

# Голосовой интерфейс

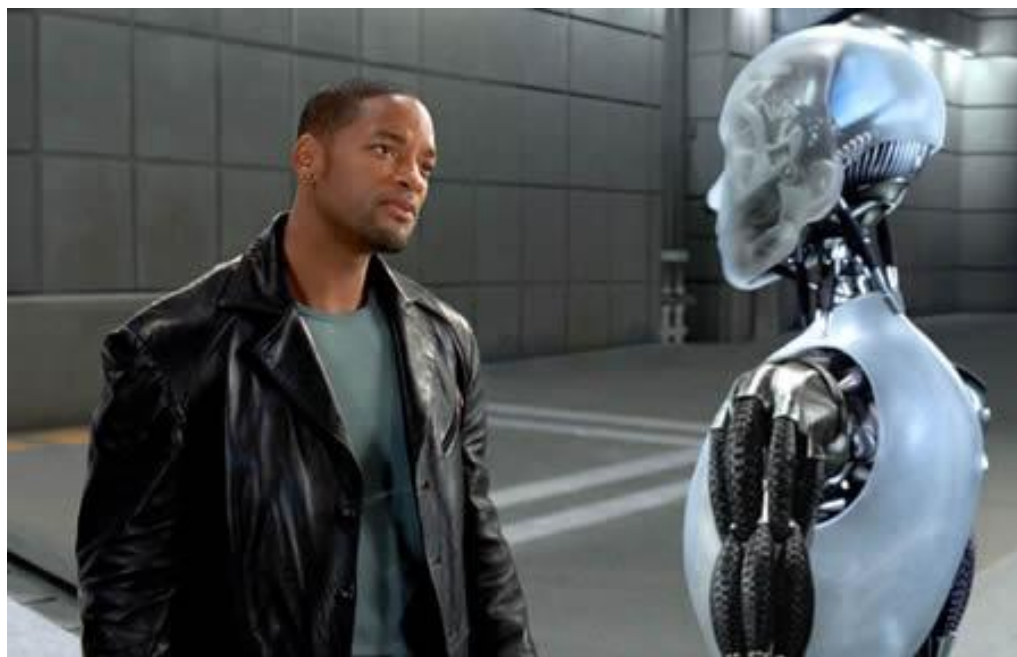
Ст. ИС-21 ИПСА  
Рябченко Алена  
Александровна

# Интерфейс это...

В области информационных технологий (ИТ) средства взаимодействия пользователя с технической системой принято называть интерфейсом.



Одной из важнейших задач разработки современных технических систем является обеспечение наиболее интуитивного и естественного интерфейса с пользователем, то есть современные компьютерные приложения ориентированные на пользователя.



# Типы интерфейсов

По наличию тех или иных средств ввода, интерфейсы разделяются на типы:

1. голосовой,



2. жестовый ,



3. ВОЗМОЖНЫ  
смешанные варианты.



Одной из естественных форм взаимодействия для человека является **речь**. Голосовой интерфейс может улучшить существующий пользовательский интерфейс - он обеспечивает более удобный и менее ограниченный способ взаимодействия человека с компьютером.





Разработка голосового интерфейса является достаточно сложной и комплексной, и требует от разработчика знаний в различных предметных областях, таких как компьютерные науки, лингвистика и психология поведения человека.

# Преимущества голосового интерфейса:



- оперативность и естественность;
- минимум специальной подготовки
- возможность управления объектом в темноте, за пределами его визуальной видимости (в частности, с использованием существующей телефонной сети);
- возможность использования одновременно ручного (с помощью клавиатуры) и голосового ввода информации;
- обеспечение мобильности оператора при управлении.

# Распознавание речи

**Распознавание речи** - технология, позволяющая использовать естественный для человека речевой интерфейс для взаимодействия с электронной техникой.

**Сложность** распознавание речи состоит в том, что совокупность таких характеристик голоса и речи как тембр, громкость, высота, темп, интонация, качество дикции делают речь каждого человека по-своему неповторимой и уникальной как отпечатки пальцев.



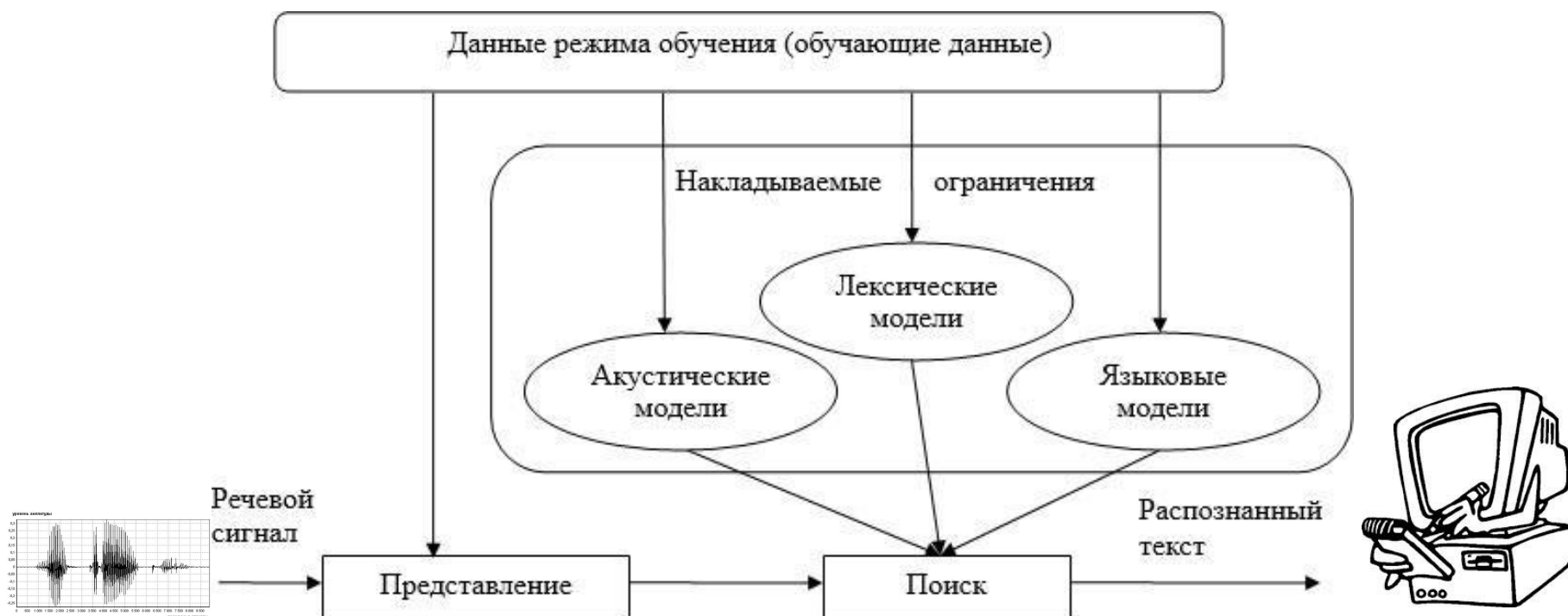
ИЛИ





# Системы распознавания речи

Системы распознавания речи - это системы, анализирующие акустический сигнал алгоритмами, основанными на разнообразных теориях, предполагающих, какие характеристики речевого сигнала создают ощущения звуков данного языка, и математических методах, с той или иной точностью выделяющих значащие параметры акустического сигнала и преобразующие его в различной полноте в необходимую форму.



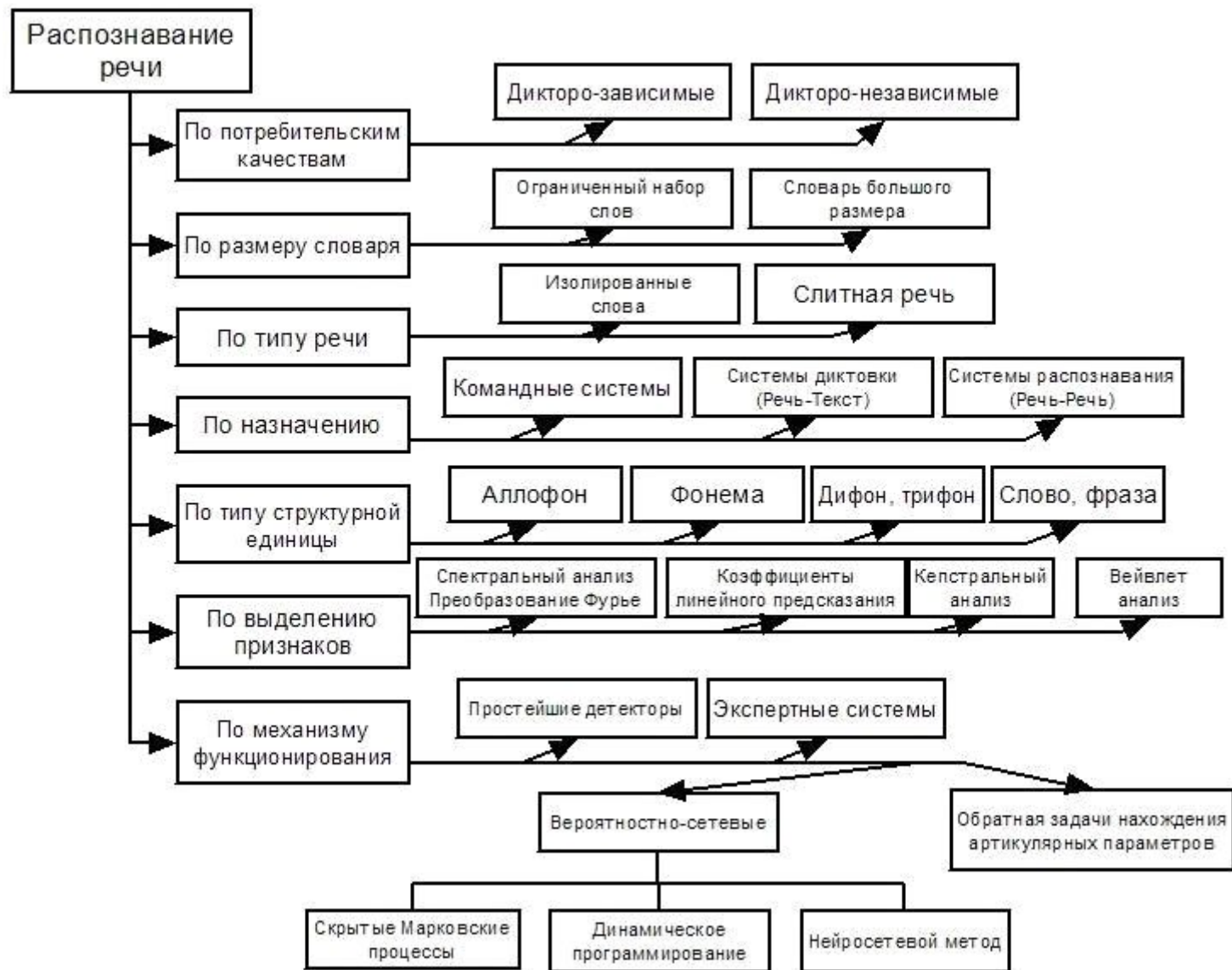
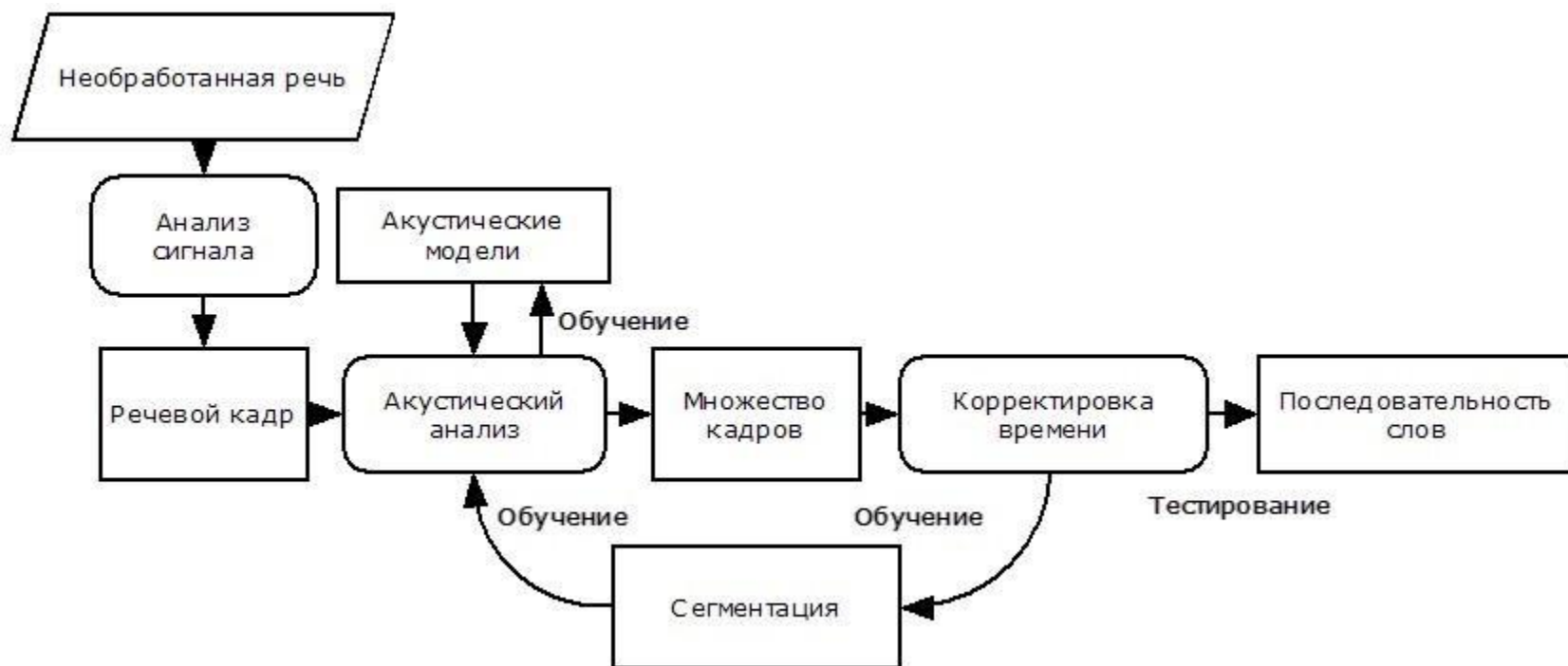


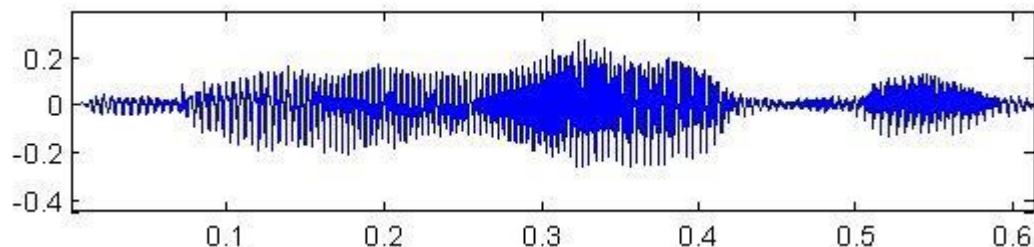
Рис.1 Классификация систем распознавания речи

# Структура стандартной системы распознавания речи



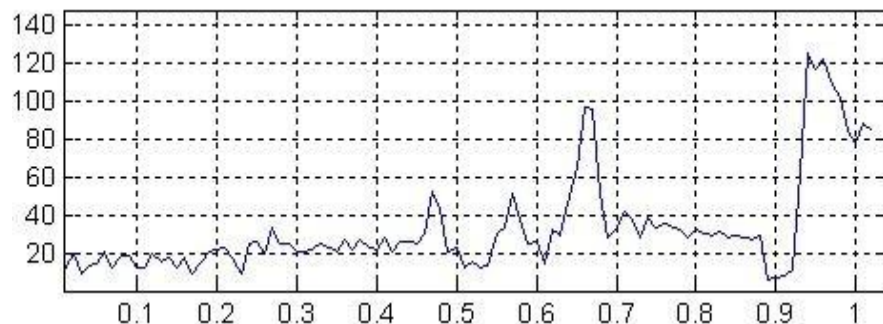
# Необработанная речь

Обычно, поток звуковых данных, записанный с высокой дискретизацией (20 КГц при записи с микрофона либо 8 КГц при записи с телефонной линии).



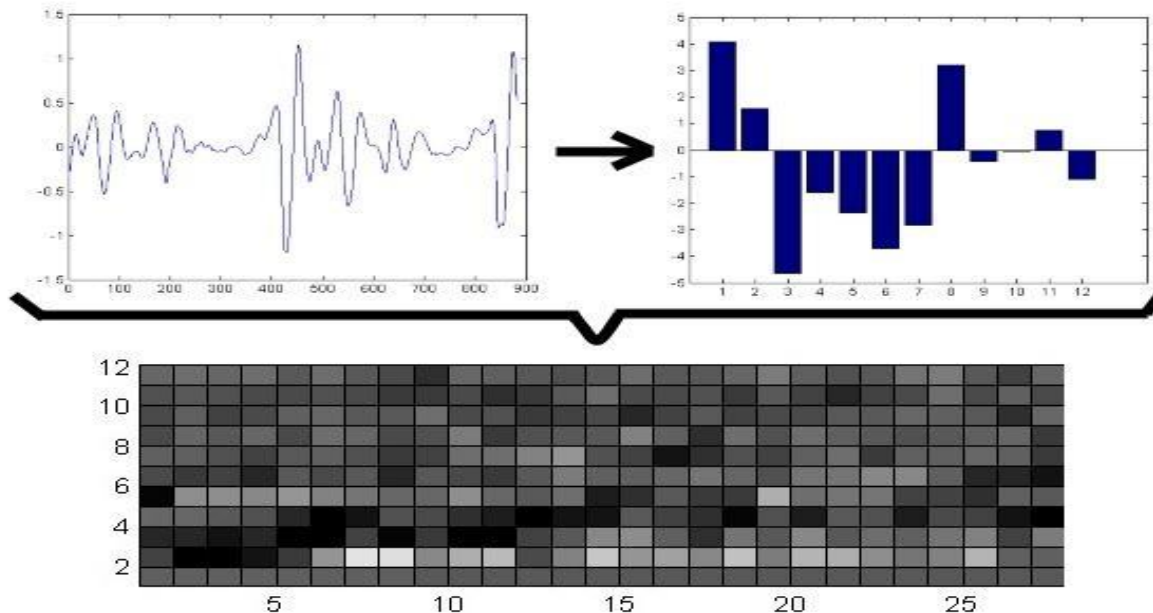
# Анализ сигнала

- Поступающий сигнал должен быть изначально трансформирован и сжат, для облегчения последующей обработки. Есть различные методы для извлечения полезных параметров и сжатия исходных данных в десятки раз без потери полезной информации. Наиболее используемые методы:
  - *анализ Фурье;*
  - *линейное предсказание речи;*
  - *кепстральный анализ.*



# Речевые кадры

Результатом анализа сигнала является последовательность речевых кадров. Обычно, каждый речевой кадр – это результат анализа сигнала на небольшом отрезке времени (порядка 10 мс.), содержащий информацию об этом участке (порядка 20 коэффициентов).



# Акустические модели

- Для анализа состава речевых кадров требуется набор акустических моделей. Рассмотрим две наиболее распространенные из них.
  - *Шаблонная модель.* В качестве акустической модели выступает каким-либо образом сохраненный пример распознаваемой структурной единицы (слова, команды).
  - *Модель состояний.* Каждое слово моделируется как последовательность состояний указывающих набор звуков, которые возможно услышать в данном участке слова, основываясь на вероятностных правилах.

# Акустический анализ

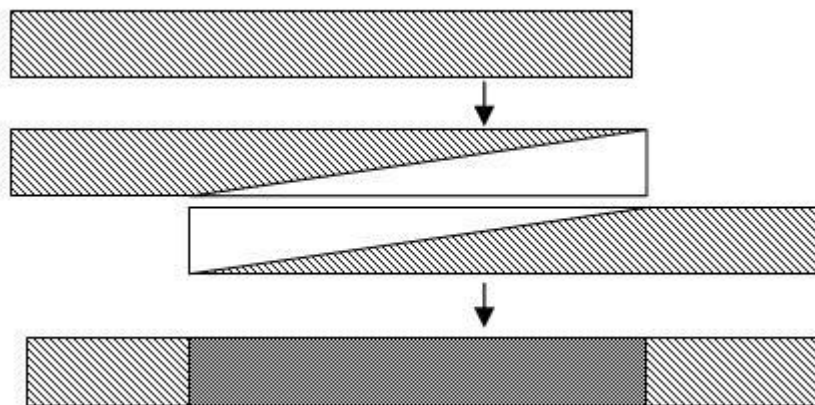


Состоит в сопоставлении различных акустических моделей к каждому кадру речи и выдает матрицу сопоставления последовательности кадров и множества акустических моделей.



# Корректировка времени

Используется для обработки временной вариативности, возникающей при произношении слов (например, “растягивание” или “съедание” звуков).



# Последовательность слов

В результате работы, система распознавания речи выдает последовательность (или несколько возможных последовательностей) слов, которая, наиболее вероятно, соответствует входному потоку речи.



# Программная реализация голосового интерфейса

Используем:

- 1 ) JDK 6 ( J2SE )
- 2 ) Eclipse SDK
- 3 ) Sphinx 4.0
- 4 ) JSAPI ( Included in Sphinx 4.0 )



# Sphinx-4



- Sphinx-4 – самый известное и наиболее работоспособное из открытых программных продуктов для распознавания речи на сегодняшний день.
- Структура Sphinx-4 была разработана с высокой степенью гибкости и модульности. На рисунке показана общая архитектура системы. Каждый помеченный элемент на рисунке представляет собой модуль, который может быть легко заменен.

# Структура Sphinx-4



Главными блоками являются:

1. предварительный анализ,
2. декодер,
3. лингвистический модуль.



1. Акустическая модель преобразует звук в фонемы
2. Словарь преобразует фонемы в слова
3. Языковая модель помогает построить предложения исходя из порядка слов

**Все три составляющие обязательные**, но вместо языковой модели возможно просто описать порядок слов в jsGF словаре примерно так:

camera.jsGF

```
#JSGF V1.0;
```

```
grammar camera;
```

```
public <camera> = <do> ( <cam> <prenum> <num1> | <num2> <cam> );
```

```
<do> = ( включить | выключить | отключить );
```

```
<cam> = ( камеру | видео-камеру );
```

```
<prenum> = [ номер ]
```

```
<num1> = ( один | два | три | четыре | пять | шесть | семь | восемь | девять );
```

```
<num2> = ( первую | вторую | третью | четвертую | пятую | шестую |  
седьмую | восьмую | девятую );
```

# Голосовой калькулятор



## **Мин количество информации:**

1. Поле «А» – 1 символ;
2. Поле «В» – 1 символ;
3. Кнопка «+» или «-».

## **Вводит информацию можно с помощью:**

1. Мышки и клавиатуры;
2. Только клавиатуры, используя клавишу «Tab» для перехода между полями;
3. Голосовые команды.

# Оценка эффективности

- Оценку эффективности и продуктивности интерфейса человека с системой, который реализован в виде компьютерной программы можно сделать с помощью оценки времени выполнения операций для достижения целей пользователя.
- Математическая модель для оценки временных параметров интерфейсов позволяет прогнозировать время выполнения какой-либо задачи пользователем.

В методике GOMS (the model of goals, objects, methods, and selection rules) расчёт эффективности основывается на понятии ментальных операций и их последовательной расстановке.

## Использованы следующие обозначения:

- $H(\text{перемещение})=0,4(\text{с})$ ,
- $M(\text{ментальная подготовка})=1,35(\text{с})$ ,
- $K(\text{нажатие клавиши})=0,28(\text{с})$ ,
- $P(\text{указание})=1,1(\text{с})$ .





# Пример расчет эффективности по методике GOMS

Используем мышку и клавиатуру, введенные данные без ошибок, значения «А» и «В» по 1 символу:

НКНРКНКНРК =>

НМКНМРКНМКНМРМК

$$4 * 0,4 + 5 * 1,35 + 4 * 0,28 + 2 * 1,1 = 11,67(с).$$

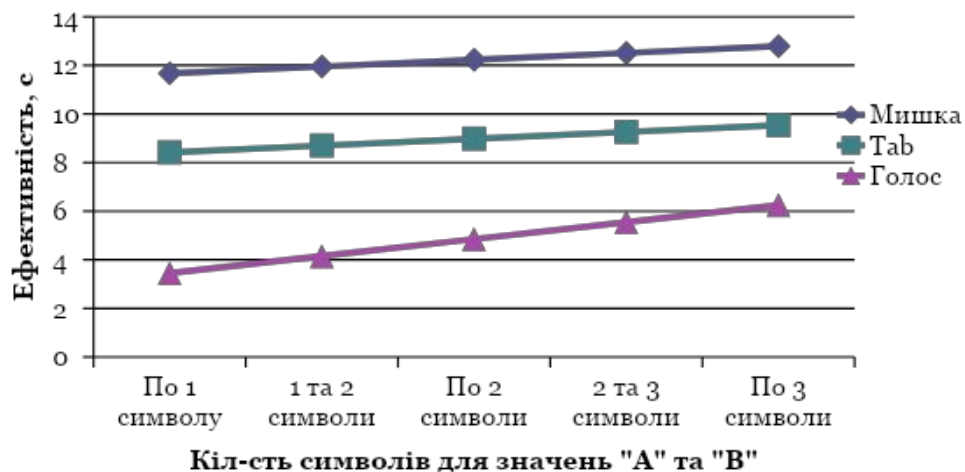
С увеличением количества символов, скорость интерфейса будет снижаться на количество дополнительных  $n$  кликов, то есть на  $n * 0,28с$ .



# Эффективность интерфейсов, если нет ошибок

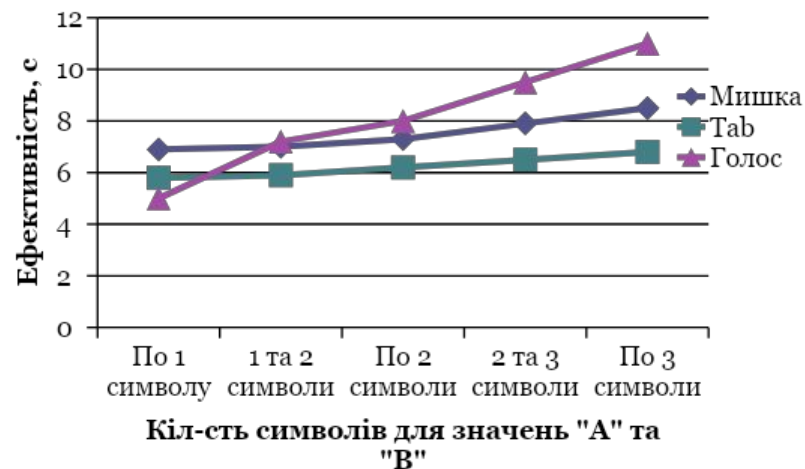
## Методика GOMS

Кіл-ть символів для значень «А» та «В»	Ефективність, с		
	Мишка	«Tab»	Голос
По 1 символу	11,67	8,42	3,45
1 та 2 символи	11,95	8,7	4,15
По 2 символи	12,23	8,98	4,85
2 та 3 символи	12,51	9,26	5,55
По 3 символи	12,79	9,54	6,25

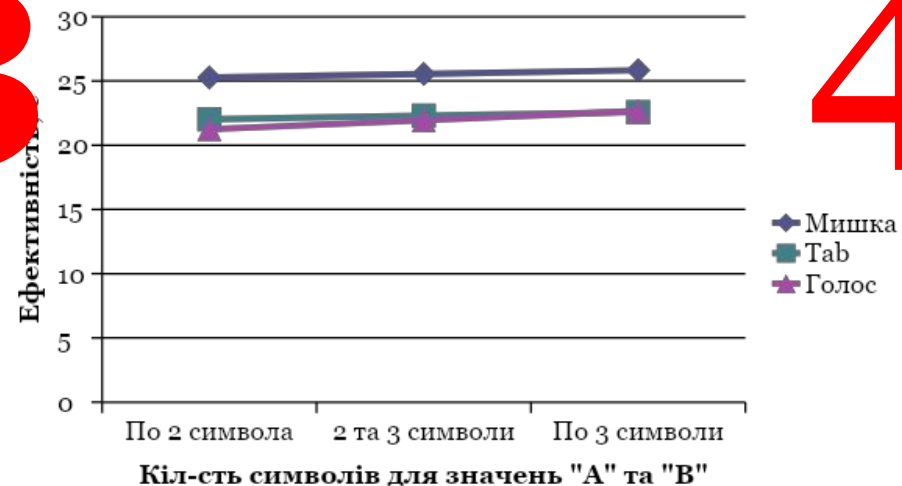
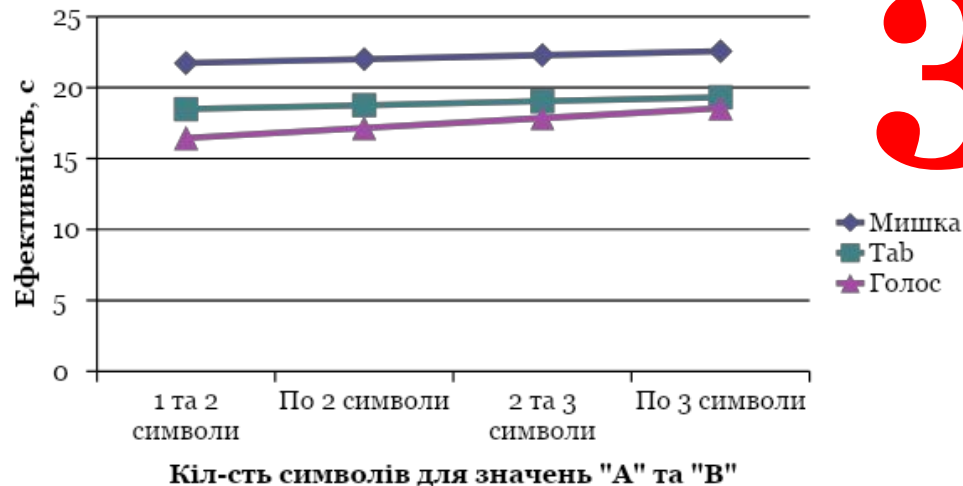
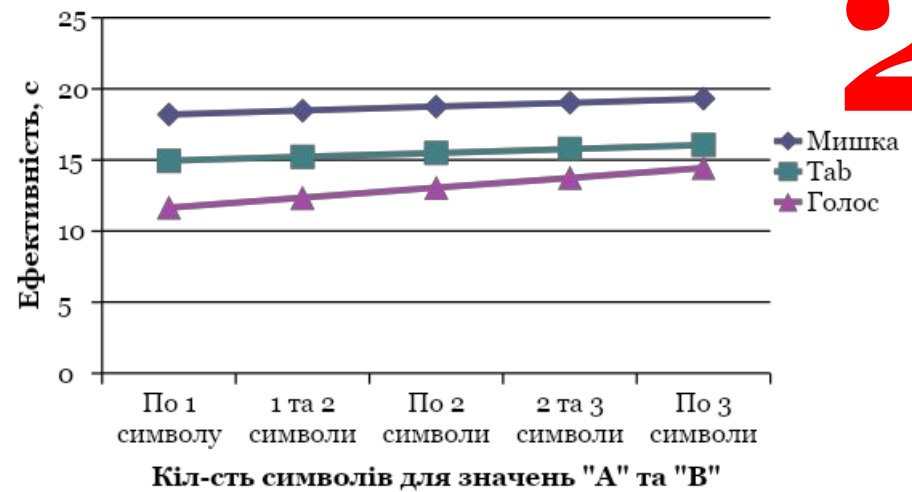
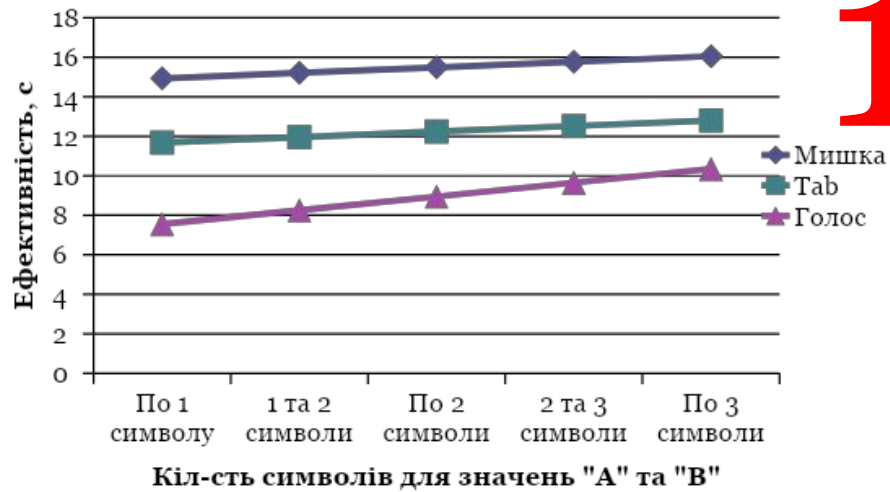


## Экспериментальный метод

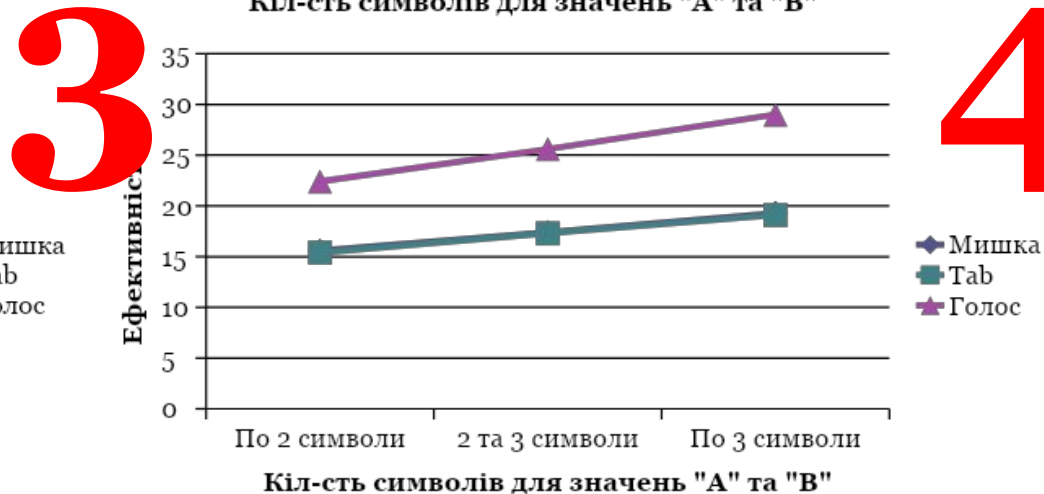
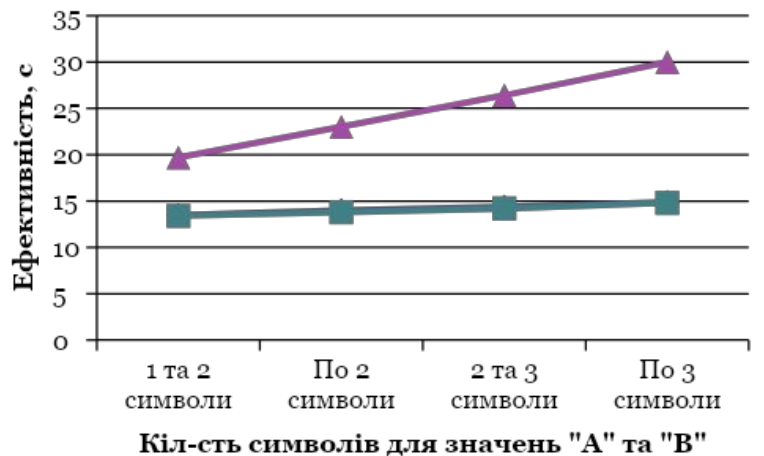
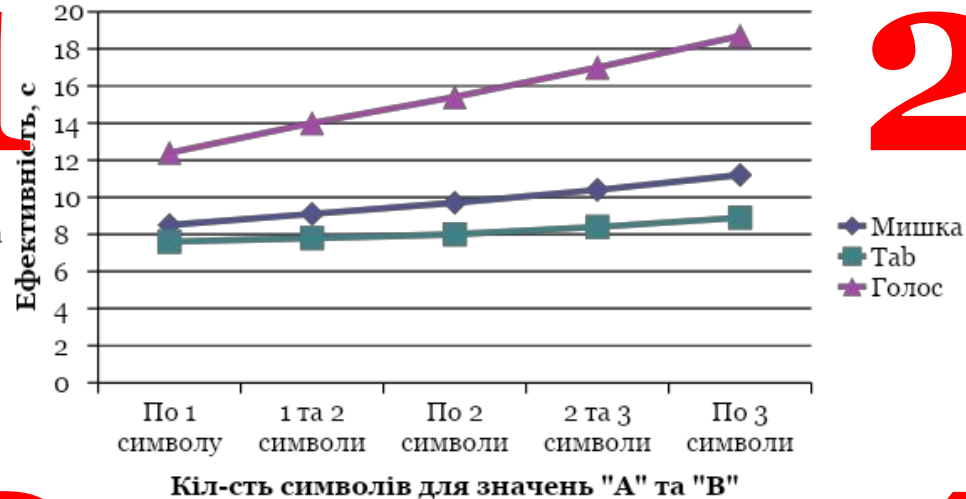
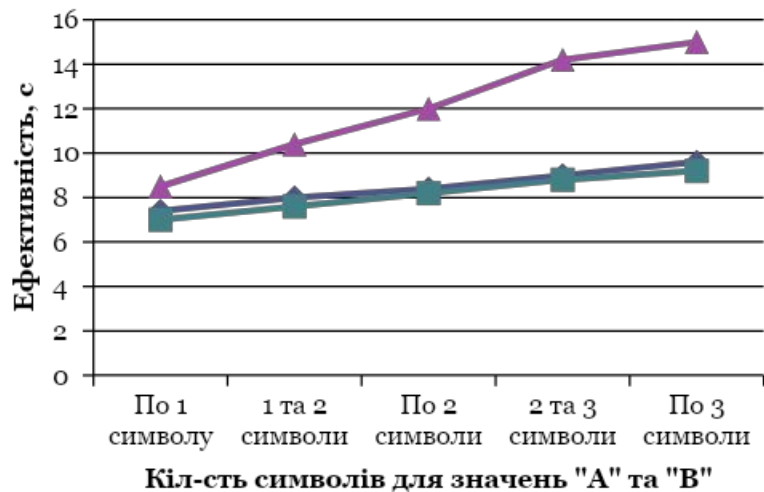
Кіл-ть символів для значень «А» та «В»	Ефективність, с		
	Мишка	«Tab»	Голос
По 1 символу	6,9	5,8	5
1 та 2 символи	7	5,9	7,2
По 2 символи	7,3	6,2	8
2 та 3 символи	7,9	6,5	9,5
По 3 символи	8,5	6,8	11



# Эффективность интерфейсов по методике GOMS, если есть 1-4 ошибки



# Эффективность интерфейсов (экспериментальный метод), если есть 1-4 ошибки



# Вывод

Смешанный тип интерфейса наиболее эффективный для использования в программных приложениях.



Спасибо за внимание!

