

Графические схемы и диаграммы в алгоритмизации и программировании

1. Схемы ISO 5807:1985 - это схемы, которые представляют собой стандартизированный метод графического представления алгоритмов и структур данных, используемый в технической документации.

Применение: Эти схемы часто применяются для документирования алгоритмов в технических спецификациях и руководствах пользователя.

Достоинства: Обеспечивают четкое и структурированное представление алгоритмов, легко интерпретируются различными членами команды разработки.

Недостатки: Могут оказаться неудобными для представления сложных и ветвящихся алгоритмов, так как их размер и сложность могут быстро увеличиваться.

Пример представлен на рисунке 1.

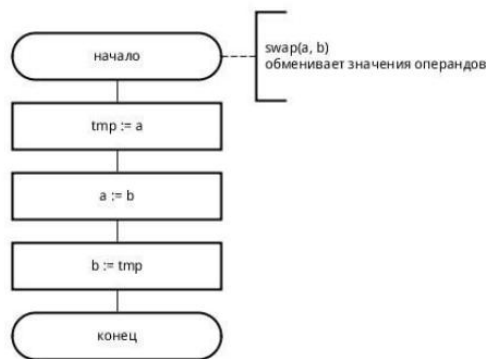


Рис. 1: Схема ISO 5807:1985

Дополнительная информация представлена по ссылкам:

<https://docs.cntd.ru/document/9041994>

<https://otus.ru/nest/post/1778/>

2. ДРАКОН-схемы (ДРАКОН - Дружелюбный русский алгоритмический язык, который обеспечивает наглядность) - это графический язык программирования, который используется для представления алгоритмов с использованием блоков и стрелок.

Применение: ДРАКОН-схемы наиболее удобны для описания алгоритмов в программировании, особенно в системах реального времени, где важно понимать параллельность и синхронизацию операций.

Достоинства: Легко читаемы и понятны, помогают визуализировать сложные алгоритмы, позволяют легко отслеживать поток управления программой.

Недостатки: В больших проектах с большим количеством условных ветвлений и циклов схемы могут стать громоздкими и трудными для анализа.

Пример представлен на рисунке 2.



Рис. 2: ДРАКОН-схема

Дополнительная информация представлена по ссылке:

<https://habr.com/ru/articles/541478/>

3. Диаграммы деятельности UML - схемы, которые представляют собой графическое изображение последовательности действий или операций в системе.

Применение: Широко используются для моделирования бизнес-процессов, алгоритмов и операций в программном обеспечении.

Достоинства: Позволяют наглядно представить последовательность действий и условий в системе, обеспечивают легкость в понимании логики работы программы.

Недостатки: В некоторых случаях могут стать слишком сложными для отображения больших и сложных процессов.

Пример представлен на рисунке 3.

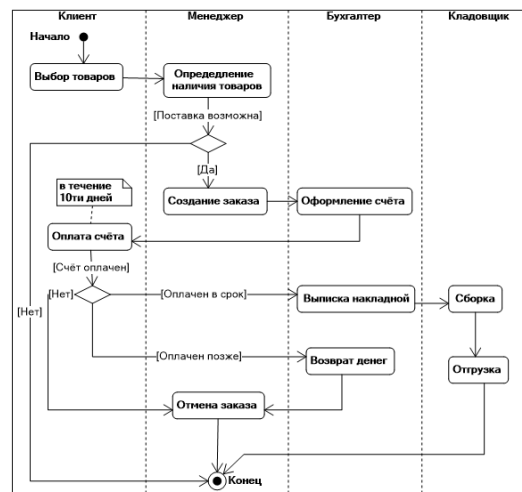


Рис. 3: Диаграмма деятельности UML

Дополнительная информация представлена по ссылкам:

<https://www.cybermedian.com/ru/a-comprehensive-guide-to-14-types-of-uml-diagram/>

[https://itonboard.ru/analysis/664-](https://itonboard.ru/analysis/664-diagramma-dejatelnosti-rukovodstvo-dlja-nachinajushhih/?ysclid=lsuz0pkgw1191638757)

[diagramma-dejatelnosti-rukovodstvo-dlja-nachinajushhih/?ysclid=lsuz0pkgw1191638757](https://itonboard.ru/analysis/664-diagramma-dejatelnosti-rukovodstvo-dlja-nachinajushhih/?ysclid=lsuz0pkgw1191638757)

4. Диаграммы потоков данных (DFD) - схемы представляют собой графическое изображение потоков данных в системе, включая процессы, хранилища данных и потоки данных между ними.

Применение: Используются для моделирования структуры и потоков данных в информационных системах.

Достоинства: Помогают в понимании потоков данных в системе, легко воспринимаемы и понятны, позволяют выявить потенциальные узкие места в системе.

Недостатки: Могут оказаться неудобными для подробного описания действий внутри процессов, не позволяют полностью охватить всю сложность системы.

Пример представлен на рисунке 4.

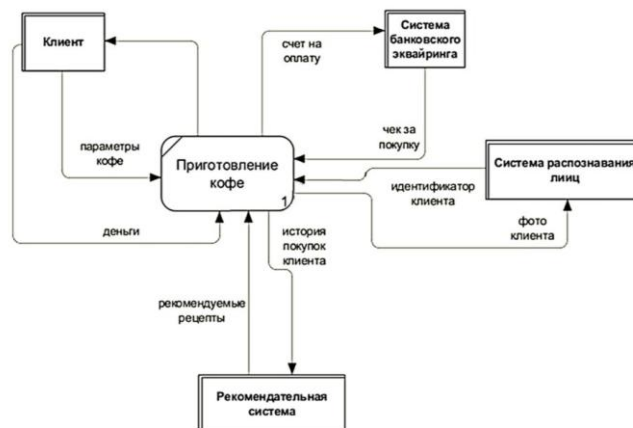


Рис. 4: Диаграмма потоков данных

Дополнительная информация представлена по ссылкам:

<https://www.lucidchart.com/pages/ru/%D0%B4%D0%B8%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0-dfd>

<https://habr.com/ru/articles/668684/>

5. Р-схемы (Program Evaluation and Review Technique) - метод графического представления алгоритмов и программ, который использует специальные символы и стрелки для показа последовательности действий, ветвлений и других конструкций.

Применение: Р-схемы широко используются для визуализации алгоритмов и программ, особенно при планировании и управлении проектами. Они помогают лучше понять структуру программного кода, выявить потенциальные ошибки и оптимизировать процессы.

Достоинства: Предоставляют наглядное представление алгоритма или программы, что делает их легко читаемыми и понятными, позволяют четко описать последовательность действий и условия выполнения задач, а также визуализация алгоритмов позволяет обнаружить узкие места и оптимизировать процессы.

Недостатки: Для очень сложных алгоритмов или программ Р-схемы могут стать громоздкими и трудными для понимания, а также иногда одна и та же схема может быть интерпретирована по-разному различными людьми, что может привести к недопониманию.

Пример представлен на рисунке 5.



Рис. 5: Р-схема алгоритма

Дополнительная информация представлена по ссылкам:

https://docs.nevacert.ru/files/gost/gost_19.005-1985.pdf

<https://elib.bsu.by/bitstream/123456789/10196/1/Konakh64.pdf>

6. Диаграммы Нэсси-Шнейдермана - это метод графического представления алгоритмов, в котором используются блоки различных форм для представления последовательных действий, ветвлений и циклов.

Применение: Они удобны для представления алгоритмов и программных процессов. Часто используются в учебных материалах по информатике и при обучении программированию начинающих.

Достоинства: Легко читаются и понимаются даже без специальных знаний, позволяют четко видеть последовательность действий и условных ветвлений, а также подходят для использования в качестве вводного материала при обучении основам программирования.

Недостатки: Могут быть неудобными для представления сложных алгоритмов или не подходящими для определенных типов алгоритмов, например, рекурсивных функций.

Пример представлен на рисунке 6.

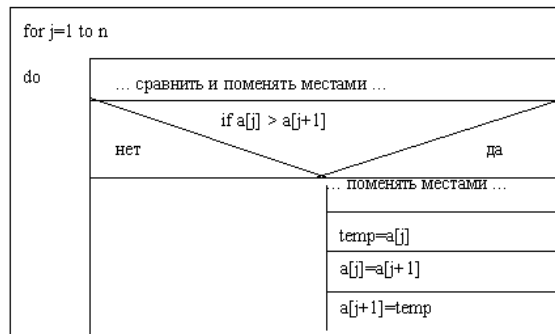


Рис. 6: Диаграмма Нэсси-Шнейдермана

Дополнительная информация представлена по ссылкам:

https://bstudy.net/724984/informatika/diagrammy_nassi_shneydermana?ysclid=lsv1vp5c5t336055479

https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_%D0%9D%D0%B0%D1%81%D1%81%D0%B8_%E2%80%94%D0%A8%D0%BD%D0%B5%D0%B9%D0%B4%D0%B5%D1%80%D0%BC%D0%B0%D0%BD%D0%B0#:~:text=%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0%