



Общие сведения о программометрике

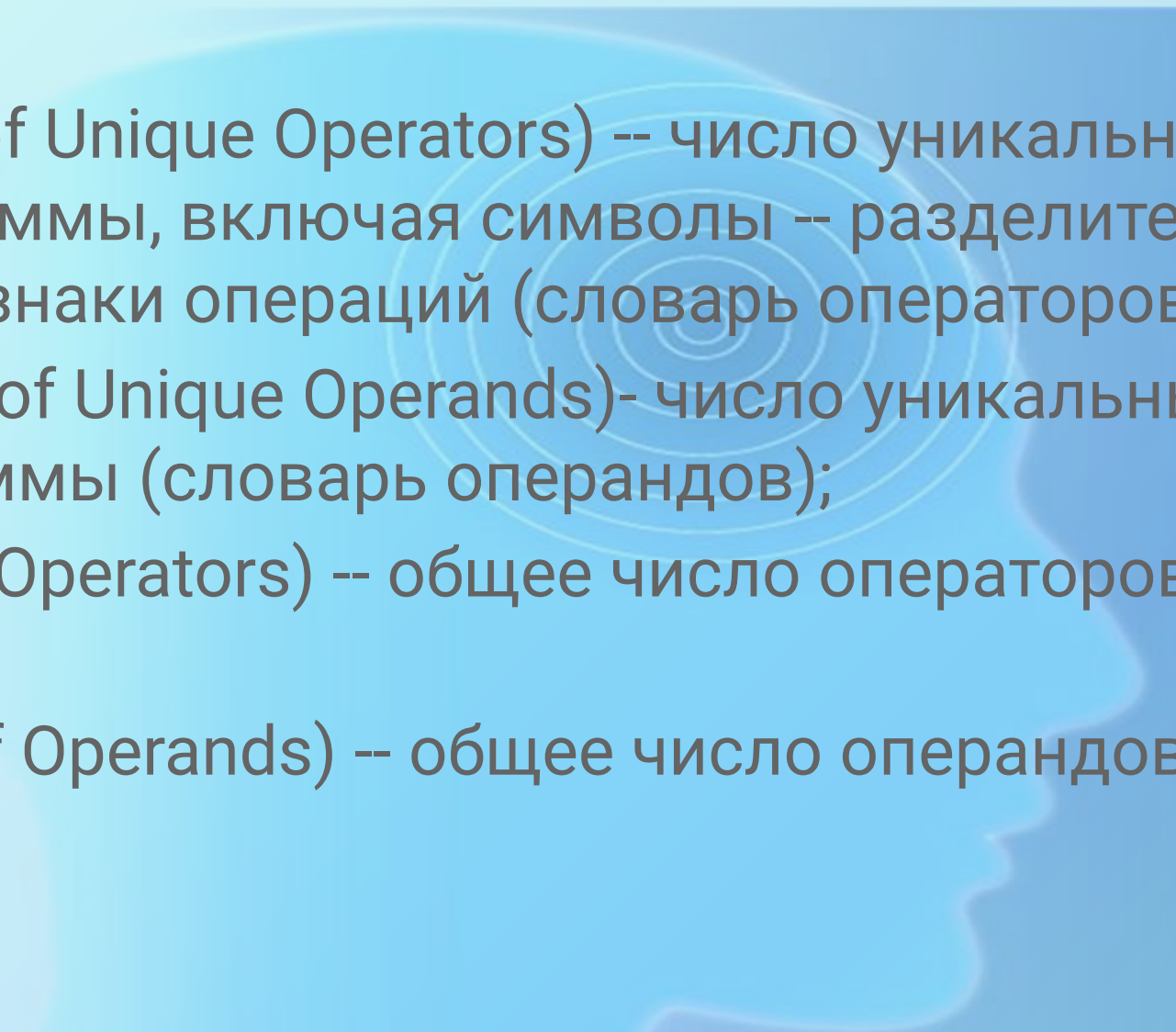
лектор Абдурахманова Н.Н.

Программометрика

- Программометрика позволяет решить следующие задачи:
- - количественный анализ возможности и целесообразности разработки автоматизированных процедур и функций информационных систем (ИС) в заданной постановке
- - численная оценка основных параметров (объем, количество модулей, уровень иерархии, надежность в начальный период эксплуатации) будущих программных средств (ПС) на основе постановки задач;
- - планирование и управление разработкой ПС, оценка трудоёмкости его создания, технико-экономическое обоснование;
- - решение некоторых вопросов, связанных с метрологией качеств ПС

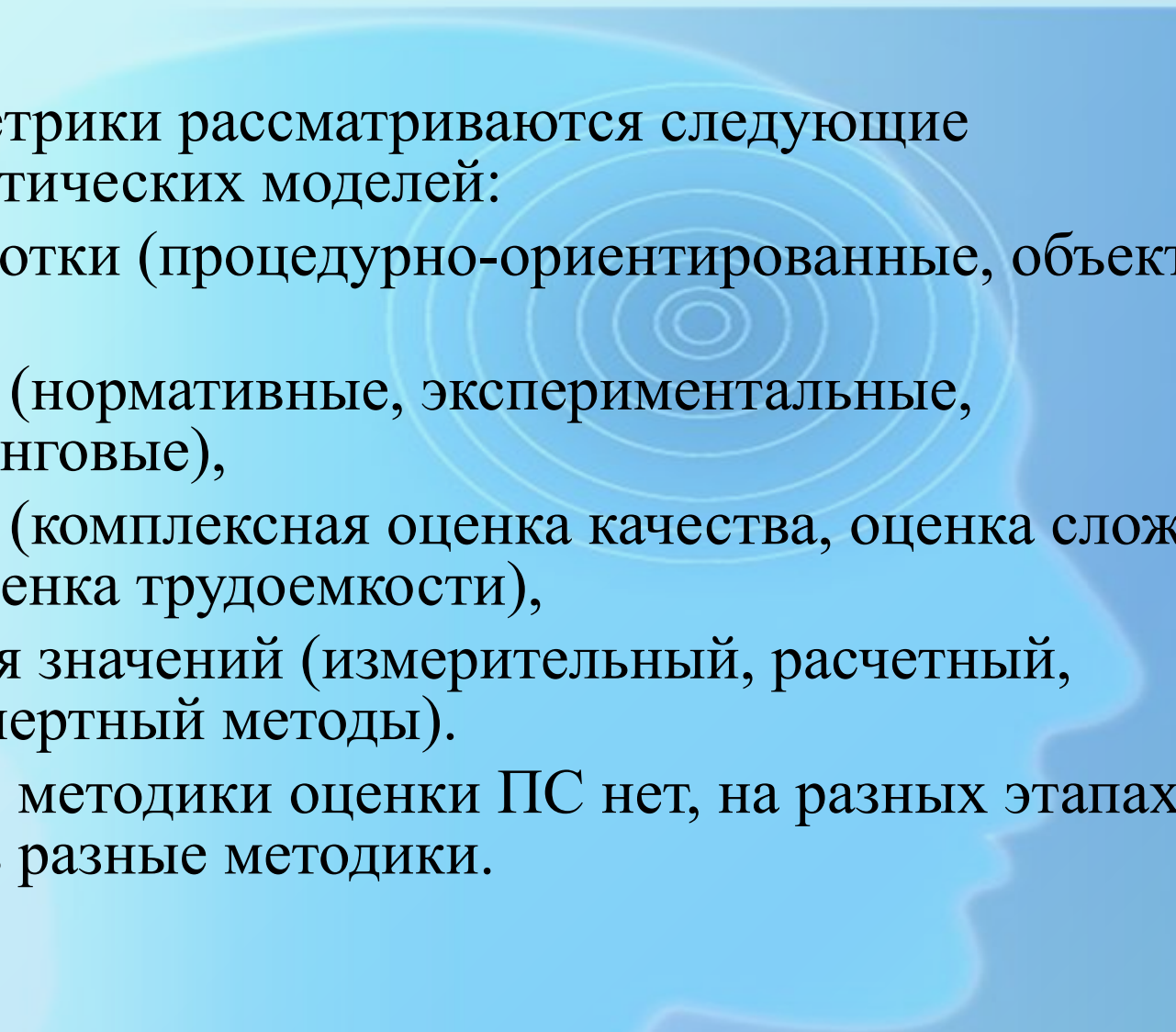
Примеры стандартных метрик

- **Традиционной характеристикой** размера программ является количество строк исходного текста. Оценка размера программ есть оценка по номинальной шкале, на основе которой определяются только категории программ без уточнения оценки для каждой категории. К данной группе оценок можно отнести метрику Холстеда. Основу этой метрики составляют четыре измеряемые характеристики программы:

- 
- · NUOprtr (Number of Unique Operators) -- число уникальных операторов программы, включая символы -- разделители, имена процедур и знаки операций (словарь операторов);
 - · NUOprnd (Number of Unique Operands)- число уникальных операндов программы (словарь операндов);
 - · Noprtr (Number of Operators) -- общее число операторов в программе;
 - · Noprnd (Number of Operands) -- общее число операндов в программе.

- Опираясь на эти характеристики, получаемые непосредственно при анализе исходных текстов программ, М. Холстед вводит следующие оценки:
- словарь программы (Halstead Program Vocabulary) $HPVoc = NUOprtr + NUOprnd$;
- длина программы (Halstead Program Length) $HPLen = Noprtr + Noprnd$;
- объем программы (Halstead Program Volume) $HPVol = HPLen \log_2 HPVoc$.

- Далее Холстед вводит оценку сложность программы (Halstead Difficulty), которая вычисляется как
$$HDiff = NUOprtr / 2 * (NOprnd / NUOprnd)$$
- Используя HDiff Холстед вводит оценку HEff (Halstead Effort):
- $HEff = HDiff * HPVol$, с помощью, которой описывается усилия программиста при разработке.

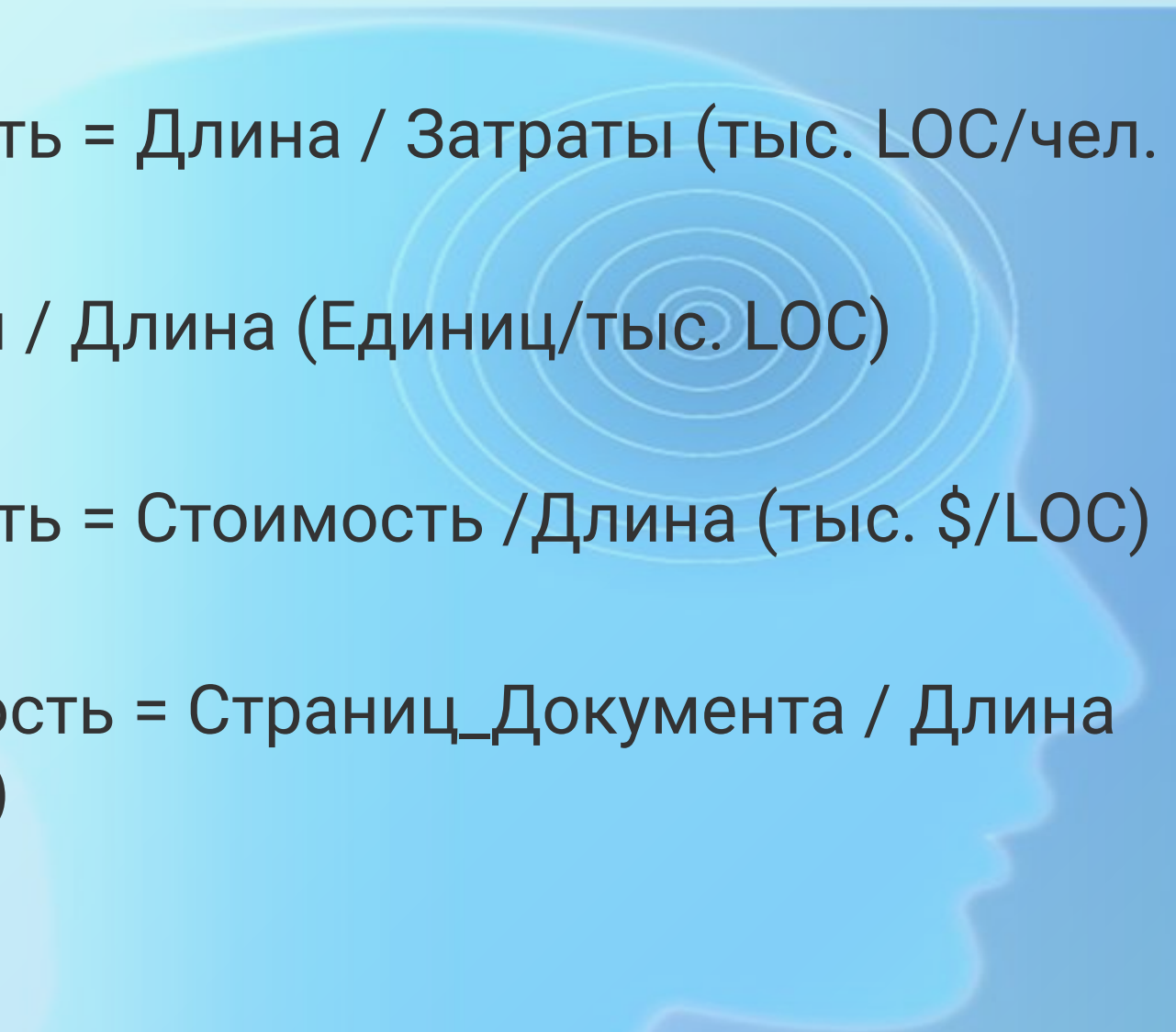
- 
- В рамках программометрики рассматриваются следующие классификации математических моделей:
 - по методологии разработки (процедурно-ориентированные, объектно-ориентированные),
 - по системе измерений (нормативные, экспериментальные, иерархические, одноранговые),
 - по назначению модели (комплексная оценка качества, оценка сложности, оценка надежности, оценка трудоемкости),
 - по методу определения значений (измерительный, расчетный, регистрационный, экспертный методы).
 - Так как универсальной методики оценки ПС нет, на разных этапах ЖЦ необходимо применять разные методики.

Классификация метрик по признаку ориентации

- размеро-ориентированные метрики, использующиеся для сбора результатов прямых измерений программного продукта и его качества, а также процесса разработки;
- функционально-ориентированные метрики, которые являются косвенными мерами, характеризующими функциональное назначение продукта и особенности его входных и выходных данных;
- человеко-ориентированные метрики, которые также являются косвенными мерами, позволяющими судить об отношении персонала (разработчиков и пользователей), об эффективности и качестве работы программного изделия, удобстве взаимодействия с ним, простоте обучения и т. д.

Размерно-ориентированные метрики

- Прямо измеряют программный продукт и процесс его разработки.
- Основываются размерноориентированные метрики на LOC – оценках (Lines Of Code).
- LOC-оценка – это количество строк в программном продукте.

- 
- Производительность = Длина / Затраты (тыс. LOC/чел. -мес.)
 - Качество = Ошибки / Длина (Единиц/тыс. LOC)
 - Удельная стоимость = Стоимость /Длина (тыс. \$/LOC)
 - Документированность = Страниц_Документа / Длина (Страниц/тыс. LOC)

Основные показатели SLOC

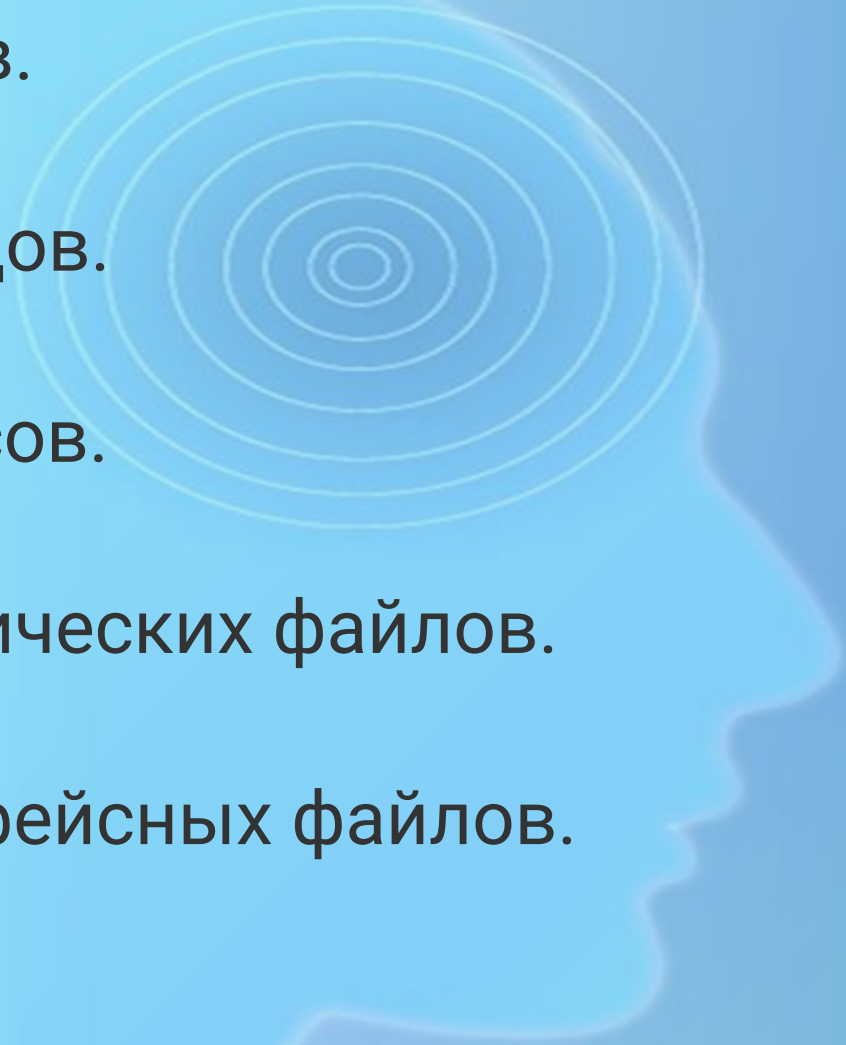
- Количество «физических» строк кода (общее число строк исходного кода, включая комментарии и пустые строки)
- Количество «логических» строк кода (количество команд и зависит от используемого языка программирования)

Функционально-ориентированные метрики

- косвенно измеряют программный продукт и процесс его разработки.
- Вместо подсчета LOC-оценки при этом рассматривается не размер, а функциональность или полезность продукта.

Информационные характеристики метрики

- 1. Количество внешних вводов.
- 2. Количество внешних выводов.
- 3. Количество внешних запросов.
- 4. Количество внутренних логических файлов.
- 5. Количество внешних интерфейсных файлов.



- $FR = \text{Общее количество} * (0,65 + 0,01 * Fi)$
- где Fi - коэффициент регулирования сложности ($i=1 \dots 14$)

Расчет метрик на основе FP

- Производительность = Функц. Указатель / Затраты (FP/чел. - мес.)
- Качество = Ошибки / Функц. Указатель (Единиц/FP)
- Удельная Стоимость = Стоимость / Функц. Указатель (Тыс. \$/FP)
- Документированность=Страниц. Документа/Функц. Указатель (Страниц/FP)

Вычисление значения указателя СВОЙСТВ

- $FR = \text{Общее количество} * (0,65 + 0,01 * Fi)$ где Fi - коэффициент регулировки сложности ($i=1..14$)
- Для сложных систем реального времени это значение на 25-30% больше значения, вычисляемого по таблице для количества функциональных указателей

Features Points

- Для продуктов с высокой алгоритмической сложностью используются метрики свойств (Features Points).
- Они применимы к системному и инженерному ПО, ПО реального времени и встроенному ПО.

Features Points

- Для вычисления указателя свойств добавляется одна характеристика количество алгоритмов.
- Алгоритм определяется как ограниченная программа вычислений, которая включается в общую компьютерную программу.

- Спасибо за внимание

