



● РАЗВИТИЕ ИНТЕЛЛЕКТА ШКОЛЬНИКОВ

С. М. Окулов

АБСТРАКТНЫЕ ТИПЫ ДАННЫХ



ИЗДАТЕЛЬСТВО

БИНОМ

С. М. Окулов

АБСТРАКТНЫЕ ТИПЫ ДАННЫХ



Москва
БИНОМ. Лаборатория знаний
2009

УДК 519.85(023)

ББК 22.18

О52

Окулов С. М.

О52 **Абстрактные типы данных / С. М. Окулов. — М. : БИНОМ. Лаборатория знаний, 2009. — 250 с. : ил. — (Развитие интеллекта школьников).**

ISBN 978-5-94774-869-7

Абстракция, абстрагирование — одна из составляющих мыслительного процесса творческой личности. Для развития этого компонента мышления в процессе обучения информатике есть дополнительные возможности, так как знание абстрактных типов данных, умение оперировать ими — это необходимый элемент профессиональной культуры специалиста, связанного с разработкой программных комплексов.

Книга предназначена для школьников, преподавателей информатики и студентов младших курсов университетов. Она может быть использована как в обычных школах при проведении факультативных занятий, так и в образовательных учреждениях с углубленным изучением информатики.

УДК 519.85(023)

ББК 22.18

По вопросам приобретения обращаться:

(499) 157-52-72, e-mail: binom@Lbz.ru

<http://www.Lbz.ru>

ISBN 978-5-94774-869-7

© Окулов С. М., 2009

© БИНОМ. Лаборатория знаний, 2009

Оглавление

Предисловие	5
Введение	12
Глава 1. Матрицы	15
1.1. Основные понятия	15
1.2. Операции над матрицами	18
1.3. Элементарные преобразования матриц	27
Глава 2. Списки	36
2.1. Основные понятия о ссылочном типе данных (указателях)	36
2.2. Линейный список	39
2.3. Реализация линейного списка с использованием массивов	47
2.4. Двусвязные списки	51
Глава 3. Стек	58
3.1. Основные понятия	58
3.2. Реализация стека через линейный список	59
3.3. Реализация стека с использованием массива	60
3.4. Постфиксная, префиксная и инфиксная формы записи выражений	66
3.5. Стек и рекурсивные процедуры	73
Глава 4. Очередь	84
4.1. Определение и реализация очереди с использованием списков	84
4.2. Реализация очереди с помощью массива	86
Глава 5. Деревья	94
5.1. Основные понятия	94
5.2. Двоичные деревья поиска	96
5.3. Способы описания деревьев	105
5.4. Оптимальные двоичные деревья поиска	115

Глава 6. Множества	125
6.1. Основные понятия	125
6.2. Стандартные способы реализации множества	127
6.3. Объединение непересекающихся множеств	129
6.4. Использование древовидных структур данных в задаче объединения непересекающихся множеств	134
6.5. Словари и хеширование	141
Глава 7. Очереди с приоритетами	150
7.1. Двоичная куча и пирамидальная сортировка	150
7.2. Очередь с приоритетом на базе двоичной кучи	161
7.3. Биномиальная куча	165
Глава 8. Сбалансированные деревья	178
8.1. AVL-деревья	178
8.2. «2–3»-деревья	189
8.3. B-деревья	207
8.4. Красно-черные деревья	225

Предисловие

Когда человек познает самого себя,
сфинкс засмеется.

Надпись на постаменте сфинкса

Выдвинем *утверждение* о том, что информатика обладает исключительным, присущим только этому предмету ресурсом по интеллектуальному развитию школьника.

Схема доказательства этого утверждения: 1) дадим неформальное определение информатики как школьного предмета; 2) выделим особенности учебной деятельности, присущие только предмету «Информатика»; 3) установим соответствие между деятельностью в информатике и интеллектуальным развитием — например, на базе основных положений теории интеллектуального развития Ж. Пиаже¹⁾.

1.

Во-первых, в школьном курсе информатики должны изучаться фундаментальные основы этого предмета²⁾.

Во-вторых, между понятиями «информатика» и «computer science» мы практически ставим знак эквивалентности. Из этого вытекает, например, то, что в информатике не

¹⁾ Жан Пиаже (1896–1980) — психолог, чье влияние на развитие психологии XX века было определяющим. В генетической эпистемологии (как называют его теорию) отражено глубокое убеждение в том, что развитие интеллекта проходит определенные стадии и является результатом динамического взаимодействия ребенка и окружающей среды.

²⁾ О фундаментальных основах информатики вы можете прочитать короткое эссе в предыдущей книге серии «Развитие интеллекта школьников»: Окулов С. М., Лялин А. В. «Ханойские башни». М.: БИНОМ. Лаборатория знаний, 2008.

следует изучать информационные процессы в целом (везде и всюду), а только те из них, которые присущи компьютерной науке.

Очевидно, каждому виду человеческой деятельности, каждой науке свойственны свои способы и методы выражения мысли, свои методы формализации и создания моделей, свой язык описания процессов. С внедрением же компьютера в какую-то предметную область задачи и проблемы этой предметной области должны быть представлены в системе понятий компьютерной науки. Однако обратное утверждение неверно.

В-третьих, синтезирующим видом деятельности в нашем понимании информатики является программирование¹⁾.

Таким образом, утверждение о том, что информатика может изучаться на основе программирования, через программирование, имеет право на существование. Не путем освоения информационных технологий как таковых (это уже задача, например, отдельного предмета «Прикладная информатика»), не путем разбора понятий «информационный процесс», «моделирование», «формализация» в целом, — а в ходе самостоятельной разработки некоего целого с использованием, на основе фундаментальных понятий предметной области. Ключевое положение здесь — *самостоятельная деятельность*²⁾, которой присущи некие черты (назовем их, например, А, В, С).

2.

Программирование — это многогранная деятельность. Кратко охарактеризуем только один ее аспект. Программу можно рассматривать как *формализованную запись метода решения* некой сложной проблемы. Программа попадает и под понятие *модели*, ибо последняя — не что иное, как имитация структуры или функционирования объекта (например, коммивояжера, которому требуется определенным образом обойти заданную совокупность городов). Но главное даже не в этом, а в том, что программа есть *динамичес-*

1) Обоснование этого положения дано в монографии: *Окулов С. М.* Информатика: развитие интеллекта школьника. М.: БИНОМ. Лаборатория знаний, 2005.

2) Самостоятельная деятельность не исключает учителя. Она понимается по М. Мамардашвили: человек может что-то понять, узнать только сам, никому не дано пребывать в состоянии чужой мысли и заставить другого мыслить.

кая модель со всеми вытекающими из этого положения следствиями, принципиальным из которых является появление возможности исследовать эту модель. И именно в данной возможности состоит принципиальное отличие информатики от любого другого школьного предмета!

Что происходит при решении сложной проблемы (задачи), при разработке и исследовании динамической модели? Сложная задача разбивается на взаимосвязанные подзадачи. Последние, в свою очередь, опять разделяются на свои подзадачи и т. д. вплоть до самых низших уровней нашего понимания задачи.

Что мы при этом получаем? Во-первых, сложная задача (проблема) описывается некой иерархической структурой; во-вторых, определяются принципы ее декомпозиции; в-третьих, так как каждый уровень — это определенный *уровень абстрагирования*, создается инструментарий для описания абстракций. *Иерархия* — это ранжированная или упорядоченная система абстракций, расположение частей или элементов целого в порядке от высшего к низшему. В результате деятельности программиста создается иерархическая структура из абстракций (А). Но ведь «абстракция — это такие существенные характеристики исследуемого объекта, которые отличают его от всех других видов объектов и, таким образом, четко определяют особенности данного объекта с точки зрения дальнейшего рассмотрения и анализа»¹⁾. Г. Буч (и не только он²⁾) разделяет *абстракцию сущности объекта* — когда объект представляет собой модель существенных сторон предметной области, и *абстракцию поведения* — когда объект состоит из обобщенного множества операций, каждая из которых выполняет определенную функцию. Другими словами, сущность задачи (проблемы) и метода ее решения мы описываем (очевидно, на каком-то языке, а это уже и есть формализованная запись!) на уровне данных в программе, а ее поведение — действиями над данными.

Но как осуществляется движение от проблемы (задачи) к модели, к динамической модели? С помощью логических приемов мышления, основными из которых являются ана-

¹⁾ Буч Г. Объектно-ориентированное проектирование с примерами применения. М.: Конкорд, 1992.

²⁾ Здесь можно сослаться на работы В. М. Глушкова, известного специалиста XX в. по кибернетике, и на работы родоначальника школьной информатики А. П. Ершова.

лиз, всегда сочетающийся с абстрагированием¹⁾, и синтез! Но описывают ли они полностью этот процесс движения к результату? Вряд ли! Говорят, что этими дефинициями отражается рациональная сторона мышления — поэтому далее вводится понятие метода. *Метод* (в самом широком смысле слова) — это «путь к чему-либо», способ достижения определенной цели, совокупность приемов или операций практического или теоретического освоения действительности. Р. Декарт писал в своем философском учении: «Под методом же я разумею достоверные и легкие правила, строго соблюдая которые, человек никогда не примет ничего ложного за истинное» и сможет добывать новое знание — все, что он способен познать, — «без излишней траты умственных сил»²⁾. Р. Декарт считал, что для реализации правил его рационалистического метода необходима *интуиция*, с помощью которой усматриваются первые начала (принципы), а затем с помощью дедукции получаются следствия из этих начал. Но нужна не только интуиция, но и *инсайт*, и *воображение*, — другими словами, все свойства (или качества) интеллекта, не укладывающиеся в его рациональную составляющую (*B*).

В чем отличие хорошей программы от плохой? Чем отличается добротной сделанная динамическая модель (даже миниатюрная) от «поделки»? В хорошей программе нет никаких излишеств ни на уровне абстракций сущности, ни на уровне абстракций поведения. Хорошая программа стремится (состоит) к возникновению и организации структур с простыми и четкими формами, к простым и устойчивым состояниям. То есть она функционирует «без излишней траты умственных сил»! И опять-таки выделим главное: в результате исследования (экспериментальной деятельности с программой) достигается соответствие между динамической моделью и объектом, для которого она строится (*C*).

¹⁾ Имеются в виду все типы абстракции: абстракция отождествления, или обобщающая абстракция; абстракция аналитическая, или изолирующая; абстракция идеализирующая, или идеализация; абстракция актуальной бесконечности (отвлечение от принципиальной невозможности зафиксировать каждый элемент бесконечного множества, т. е. когда бесконечные множества рассматриваются как конечные); абстракция потенциальной осуществимости (предполагается, что может быть осуществлено любое, но конечное число операций в процессе деятельности).

²⁾ Декарт Р. Избранные произведения. М.: Госполитиздат, 1950.

3.

По Ж. Пиаже, интеллект, подобно всем биологическим функциям, является продуктом эволюционной адаптации, поэтому для его понимания следует изучать умственную деятельность от момента рождения, наблюдая за его развитием и изменением в процессе взросления.

Два основных принципа интеллектуального роста ребенка: организация (А) и адаптация (В). *Адаптация* (приспособление к условиям окружения) состоит из ассимиляции и аккомодации. *Организация* — это структуризация интеллекта. Наиболее простой уровень такой структуризации — *схема*, являющаяся мысленной репрезентацией некоторого действия (физического или мысленного), выполняемого над объектом. Для новорожденного сосание, хватание, смотрение — это схемы, т. е. его способы познания окружающего мира и воздействия на этот мир. С развитием интеллект ребенка структурируется, или организуется, со все возрастающей сложностью и степенью интеграции. Кроме того, как схемы действий, так и вновь образуемые структуры все более интериоризируются, т. е. начинают совершаться в его голове как быстрые, короткозамкнутые мысленные последовательности.

Еще один фундаментальный аспект теории Ж. Пиаже — *знание есть действие*: «Познание начинается с действия, а всякое действие повторяется или обобщается (генерализуется) через применение к новым объектам, порождая тем самым некоторую «схему»... Основная связь, лежащая в основе всякого знания, состоит не в простой «ассоциации» между объектами (поскольку это понятие отрицает активность субъекта), а в «ассимиляции» объектов по определенным схемам, которые присущи субъекту. Этот процесс является продолжением различных форм биологической ассимиляции, среди которых когнитивная ассимиляция представляет лишь частный случай и выступает как процесс функциональной интеграции. В свою очередь, когда объекты ассимилированы схемами действий, возникает необходимость приспособления («аккомодации» — (С)) к особенностям этих объектов, это приспособление (аккомодация) является результатом внешних воздействий, т. е. результатом опыта»¹⁾. Знания об объекте определяют в

¹⁾ Пиаже Ж. Психогенез знаний и его эпистемологическое значение // Семиотика: Антология / Сост. Ю. С. Степанов. М.: Академ-Проект, Екатеринбург: Деловая книга, 2001.

конечном счете те действия, которые вы над ним можете совершить.

Согласно Ж. Пиаже, в развитии интеллекта человека можно условно выделить четыре главных периода развития: *сенсомоторная стадия* (от рождения до 2 лет), *дооперациональная стадия* (от 2 до 7 лет); *стадия конкретных операций* (от 7 до 11 лет) и *стадия формальных операций* (от 11 лет и подростковый период). Последний период имеет наибольшее отношение к рассматриваемой нами проблеме. На предыдущей стадии ребенок ограничен координацией конкретных объектов в действительной ситуации. Он все еще не может координировать вероятностные события в гипотетической или более абстрактной формализованной ситуации, не может решать задачи без привязки к непосредственно воспринимаемой реальности. Главным же результатом освоения такой координации является то, что подросток может вызвать в уме системы операций, не присутствующие конкретной наблюдаемой ситуации, он способен координировать мысленные системы в системы более высокого порядка (*иерархия абстракций*). Наступает период умственной зрелости. Именно об этом периоде (стадии) формирования интеллекта идет речь: о периоде формирования и становления гипотетического, абстрактного мышления.

Рассмотрим особенности стадии формальных операций. Это — период формирования того, что называют *теоретическим мышлением*. Его логическими формами (способами отражения действительности посредством взаимосвязанных абстракций) являются *понятия, суждения и умозаключения*. Они отражают как бы результат логических действий ума при решении проблемы. *Проблема* — ситуация, в которой возникают задачи, связанные с интеллектуальной деятельностью; говоря философским языком, — *форма теоретического знания*. Ее содержанием является то, что еще не познано человеком, но что нужно познать (знание о незнании; вопрос, возникший в ходе познания и требующий ответа). Проблемная ситуация, согласно М. Вертгеймеру¹⁾, не является чем-то замкнутым в себе, поэтому она ведет нас к решению, к структурному завершению. Точ-

¹⁾ Макс Вертгеймер (1880–1943) — немецкий и американский психолог, один из основателей и главных теоретиков *гештальт-психологии*. «Гештальт» — это немецкий термин, не имеющий точного английского и русского аналога, но вошедший в научный язык. Этот термин используется для обозначения целого, полной структуры, множества, природа которого не обнаруживается с помощью простого анализа отдельных частей, его составляющих.

но так же решенная задача не должна быть завершенной «вещью в себе». Она снова может функционировать как часть, которая заставляет нас выходить за ее пределы, побуждает рассматривать и осмысливать более широкое поле.

Итак, у нас есть вход (проблема) и есть результат (понятия, суждения и умозаключения, или новый уровень организации интеллекта, по Ж. Пиаже). Запускается («принцип действия», по Ж. Пиаже) процесс мышления («процесс адаптации», по Ж. Пиаже) для достижения этого результата. Этот процесс — его логическая составляющая — состоит из таких основных приемов мышления, как *анализ, синтез, абстрагирование, сравнение, обобщение*. К этому классу понятий относятся также *индукция* (индуктивное обобщение) и *дедукция* (дедуктивный вывод). Но в этом процессе в обязательном порядке есть и нерациональная составляющая, без которой (ибо это процесс адаптации, а не ассоциации с уже познанным!) результат, как правило, недостижим.

А теперь, согласно ранее приведенной схеме доказательства утверждения, установим соответствие (которое было намечено в предыдущем тексте буквами *A, B* и *C*)¹⁾. Мы конструируем программу как иерархию абстракций, порождаем некую схему — возможно ли это без структуризации интеллекта (*A*)? Ответ однозначен! При чем интеллект постоянно как бы тренируется в порождении схем. Мы используем методы, а, по Ж. Пиаже, это не что иное, как адаптация созданной схемы (иерархии абстракций) к исследуемому объекту (*B*). У нашей динамической модели нет согласования с объектом исследования? Начинается творческий процесс поиска причин несоответствия. При этом происходит или аккомодация, или новый виток генерации иерархии абстракций (*C*).

Афоризм, порожденный этой идеей: «Целое (т. е. гештальт) отличается от суммы его частей».

1) Вероятно, существует весьма ограниченное количество законов, управляющих мирозданием и, в частности, процессом познания человеком окружающего мира и себя. Люди говорят о своем, на своем языке, на языке своей предметной области. Но говорят ли они о разном?

Введение

Инструменты, которые мы применяем, оказывают глубокое (и тонкое) влияние на наши способы мышления и, следовательно, на нашу способность мыслить.

Э. Дейкстра

Структура данных — это не что иное, как способ взаимосвязи элементов данных. Другими словами, из элементов данных конструируется некое целое, с некоторыми свойствами, и это целое есть структура данных. Дальнейший шаг — в единое целое связываются как структуры данных, так и выполняемые над ними операции. Это новое целое носит в информатике название «*абстрактные типы данных*» (*abstract data type*), т. е. является множеством абстрактных объектов, представляющих неким образом организованные элементы данных, и определенного на этом множестве набора операций, которые могут быть выполнены над его элементами. Зачем нужна эта цепь абстрагирования? Она позволяет более рационально мыслить при решении проблем с использованием компьютера и в итоге получать результат «с наименьшей затратой умственных сил», — ибо уже не конкретное данное (число, символ и т. д.), а абстрактные типы данных становятся элементарными «кирпичиками» конструирования программного продукта¹⁾. И эта книга — о таких элементарных кирпичиках!

Следует отметить, что данный основополагающий раздел информатики не представлен должным образом в рекомендациях по преподаванию информатики, разработанных

¹⁾ Например, если школьнику достаточно сказать при обсуждении задачи фразу вроде «требуется использовать очередь с приоритетом», и этого оказывается достаточно (продолжать не надо), то вызывает восхищение, с какой скоростью и с каким качеством реализует этот школьник такую очередь с приоритетом. Автор уже не способен «на такие подвиги»...

в ведущем вузе России — МГУ¹⁾, хотя он есть в сокращенном варианте (правда, под названием «Фундаментальные структуры данных») в зарубежном аналоге²⁾ этой работы. В школьном же курсе информатики он если и затрагивается, то только косвенно.

Структура книги

В этой книге описываются абстрактные типы данных — матрицы, списки, стек, очередь, двоичные деревья, множества, очереди с приоритетом, сбалансированные деревья. Материал по каждому из этих типов данных разбит на разделы, и практически все они сопровождаются упражнениями разной степени сложности. Упражнения, отмеченные символом «*», — повышенной сложности.

Необходимые условия для работы с книгой

Начального курса информатики, проводимого на основе программирования³⁾, достаточно для понимания и свободного освоения материала данной книги. Для записи алгоритмов используется ограниченное подмножество возможностей (сведенное до минимума) языка программирования Паскаль. Это подмножество можно назвать «псевдокодом», — но «живым» псевдокодом, ибо при его воспроизведении в системе программирования, основанной на языке Паскаль, получается работоспособный код программы!

Преподавателю

Практически каждый раздел — это материал для одного занятия. Особенность текста книги — в том, что автор старался его сделать «под голос», под реальное изложение материала в аудитории (большинство материалов неоднократно проговаривалось при работе со школьниками и, в рамках предмета «Информатика», со студентами первого курса). Автор стремился достичь в тексте синтеза в единое целое слов, иллюстраций и фрагментов логики. Текст фрагментов кода здесь не есть некое приложение к словам, а является его необходимой и обязательной составляющей. (Сам автор при обсуждении материала книги использовал аудиторию с

1) Преподавание информатики и математических основ информатики для непрофильных специальностей классических университетов. М.: Интернет-ун-т информ. технологий, 2005.

2) Рекомендации по преподаванию информатики в университетах. Computer Curricula 2001: Computer Science: Пер. с англ. СПб., 2002. С. 162.

3) Например, на основе первых частей книги: *Окулов С. М.* Основы программирования. 2-е изд. М.: БИНОМ. Лаборатория знаний, 2005.

проекционным оборудованием. Доска применялась только для написания вспомогательного материала (экспромтов), возникающих при обсуждении. Текст занятия обычно предоставлялся слушателю.)

Школьникам и студентам младших курсов

Информатика как учебный предмет обладает удивительной особенностью. Можно прослушать тысячу лекций, при этом все понимать, и — не знать предмета! Нет, — на вербальном уровне вы, конечно, сможете что-то рассказать, что-то воспроизвести и получить очередную оценку. Но вы не сможете свободно выражать свои мысли на языке, понятном компьютеру, вы не сможете окунуться в мир отладки и поиска ошибок, в мир экспериментов, т. е. в ту атмосферу творчества, мучений, озарений, которая присуща деятельности при создании работающих программ, пусть даже и миниатюрных! А ведь только после этих самостоятельных мучений (а мучение, как и мысль, может быть только ваше!) приходит истинное знание предмета, когда все эти списки, очереди, деревья становятся элементарными конструкциями вашей мыслительной деятельности при решении проблем. Материал книги предоставляет возможность для такой самостоятельной деятельности, ибо, во-первых, он написан простым и доступным языком и, во-вторых, текст программных фрагментов хотя и работоспособен, но не является завершенным («набрал, запустил, получил»), и для получения результата вам необходимо будет еще приложить определенные усилия.

Благодарности

Хочу сказать огромное спасибо всем студентам и школьникам, проявившим терпение и выдержку при виде ошибок, которые автор допускал (не всегда по злому умыслу) при чтении этого курса, а также выносившим все экспромты автора и его заблуждения.

Матрицы

Пожелав, чтобы все было хорошо, а худого, по возможности, ничего не было, бог каким-то образом все подлежащее зрению, что застал не в состоянии покоя, а в нестройном и беспорядочном движении, из беспорядка привел в порядок, полагая, что последний всячески лучше первого.

Платон

1.1. Основные понятия

Многие вещи нам непонятны не потому, что наши понятия слабы, но потому, что сии вещи не входят в круг наших понятий.

Козьма Прутков

Матрица A размера $n \times m$ есть прямоугольная таблица чисел, содержащая n строк одинаковой длины (или m столбцов одинаковой длины). Матрица A записывается в виде

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}$$

или, сокращенно,

$$A = (a_{ij}),$$

где $i = 1, \dots, n$ — номер строки, а $j = 1, \dots, m$ — номер столбца. Матрица, у которой $n = m$, называется *квадратной*.

Элементы $a_{11}, a_{22}, \dots, a_{nn}$ образуют *главную диагональ* матрицы A .

Матрица, все элементы которой равны нулю, называется *нулевой*, а квадратная матрица, на главной диагонали которой стоят единицы, а все остальные элементы равны нулю, называется *единичной* (обозначим ее как I).

Представление матриц в памяти компьютера

Напомним уже известные вам факты. Память компьютера представляет собой последовательность ячеек, имеющих уникальные адреса, и возможен прямой доступ к содержимому ячейки памяти, если известен ее адрес. Способ вычисления адреса любого элемента одномерного массива определяется на основании *дескриптора*, который создается при описании этого массива и содержит данные о его физической структуре.

Пусть имеется описание

```
Var V: Array[i..k] Of Integer.
```

Тогда дескриптор для этого массива имеет структуру, показанную в табл. 1.1.

Таблица 1.1

Имя	V
Адрес начального элемента (физический адрес первого элемента массива)	$address(V[i])$
Индекс начального элемента	i
Индекс конечного элемента	k
Тип элемента	Integer
Длина элемента	$w (=2)$

Адрес элемента V с индексом j вычисляется по следующей формуле: $address(V[j]) = address(V[i]) + (j - i) * w$.

Для представления матрицы в памяти компьютера обычно используется двумерный массив, каждый элемент которого идентифицируется парой индексов, где первый индекс определяет номер строки, а второй — номер столбца, на пересечении которых расположен заданный элемент.

Существует два способа отображения логической структуры массива в физическую: *отображение по строкам* и *отображение по столбцам*. В первом случае элементы массива располагаются в памяти последовательно строка за строкой, а во втором — столбец за столбцом. Какое именно представление используется — по строкам или столбцам, — зависит от конкретной системы программирования, так как оно однозначно определяет функцию вычисления адреса

[. . .]

РАЗВИТИЕ ИНТЕЛЛЕКТА ШКОЛЬНИКОВ

Абстракция, абстрагирование – одна из составляющих мыслительного процесса творческой личности. Для развития этого компонента мышления в процессе обучения информатике есть дополнительные возможности, так как знание абстрактных типов данных, умение оперировать ими – это необходимый элемент профессиональной культуры специалиста, связанного с разработкой программных комплексов.

Книга предназначена для школьников, преподавателей информатики и студентов младших курсов университетов. Она может быть использована как в обычных школах при проведении факультативных занятий, так и в образовательных учреждениях с углубленным изучением информатики.



ОКУЛОВ Станислав Михайлович

Декан факультета информатики Вятского государственного гуманитарного университета, кандидат технических наук, доктор педагогических наук, профессор. Автор 9 изобретений и автор (соавтор) 14 книг по информатике для школьников и студентов.

Область интересов: развитие интеллектуальных способностей школьника при активном изучении информатики, исследование ассоциативных систем обработки информации.

С 1993 по 2003 год деятельность в вузе совмещал с работой учителя информатики. За это время его ученики отмечены 33 дипломами (1-й и 2-й степени) на Российских олимпиадах школьников по информатике; трое из них представляли Россию на международных олимпиадах.