАЦИЯ ЗАДАЧ (ОБА ТИПА)



## Помощник «Алиса» от Яндекс

- отечественный (не подвержен ограничениям)
- простая регистрация (нужен аккаунт в Яндексе)
- бесплатный



# ЯНДЕКС

## Какие задачи можно решить с YandexGPT

На этапе закрытого тестирования мы вместе с вами хотим понять, какие задачи можно решать наиболее эффективно и приоритезировать сценарии развития сервиса.



### Анализ текстовой информации

Получите агрегированный результат по накопленным текстовым данным: информации в рабочих чатах, отзывам пользователей и другим типам данных. YandexGPT поможет как обобщить, так и интерпретировать информацию.



### Создание контента

Ускоряйте создание текстов, улучшайте их качество и стиль. Создавайте тексты по шаблону для информационных рассылок, описание товаров для интернет-магазина и других приложений.



### Чат-боты

Развивайте чат-бота для службы поддержки: научите бота отвечать на разные вопросы пользователей как типовые, так и более сложные.

<https://browser.yandex.ru/alice/>

# ГЕНЕРАЦИЯ ТЕОРЕТИЧЕСКИХ ЗАДАЧ



2

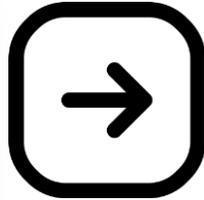
## Пример использования помощника «Алиса»

Придумай задачу на измерение количества информации

Определите, сколько информации содержится в сообщении, которое состоит из 10 символов, каждый из которых может принимать значения от 1 до 9.

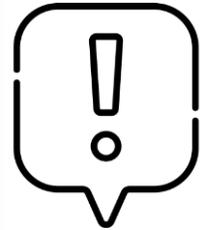
Ответь иначе

В книге 300 страниц. На каждой странице 40 строк по 80 символов. Сколько информации содержит книга?



7 класс

Измерение количества информации  
Комбинаторные задачи



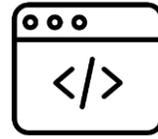
- Нет вопросов по авторскому праву
- Текст сгенерирован – его нет в поисковых выдачах, его трудно скопировать откуда-то и легко создать новый («Ответь иначе»)

Задача:

Есть 10 символов, которые могут использоваться для создания пароля, и каждый символ может быть использован только один раз. Сколько комбинаций из 8 символов можно составить из этих символов?

Навык «Давай придумаем»

# ГЕНЕРАЦИЯ ЗАДАЧ НА ПРОГРАММИРОВАНИЕ



2

## Пример использования помощника «Алиса»

**Придумай задачу на программирование ветвлений**

Задача: Напишите программу на языке Python, которая запрашивает у пользователя число и выводит на экран сообщение "Больше" или "Меньше" в зависимости от того, больше или меньше введенное число, чем 10.

Ответ иначе    Хороший ответ    Плохой ответ

**придумай задачу на обработку данных во время ввода**

Задача на обработку данных при вводе: Напишите программу, которая считывает данные из стандартного ввода и выводит их в обратном порядке. Например, если пользователь вводит "1 2 3 4 5", программа должна вывести "5 4 3 2 1".

**Придумай задачу на программирование циклов с условием**

Задача: Считать числа из файла и вывести их на экран, если они больше 10 или меньше 20.

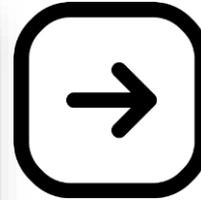
Ответ иначе

Задача заключается в том, чтобы найти сумму всех чисел в массиве, которые меньше 10 и больше 20, используя цикл for.

Ответ иначе

Задача на программировании циклов с условием заключается в том, чтобы вывести на экран все числа, которые делятся на 3 или 5, но не делятся на оба этих числа одновременно.

Ответ иначе    Хороший ответ    Плохой ответ



7-9 класс

Программирование  
Подготовка к ОГЭ



Задачи созданы  
нейросетью – не  
взяты из сборников,  
к которым есть  
ответы

# ГЕНЕРАЦИЯ ИЗОБРАЖЕНИЙ

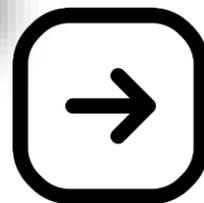


<https://www.sberbank.com/promo/kandinsky/>



## 7 класс

Обработка изображений  
Создание презентаций



## Подготовка к ОГЭ

Задание 13.1 (изображения)



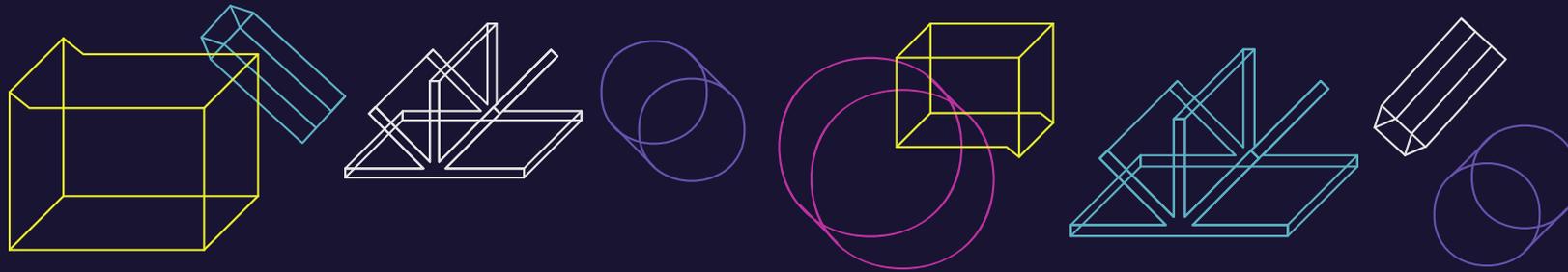
- Нет вопросов по авторскому праву
- Изображение сгенерировано (можно выдать задание с уникальным контентом)



## Нейросеть «Кандинский»

- отечественный (не подвержен ограничениям)
- не требует регистрации
- бесплатный

Генерация по запросу:  
«Фото Московский кремль»



# ГЕНЕРАЦИЯ РЕШЕНИЙ ЗАДАНИЙ

Разобраться в новом или просто составить задачу с ответом?

# ГЕНЕРАЦИЯ РЕШЕНИЙ ЗАДАНИЙ

3



# РЕШЕНИЕ ТЕОРЕТИЧЕСКИХ ЗАДАЧ



## Пример использования помощника «Алиса»

Придумай задачу на определение количества информации в растровом изображении

Для решения данной задачи вам необходимо определить количество информации, необходимое для кодирования пикселей изображения. Для этого вы можете использовать формулу:

$$I = K * \log_2(N)$$

где  $I$  - количество информации,  $K$  - количество возможных значений пикселя (например, для 8-битного изображения  $K = 256$ ),  $N$  - количество пикселей в изображении.

Например, пусть у вас есть изображение размером 1024x768 пикселей с 8-битными цветами. Тогда количество информации для кодирования каждого пикселя будет равно  $I = 8 \text{ бит} / 8 = 1 \text{ бит}$ . Таким образом, общее количество информации для изображения будет равно:

$$I = 1024 * 768 * 1 = 786432 \text{ бит}$$

Ответ: 786432 бита.

Задача:

Есть 5 символов, которые можно использовать для создания пароля. Какие комбинации из 4 символов могут быть созданы при условии, что каждый символ может использоваться только один раз?

Ответ:

Всего можно создать 125 комбинаций.

Ответь иначе

Задача:

Есть 10 символов, которые могут использоваться для создания пароля, и каждый символ может быть использован только один раз. Сколько комбинаций из 8 символов можно составить из этих символов?

Решение:

Для начала нужно выбрать 8 символов из 10. Это можно сделать, перемножив количество способов выбрать каждый символ:

$$10C8 = 10! / (8! * 2!) = 45$$

Затем нужно учесть, что каждый символ должен быть использован только один раз, поэтому мы можем перемножить количество способов выбора каждого символа с учетом этого ограничения:

$$(10 - 1)! / ((8 - 1)!) * 1 = 9! / 7! * 1$$

Таким образом, общее количество комбинаций равно:

$$45 * 9! / 7! = 2160$$

Ответ: 2160 комбинаций



Некоторые запросы условий задач уже содержат алгоритмы, поэтому рекомендуется предварительная подготовка текстов задач перед уроком

# РЕШЕНИЕ ТЕОРЕТИЧЕСКИХ ЗАДАЧ



3

## Пример использования помощника «Алиса»

Для определения количества информации в растровом изображении можно использовать алгоритм подсчета количества различных цветов (количество бит, необходимых для кодирования каждого цвета).

1. Определить размер изображения в пикселях.
2. Определить количество бит на пиксель. Это зависит от количества цветов, которые могут быть закодированы в каждом пикселе. Например, для 8-битного изображения каждый пиксель может содержать 256 различных цветов, что требует 8 бит на каждый пиксель.
3. Определить общее количество бит, необходимое для кодирования всех пикселей в изображении. Для этого нужно умножить количество пикселей на количество бит на один пиксель.

придумай алгоритм решения задачи на определение количества информации в растровом изображении

4. Разделить общее количество бит на 8 (количество бит в байте), чтобы получить количество байтов, необходимых для хранения изображения.
5. Умножить количество байтов на стоимость передачи одного байта, чтобы получить стоимость передачи изображения.
6. Прибавить к стоимости передачи изображения стоимость обработки изображения на компьютере, если это требуется для дальнейшего использования изображения.
7. Получить количество информации в изображении путем вычитания полученной стоимости из стоимости изображения.

# РЕШЕНИЕ ТЕОРЕТИЧЕСКИХ ЗАДАЧ



3

## Пример использования помощника «Алиса»

Придумай как посчитать количество путей в графе

Для подсчета количества путей в графе можно использовать следующий алгоритм:

1. Создать список путей, который будет хранить все возможные пути в графе.
2. Создать пустой список путей для хранения текущего пути.
3. Пройти по всем вершинам графа и добавить их в список путей.
4. Если текущая вершина имеет более одного ребра, то добавить ее в список путей и перейти к следующей вершине.
5. Если текущая вершина является конечной вершиной, то добавить путь в список возможных путей.
6. Повторять шаги 3-5 для каждой вершины графа.
7. После того, как все вершины графа будут пройдены, количество возможных путей будет равно длине списка возможных путей.

Ответь иначе

Для подсчета количества путей в графе можно воспользоваться алгоритмом обхода в глубину (depth-first search, DFS). Этот алгоритм позволяет обойти все пути в графе, начиная с начальной вершины и двигаясь по ребрам графа до тех пор, пока не будет достигнута конечная вершина или не будет достигнут конец графа.

Вот пример реализации алгоритма DFS на Python:

```
import networkx as nx
# Создаем граф
G = nx.karate_club_graph()
# Инициализируем количество путей
count = 0
# Выполняем обход графа в глубину
for node in G.nodes():
    if G.degree(node) == 0 and G[node] == True:
        count += 1
print("Количество путей:", count)
```

# РЕШЕНИЕ ЗАДАЧ НА ПРОГРАММИРОВАНИЕ



## Пример использования помощника «Алиса»

придумай задачу на обработку данных во время ввода

Задача на обработку данных при вводе:

Напишите программу, которая считывает данные из стандартного ввода и выводит их в обратном порядке. Например, если пользователь вводит "1 2 3 4 5", программа должна вывести "5 4 3 2 1".

Пример использования:

```
```python
data = input()
rev_data = ""
for i in range(len(data)):
    rev_data += data[i]
print(rev_data)
```
```



Некоторые запросы условий задач уже содержат алгоритмы, **поэтому рекомендуется предварительная подготовка текстов задач перед уроком**

# РЕШЕНИЕ ЗАДАЧ НА ПРОГРАММИРОВАНИЕ



3

«Алиса» может генерировать общий алгоритм и на конкретном языке (по запросу)

Придумай алгоритм решения задачи в списке на подсчёт количества чисел, которые делятся на 3 и не делятся на 7

Алгоритм решения этой задачи может быть следующим:

1. Создайте пустой список для хранения чисел, которые не делятся на 7.
2. Пройдитесь по списку чисел и для каждого числа проверьте, делится ли оно на 3.

Если число делится на 3, добавьте его в список чисел, которые не делятся на 7.

Повторяйте этот процесс до конца списка чисел.

3. После завершения прохода по списку, вы получите список чисел, которые не делятся ни на 3 ни на 7. Количество таких чисел будет равно размеру этого списка.

## Общий алгоритм

Придумай алгоритм решения задачи на python в списке на подсчёт количества чисел, которые делятся на 3 и не делятся на 7

Для решения этой задачи можно использовать следующий код на Python:

```
lst = [1, 2, 3, 4, 5]
count = 0
for i in range(len(lst)):
    if lst[i] % 3 == 0 and lst[i] // 7 != 0:
        count += 1
print(count)
```

Этот код создает список `lst`, содержащий числа от 1 до 5, затем создает переменную `count`, равную 0. Далее он использует цикл `for`, чтобы пройти по каждому элементу списка `lst`. Внутри цикла проверяется, делится ли текущее число на 3 без остатка и делится ли оно без остатка на 7 (оператор `//`). Если оба условия выполняются, то число добавляется в переменную `count`. В конце цикла переменная `count` содержит количество чисел, удовлетворяющих условию задачи.

Ответь иначе

Алгоритм для решения этой задачи на C++:

```
```c++
#include <iostream>
#include <vector>

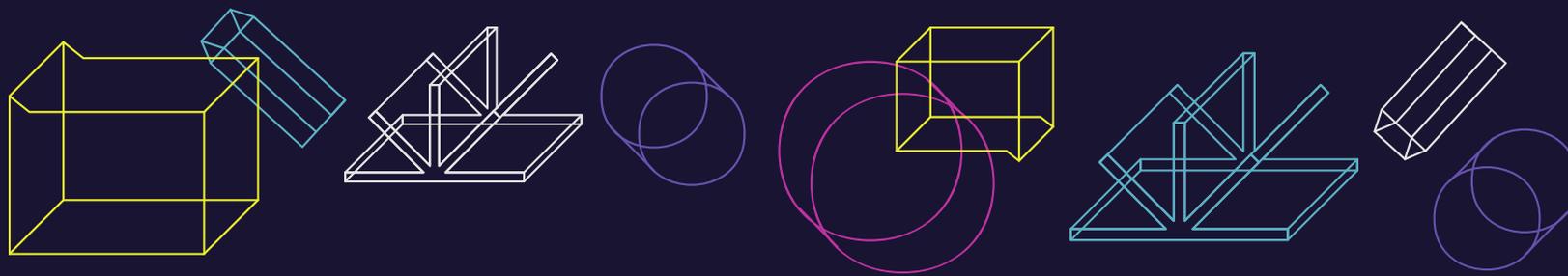
using namespace std;

int main() {
    vector<int> list = {1, 2, 3, 4, 5}; // создаем
    список чисел
    int count = 0; // инициализируем переменную
    для подсчета
    for (int i = 0; i < list.size(); i++) {
        // проверяем делимость на 3
        if (list[i] % 3 == 0) {
            // проверяем делимость на 7
            if (list[i] / 7 == 0 || list[i] / 7 % 7 != 0) { // если не
            делится нацело, то и на 7 не делится
                count++; // увеличиваем количество
                найденных чисел
            }
        }
    }

    cout << "Количество чисел, делящихся на 3 и
    не делящихся на 7: " << count << endl; //
    выводим результат
    return 0;
}
```

Алгоритм на Python (сгенерировался с описанием)

Алгоритм на C++ (сгенерировался не сразу без описания)



**Спасибо за внимание!**