

ВИЗУАЛИЗАЦИЯ АЛГОРИТМИЧЕСКИХ ПРОЦЕССОВ КАК ИНСТРУМЕНТ ФОРМИРОВАНИЯ АБСТРАКТНОГО МЫШЛЕНИЯ У ПРОГРАММИСТОВ

Негруша О. Е., студент

Белорусский национальный технический университет

Минск, Республика Беларусь

Научный руководитель: старший преподаватель Воронич Л. В.

Аннотация. Статья рассматривает визуализацию алгоритмов как дидактический инструмент инженерной педагогики для развития абстрактного мышления у начинающих программистов. На основе теории мультимедийного обучения и когнитивной нагрузки обоснована методика «ступенчатой визуализации» с постепенным снятием опор. Описаны элементы методики, метрики оценки и условия внедрения в курс начального программирования.

Для современного инженерного образования важны не только прикладные навыки кодирования, но и сформированное абстрактное мышление – способность оперировать обобщениями, моделями и инвариантами. Студенты начальных курсов нередко застревают на синтаксисе, не выстраивая ментальных моделей исполнения программ. Визуализация алгоритмов (далее – ВА) способна соединять конкретный код и абстрактные структуры данных, снижая порог входа и ускоряя переход к самостоятельному теоретическому рассуждению.

Вопрос состоит в том, что традиционные формы обучения программированию (лекция и сборник задач) могут перегружать когнитивные функции студентов начальных курсов деталями синтаксиса и инструментов, оставляя в тени причинно-следственные связи на уровне состояний программы. Возможен метод, который систематически делает внутренние состояния алгоритма наблюдаемыми, поддерживает переход от конкретных примеров к абстракциям, контролирует когнитивную нагрузку и снижает потребность в визуальных подсказках.

Три взаимодополняющих положения определяют выбор средств ВА:

1. Теория мультимедийного обучения: интеграция словесных и графических представлений при умеренной сложности повышает понимание за счет двойного кодирования и сегментации материала.

2. Теория когнитивной нагрузки: визуальные средства полезны, если уменьшают внешний и поддерживают значимый типы нагрузки, не увеличивая чрезмерно внутреннюю сложность [2].

3. Рекомендации СС2020 по вычислительным учебным планам: акцент на визуально-интерактивные средства при обучении абстракциям данных и алгоритмов, особенно на начальном этапе, для выстраивания ментальных моделей исполнения [1].

Далее описывается возможная методика ступенчатой визуализации (трехступенчатый цикл с угасанием опор) для базового курса программирования:

Шаг 1. Конкретизация (управляемая визуализация).

Инструменты: блок-схемы, трассировка по шагам (в данном процессе программа выполняется по шагам и при этом фиксируются состояния переменных и структур данных, анимация структур данных (стеки, очереди, деревья), подписи, которые связывают код, состояние памяти и инварианты цикла.

Дидактика: небольшие сегменты по 5–7 минут с вопросами на предсказание следующего состояния, принцип разделенного внимания (код и визуализация синхронизированы пространственно-временно).

Шаг 2. Конструирование.

Инструменты: анимации без подписей, задания на восстановление пропущенных шагов, карточки-инварианты.

Дидактика: студенты сначала делают догадку, затем соотносят с эталонной визуализацией, растет доля самостоятельного объяснения и понятия.

Шаг 3. Абстрагирование.

Инструменты: псевдокод без анимации, доказательство корректности через инварианты, оценка сложности по преобладающим операциям.

Дидактика: визуальные опоры снимаются, требуются словесно-символические аргументы и перенос в новые контексты (например, от сортировки вставками к сортировке выбором).

Возможные метрики и оценивание:

– тесты на трассировку сухим способом (т. е. пошаговое мысленное выполнение алгоритма без запуска программы) и на перенос (задания, проверяющие, может ли студент применить усвоенную схему рассуждения/инвариант в похожей иной задаче) как проверка понимания студентом причинно-следственных связей в коде и корректности построения ментальной модели исполнения алгоритма, способствует росту точности и скорости решения;

– краткие опросники ментальной нагрузки (9-балльная шкала NASA-TLX в урезанном виде на усмотрение преподавателя) для контроля избыточной наглядности [2];

– критерии качества объяснения: наличие инварианта, корректная причинность между состояниями, отсутствие апелляции к частным примерам.

Педагогические условия:

– сегментация: короткие визуальные эпизоды, интегрированные в решение задач, минимизирующие теорию без конкретного практического применения;

– принцип соответствия: визуализируется именно то, что недоступно непосредственному восприятию (скрытые состояния и инварианты);

– планомерное угасание: каждые 2–3 недели снижать насыщенность визуальных подсказок, переводя акцент с образа на символику и речь [3].

Методика повышает осмысленность рассуждений, сокращает синтаксический шум и ускоряет переход к формализации. Но также существуют возможные ограничения в виде чрезмерной анимации, которая может создавать лишнюю внешнюю нагрузку. Нужна адаптация под уровень студентов (группы, потока) и содержание темы (не все абстракции одинаково визуализируемы) [3].

Список использованных источников

1. Новый этап международной стандартизации ИТ-образования // КиберЛенинка. – URL: <https://cyberleninka.ru/article/n/novyy-etap-mezhdunarodnoy-standartizatsii-it-obrazovaniya> (дата обращения: 24.11.2025).

2. Рыбина, Е. П., Опросник NASA-TLX для измерения когнитивной нагрузки: адаптация на русскоязычной выборке/Е. П. Рыбина, А. Н. Ангельгарт, Т. А. Березнер, А. И. Расторгуева, [и др.] // Психология познания: сборник материалов Всероссийской научной конференции памяти Дж. С. Брунера, 2023. – URL: <https://publications.hse.ru/chapters/823749333> (дата обращения: 27.11.2025).

3. Computing Curricula 2020: Paradigms for Global Computing Education. – CC2020 Task Force Association for Computing Machinery New York – 2020. – URL: <https://dl.acm.org/doi/10.1145/3467967> (дата обращения: 27.11.2025).