

---

**ИССЛЕДОВАНИЯ В ОРГАНИЗАЦИОННОЙ ПСИХОЛОГИИ**

---

## Исследование способности к усвоению искусственных языков у программистов

**ОРЕЛ Екатерина Алексеевна**

*канд. психол. наук, Национальный исследовательский университет «Высшая школа экономики», Москва, Россия, eorel@hse.ru*

Профессиональная деятельность программиста предполагает работу с большим количеством искусственных языков. Часто перед профессионалами стоит задача в короткие сроки освоить новую программную среду, что требует от них высокого уровня развития вербального интеллекта. Этот факт подтверждается многими исследованиями. В настоящей работе мы исследуем динамику усвоения искусственных языков программистами, по сравнению с представителями другой профессиональной группы и обычными пользователями компьютера. Нам удалось показать, что процесс усвоения искусственных языков у программистов протекает особым образом и содержит стилевые особенности, определяемые требованиями профессиональной деятельности.

**Ключевые слова:** психология программирования; искусственный язык; интеллект; профессиональная деятельность.

Психология программирования – это наука о действиях человека, работающего с вычислительными и информационными системами [Шнейдерман, 1984], область психологии профессий, появившаяся в 60-х годах прошлого века. В ее рамках изучаются психологические особенности создания программного обеспечения и, чуть шире, проблемы взаимодействия человека и компьютера в том, что касается работы профессиональных пользователей. Поскольку информационные технологии – постоянно развивающаяся область знания, предмет исследования психологии программирования никогда не будет исчерпан.

Можно выделить следующие основные области исследования в рамках психологии программирования:

- 1) усовершенствование практики программирования: исследования, направленные на то, как облегчить практику программирования, применив к ней законы функционирования человеческой психики [Cheney, 1984, Shneiderman, Meyer, 1979];
- 2) усовершенствование методов программирования: исследование психологических особенностей программистов с тем, чтобы обосновать ввод в практику новых методов и языков программирования [Navarro-Prieto, Canas, 2001; Blackwell et al., 2001];
- 3) разработка методов оценки качества программного обеспечения: исследования того, как программное обеспечение – результат работы программистов – внедряется в практику. Исследования этой группы очень близки к области юзабилити и чаще всего рассматриваются вместе [Carver, 1989];
- 4) разработка методов оценки эффективности работы программистов и совершенствование методов обучения этой специальности: в основном эта область направлена на то, чтобы

создать метод оценки способностей к программированию, не зависящий от профессиональных знаний тестируемого [Di Presio et al., 1980].

### **Способы оценки эффективности работы программиста**

Вопрос оценки различных параметров качества работы программиста был поставлен фактически с начала массового внедрения информационных технологий в жизнь общества. Кого можно назвать хорошим программистом? Того, кто пишет больше программного кода? Но компактность программы часто является показателем ее качества [Dijkstra 1976]. Кто создает больше программ за единицу времени? Но программы несравнимы по масштабам, объему и степени сложности.

Одно из первых определений продуктивности работы программиста было дано Альбрехтом [Albrecht, 1979]. Он рассматривал продуктивность как совокупность функциональных точек, включающих количество вводов, выводов, файлов и запросов. Еще один часто используемый показатель предложили Конт, Дансмор и Шен: они использовали количество строчек исходного кода, создаваемого программистом за месяц работы. Из этого определения вытекает, что продуктивность обратно пропорциональна усилию, измеренному в единицах времени [Dunsmore, Gannon, 1980].

Исходя из этих определений получается, что самый главный вопрос при оценке эффективности работы программиста заключается в том, как снизить временные затраты на создание программы и при этом получить надежный, стабильно работающий результат. Традиционно процесс разработки программного обеспечения описывается следующей последовательностью этапов: создание спецификаций или технического задания, общий дизайн программы, создание программного кода, тестирование программы, ее внедрение и поддержка [Bell et al., 1987; Sommerville, 1985; Вирт, 2001; Иванова, 2002]. Максимум активности программиста приходится на создание программного кода, тестирование, внедрение и поддержку [Walston, Felix, 1977]. Так, по некоторым оценкам, если из приведенной выше последовательности этапов исключить техническую поддержку, то на тестирование программы уходит примерно 50% усилий программиста (по данным [Bell et al., 1987]).

В связи с этим неудивительно, что большинство эмпирических исследований и теоретических моделей деятельности по созданию программного обеспечения сосредоточены именно на указанных выше этапах (например, [Шнейдерман, 1984; Корнилова, Тихомиров 1990; Бабаева, Войскунский, 2003]). А если конкретнее – на этапе создания программного кода, поскольку, по данным многих исследований, именно он в работе программиста оказывается ключевым. Тот результат, который специалист получит на этапе написания кода, будет в наибольшей степени определять успешность всей деятельности по созданию программного обеспечения [Орел, 2006].

### **Семантико-синтаксическая модель Б. Шнейдермана**

Бен Шнейдерман – один из основных исследователей в психологии программирования. В конце 70-х – начале 80-х годов прошлого века он создал семантико-синтаксическую модель мыслительной деятельности программиста. В рамках этой модели было проведено много исследований, подтвердивших ее эмпирическую состоятельность (см., например, [Di Persio, 1980; Shneiderman, Mayer, 1979; Vessey, 1987]).

Как пишет Шнейдерман, модель, которая претендует на более или менее полное описание психологической структуры деятельности программиста, должна учитывать основные задачи, которые решает специалист [Шнейдерман, 1984]. Кроме того, модель должна быть пригодна для описания этих задач в терминах:

- мыслительных структур, которые программист хранит или должен хранить в своей памяти;
- мыслительных процессов, применяемых при использовании этих знаний или при их расширении.

Мыслительные структуры – это сложные многоуровневые знания о понятиях и методах программирования. Шнейдерман предлагает разделить их на две группы:

- *семантические знания* – относятся к общим существенным для программирования понятиям, но не связаны с каким-либо языком программирования. В семантические знания входит определение понятий нижнего уровня (представляет собой оператор присваивания, что такое индексированный массив, какие бывают типы данных); среднего уровня (обмен содержимым между регистрами, стратегии поиска большего из двух значений, суммирование массива); высокого уровня (двоичный поиск, использование стека для организации рекурсии, методы сортировки и слияния). Все эти знания получены из опыта работы, но хранятся в виде общих содержательных наборов информации, более или менее независимых от знания конкретных языков или средств программирования;
- *синтаксические знания* – относятся к конкретным практическим знаниям. По своим свойствам они точнее, подробнее и относительно легче забываются семантических знаний. В эту группу входят такие элементы, как формат повторения, условный оператор и оператор присваивания, правильные наборы символов и имена библиотечных функций. Они в значительной степени приобретаются в процессе осмысленного обучения, включающего решение задач и изучение описательных руководств, помогающих закрепить и усвоить новые понятия совместно с имеющимися семантическими знаниями или «мыслительными структурами».

В рамках этой модели было проведено большое количество исследований, подтвердивших ее на уровне эмпирических данных и позволивших усовершенствовать процесс обучения программированию. Так, серия экспериментов, проведенных Ди Пресью, Исбистером и Шнейдерманом, показала, что тесты на запоминание и воспроизведение информации, включающие оценку мыслительных структур программиста, – хорошие предикторы успешности его работы [Di Presio et al., 1980]. Другой эксперимент, в котором профессиональных программистов просили воспроизвести программу по имеющемуся образцу, показал, что информация о программе хранится в большей степени в семантических единицах, соотносимых с предыдущим опытом и уже имеющимися знаниями, чем в синтаксических формах, в виде которых программа была представлена в исходном образце [Atwood, Ramsey, 1977]. В целом в рамках этой модели было проведено большое количество экспериментов на реконструкцию программного кода, и все они приходили к более или менее сходным результатам [Sheppard et al., 1979; Curtis et al., 1989; Chaudhary, 1979; Norcio, Kerst, 1983; McKeithen et al., 1981; Canast et al., 1994].

### **Модель принятия интеллектуальных решений в диалоге с компьютером, разработанная О.К. Тихомировым и Т.В. Корниловой**

Эта модель построена на теоретическом аппарате теории деятельности А.Н. Леонтьева [Леонтьев, 1975]. Модель О.К. Тихомирова и Т.В. Корниловой – это попытка дать комплексную картину деятельности, опосредованной компьютером, которая учитывает не только когнитивный, но и мотивационный ее аспект [Корнилова, Тихомиров, 1990]. Однако стоит оговориться, что она не универсальна, уже хотя бы в том смысле, что далеко не все этапы деятельности программиста можно описать с помощью понятия «диалог с компьютером».

Согласно этой модели деятельность в диалоге с компьютером специфична уже в силу своей особой структуры: она состоит из последовательного чередования этапов «запроса» (пользователь обращается к компьютеру) и «ответа» (компьютер выдает пользователю результат обработки пользовательского запроса). По мнению авторов модели, в отличие от других форм внешнего опосредования, применение компьютерных средств требует перестройки интеллектуальных действий человека во внутреннем плане [Корнилова, Тихомиров, 1990, Тихомиров и др., 1982].

Компьютер как средство интеллектуальной деятельности предъявляет особые требования к организации мышления пользователя. Действия пользователя – как профессионального, так и непрофессионального – в диалоге с компьютером имеют одну существенную характеристику: решение задачи пользователем возможно только в случае учета им логики используемого средства [Тихомиров, Бабанин, 1986]. Но поскольку обратная связь от компьютера – это внешнее средство организации деятельности, в диалоге с компьютером у пользователя есть возможность выбрать разные способы решения одной и той же задачи [Тихомиров, Гурьева, 1986].

В зависимости от задачи, стоящей перед пользователем, мышление человека в диалоге с компьютером можно рассматривать в континууме приближения от алгоритмизированных процедур ко все более творческим видам мышления [Тихомиров и др., 1981].

На становление внутренней, или психологической, структуры интеллектуальных решений при взаимодействии с компьютером в режиме диалога наиболее существенное влияние оказывают две группы факторов [Тихомиров, Гурьева, 1986]:

- 1) «язык» диалога как форма организации «запросов – ответов», представленной конкретными внешними условиями получения компьютерных данных при доопределении альтернатив и критериев ПР; им задаются как возможности, так и ограничения деятельности пользователя в диалоге;
- 2) внутренние условия, представленные механизмами психологической регуляции деятельности пользователя.

Эта модель имеет скорее теоретическое, чем прикладное, значение. Она не отвечает на конкретный практический вопрос, какими качествами должен обладать успешный программист, но служит базисом для интерпретации результатов дальнейших исследований.

### **Современные исследования в области психологии программирования**

По ссылкам на литературные источники нетрудно заметить, что пик исследований в психологии программирования пришелся на 80-е годы прошлого века. В дальнейшем их количество постепенно шло на убыль, однако некоторые исследования можно найти и сегодня.

В настоящее время общей тенденцией в программировании можно назвать постепенное приближение искусственных языков к естественным. Языки программирования – это способы, с помощью которых человек передает компьютеру свои представления о том, что он должен сделать. Поэтому неудивительно, что сообщество программистов пришло к выводу о том, что при разработке новых языков программирования необходимо учитывать особенности мышления и речи человека.

Однако, чтобы человек мог общаться с компьютером на естественном для себя языке, необходимо научить компьютер его понимать. Первый шаг к этому – добиться четкого понимания, как люди используют естественный язык в общении. Этому, к примеру, посвящено исследование Джона Пейна, который попытался разобраться, как непрограммисты используют язык в представлении алгоритмов оперирования с изображениями и числами [Paine, 2000, 2002]. Согласно одному из полученных им результатов, люди – и дети, и взрослые – в естественном языке смешивают значения логических операторов И, ИЛИ и НЕ. Например,

предлог «и» часто используется там, где с точки зрения логики должен быть оператор ИЛИ. Например, во фразе «проходным назначается балл 95 и выше» на самом деле подразумевается оператор ИЛИ, потому что балл не может одновременно принимать значения 95 и, скажем, 120 (при условии, что эти два значения располагаются на одной и той же шкале) [Pane et al., 2001].

Другое современное направление – исследование преимуществ визуальных языков программирования (языков, в которых объекты и их связи представлены в графическом виде) по сравнению с текстовыми. Так, испанские исследователи показали, что визуальный язык легче для понимания, а следовательно, и для обучения, и его использование требует задействования меньшего количества мыслительных структур [Navarro-Prieto, Canas, 2001]. К сходным выводам в исследовании визуальных языков программирования пришли английские исследователи, изучавшие продуктивность программистов, использовавших визуальные языки, по сравнению с текстовыми [Blackwell et al., 2001].

Некоторое количество современных исследований посвящены изучению процессов понимания языков программирования [Good, Brna, 2004]. Трудность этого направления работы заключается в том, что процессы понимания глубоко интериоризированы, и нужно использовать специальные техники и методические приемы, чтобы попытаться вывести их на сознательный уровень [Von Mayrhauser, Lang, 1999]. Брукс в 1983 г. первым предложил рассматривать понимание программ как нисходящий процесс и построил его временную схему [Brooks, 1983]. Так, Майрхаузер и Лэнг разработали специальную схему анализа протоколов вербального общения между программистами, однако она позволяет анализировать только то, что выносится в диалог [Von Mayrhauser, Lang, 1999]. О'Брайан использовала сходную схему анализа вербальных высказываний, чтобы изучить различные стратегии понимания программы [O'Brien et al., 2001].

В целом большинство современных исследований в этой области связаны с изучением того, как программист работает с искусственными языками, которые за все время существования программирования все больше приближаются к естественным. Исследователи в этой сфере пытаются решить насущную проблему: каким должен быть язык программирования, чтобы обеспечить максимальную эффективность работы программиста?

Наше исследование тоже можно отнести к этой группе. В нем мы пытаемся ответить на вопрос: какова динамика усвоения искусственного языка программистами? Мы предполагаем, что различные особенности профессиональной деятельности накладывают на этот процесс свой отпечаток, поэтому программисты будут работать с искусственным языковым материалом по-другому, чем представители других профессий и простые пользователи компьютера.

Из профессиографического описания деятельности по созданию программного обеспечения, а также из результатов, полученных нами в исследовании структуры интеллекта программистов [Орел, 2007], следует, что одним из главных факторов, определяющих успешность работы программиста, являются вербальные способности. В самом деле, работа со схемой программы как с совокупностью знаков составляет один из ключевых этапов деятельности по написанию программы. А владение грамматикой языка, знание синтаксических правил, по которым строится высказывание, позволяет специалисту уделять основное внимание логике программы, ее внутренней структуре, а не расходовать ресурс внимания на поиск пропущенной запятой. Интуитивно понятно, что легкость в оперировании командами искусственного языка, приспособленность к усвоению формальных, в чем-то каждый раз новых и необычных правил в каждом новом языке программирования – важнейшая составляющая профессионально-специфических особенностей мыслительной деятельности программистов.

## Методология исследования

Предыдущее исследование, в котором были показаны различия в структуре интеллекта программистов и представителей других профессиональных групп, продемонстрировало, что изучаемая профессиональная группа значительно превосходит другие по показателям вербальных способностей и логического мышления [Орел, 2007]. Однако статический срез способностей не отражает того, как складывается и развивается их структура, способствующая успеху в работе программиста. Особенно интересным нам кажется проследить, как проходит усвоение новых языковых структур у разработчиков программного обеспечения по сравнению с представителями других профессий. В исследовательском плане был поставлен вопрос о том, не является ли способность к быстрому усвоению формальных языков ключевым фактором в структуре вербальных способностей программиста.

Главной *целью* настоящего исследования стало конструирование методики, позволяющей создать компактную модель динамики овладения искусственным языком у разных профессиональных групп. В рамках достижения поставленной цели необходимо осуществить проверку следующих *гипотез*:

- 1) в группе программистов результаты методики усвоения искусственного языка будут выше, чем результаты, которые демонстрирует случайная выборка;
- 2) в группе программистов результаты методики усвоения искусственного языка будут выше, чем результаты, которые демонстрирует выборка представителей социомической профессии.

Для проверки выдвинутых гипотез был проведен формирующий квазиэксперимент, позволяющий проследить динамику усвоения искусственного языка специалистами различных профессиональных групп.

### Методика

Специально для проведения настоящего исследования был разработан «Тест усвоения искусственного языка» (Artificial Language Mastering Test – ALMT). Он состоит из 40 однотипных вопросов, в каждом из которых респондент решает, правильно ли построена «фраза» с точки зрения грамматики искусственного языка. При этом правила, по которым строятся высказывания, испытуемому не сообщаются. После каждого задания испытуемому дается оперативная обратная связь: правильно или не правильно он ответил на поставленный вопрос. Эта обратная связь становится основой для его дальнейших ответов. Таким образом, усвоение искусственного языка, заложенного в тест ALMT, происходит прямо в процессе его выполнения.

#### *Грамматика искусственного языка*

Все слова искусственного языка, используемого в методике, состоят из трех букв, а предложения могут содержать всего три члена: подлежащее, сказуемое и дополнение.

Грамматика этого языка описывается следующими правилами:

- 1) все предложения построены в одинаковом порядке: «подлежащее – сказуемое – дополнение»;
- 2) одно и то же слово всегда является одной и той же частью речи (т.е. части речи и члены предложения всегда совпадают);
- 3) слово, обозначающее подлежащее, всегда имеет следующий порядок букв: гласная, согласная, гласная;
- 4) слово, обозначающее сказуемое, всегда имеет следующий порядок букв: гласная, согласная, согласная;

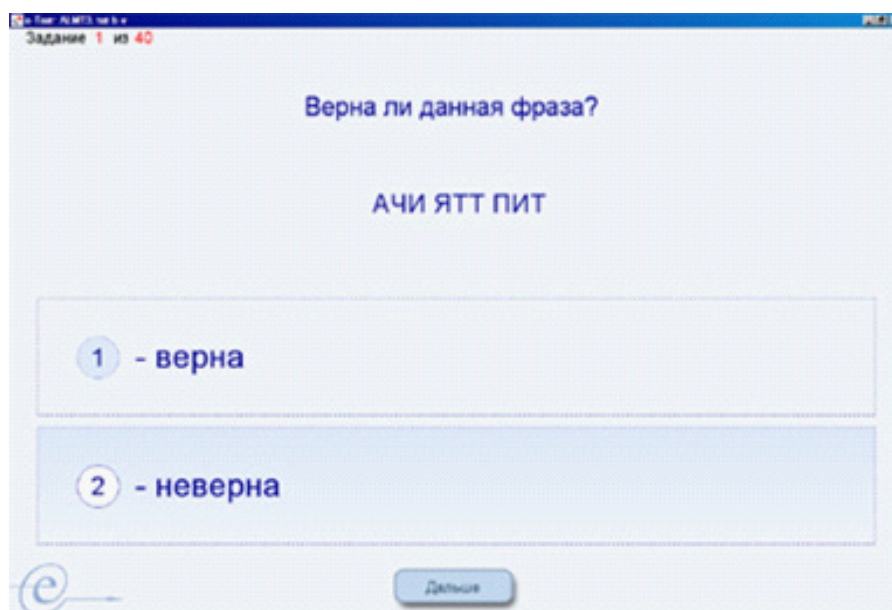


Рис. 1. Пример одного из заданий ALMT

5) слово, обозначающее дополнение, всегда имеет следующий порядок букв: согласная, гласная, согласная.

ALMT был создан в компьютерном варианте на базе универсальной инструментальной оболочки e-Test. После запуска программы испытуемому предъявлялась инструкция к тесту, затем – инструкция по работе с программой. Далее он должен был заполнить паспортную форму с некоторыми сведениями о себе (ФИО, пол, возраст, профессия). После этого запускался тест. На рис. 1 показано, как выглядит задание теста на экране компьютера.

После того как испытуемый выбрал ответ, программа переходила к следующему экрану. На нем отображался следующий вопрос, а в самом верху экрана – обратная связь по предыдущему вопросу: на 5 секунд появлялась одна из двух надписей: «Предыдущий ответ верен» зеленого цвета или «Предыдущий ответ неверен» красного цвета. Возвратиться к предыдущему вопросу и изменить ответ было невозможно. Компьютеризация процедуры тестирования позволила нам фиксировать следующие параметры:

- 1) количество верных ответов;
- 2) количество ошибок;
- 3) время, затраченное на работу с методикой в целом;
- 4) время, затраченное на работу с каждым вопросом.

В качестве обратной связи респондент получал количество верных ответов и текстовую интерпретацию, в которой описывался уровень развития способности усваивать искусственные языки.

Следует отметить, что в нашу задачу не входило моделирование языка программирования. Важнее было выяснить, действительно ли программисты лучше справляются с языковыми задачами подобного рода. Сходство искусственного языка ALMT и языков программирования состоит в том, что в обоих случаях структура языка подчиняется жестким правилам, без исключений. Правильным в данном контексте может быть единственный вариант высказывания (тогда как в естественном языке понятным – и в этом смысле более удачным – высказыванием может быть и неверное с грамматической точки зрения предложение).

Различий между искусственным языком ALMT и языками программирования больше, чем сходств: языки программирования имеют куда более широкий свод правил, с их помощью конструируется больший диапазон высказываний (команд). Усвоение языка также проис-

ходит по-разному в случае работы с нашей методикой и при работе с языками программирования. Если при изучении языков программирования правила задаются с самого начала, то в нашем эксперименте респондент усваивал правила языка в процессе его «использования». Однако это не является ограничением настоящего исследования. Наша основная цель заключалась не в том, чтобы имитировать условия изучения языка программирования, а в том, чтобы проследить динамику усвоения искусственных языковых конструкций.

### **Выборка**

В исследовании приняли участие три группы испытуемых.

*1. Основная группа.* Программисты – 21 человек от 24 до 56 лет: 19 мужчин, 2 женщины. Выборка состоит только из специалистов, которые решают сложные задачи, связанные с высоким уровнем ответственности. Из них: 6 сотрудников крупной инвестиционной компании, занимающихся разработкой системы биржевых торгов через интернет, и 15 программистов из разных регионов России, обеспечивающих проведение Единого государственного экзамена. Модальный возраст – 34 года, средний опыт работы по специальности – 8 лет и 4 месяца. Все респонденты проходили тестирование на добровольной основе.

*2. Фоновая группа.* Группу представителей социономической профессии составили студенты-психологи 3-го курса, всего 18 человек. Именно эта профессиональная группа была выбрана не только по принципу ее доступности. Дело в том, что работа с текстами (как в учебной, так и в профессиональной деятельности психолога) играет достаточно важную роль. Проведенное ранее исследование структуры интеллекта представителей различных профессиональных групп показало, что вербальные способности у психологов выражены на достаточно высоком уровне. А значит, они тоже оказываются профессионально важными качествами. Существенное отличие от программистов, однако, заключается в том, что, во-первых, психологи работают с естественным языком, а во-вторых, от них не требуется усваивать новые языки в процессе работы. То есть в силу гибкости естественного языка психологам нет необходимости подчиняться строгим грамматическим правилам, в отличие от изучаемой нами профессиональной группы.

*3. Фоновая группа 2.* Чтобы учесть искажения, вносимые профессиональной принадлежностью участников исследования, была включена вторая фоновая группа, собранная по случайному принципу из пользователей интернета. Ее финальный размер составил 39 человек различной профессиональной принадлежности (однако и программисты, и психологи, попавшие в эту группу, были исключены из анализа).

Основная и первая фоновая группы (программисты и психологи) проходили тестирование в присутствии исследователя, в отдельном помещении, так, чтобы постороннее вмешательство не отвлекало респондента от работы с методикой. Вторая фоновая группа (случайные пользователи) проходили тестирование через интернет.

По каждому респонденту собирались следующие результаты:

- 1) ответы респондента на каждый вопрос теста;
- 2) время, затраченное респондентом на ответ каждого вопроса теста;
- 3) для каждого испытуемого подсчитывалось количество правильных ответов – количество совпадений с ключом. Количество верных ответов рассматривалось нами как показатель усвоения искусственного языка.

Чтобы выяснить, существуют ли значимые различия между основной и фоновыми группами по уровню усвоения языка методики, были проведены парные сравнения между группами с помощью критерия Манна-Уитни (непараметрический аналог  $t$ -критерия Стьюдента для независимых выборок). Для анализа различий в динамике усвоения искус-



ственного языка разными группами был использован критерий Краскела–Уоллиса для нескольких независимых выборок.

## Результаты

Успешность усвоения искусственного языка различными группами респондентов. Сравнение групп программистов и студентов-психологов по уровню усвоения искусственного языка по критерию Манна–Уитни показало, что значимых различий между этими двумя группами не наблюдается ( $U = 139.5$ ,  $p$ -value = 0.162). Сравнение групп программистов и случайных пользователей по уровню усвоения искусственного языка, напротив, продемонстрировало значимые различия между сравниваемыми группами ( $U = 209.0$ ,  $p$ -value = 0.002).

### Динамика усвоения искусственного языка

Структура полученных нами данных позволила проанализировать, как происходило усвоение используемого в методике искусственного языка различными группами. Наглядное отображение этой динамики представлено на рис. 2. Графики построены по усредненному количеству правильных ответов на каждые 10 заданий теста.

Как видно из приведенного графика, программисты лучше справляются с заданиями теста уже на первых 10 вопросах. На первые 10 вопросов теста программисты дали больше верных ответов, чем фоновая группа случайных пользователей. И в дальнейшем эта тенденция сохраняется. Интересно проанализировать также и кривую, по которой идет научение на нашем материале. У программистов, несмотря на более высокие результаты с самого начала, прогресс в научении начинается с третьего десятка заданий: после 20-го вопроса количество верных ответов неуклонно растет.

Иная картина складывается на выборке случайных пользователей. У них прогресс начинается сразу же, однако после 30-го вопроса количество верных ответов не увеличивается (несмотря на то что максимум – 10 из 10 – достигнут не был).

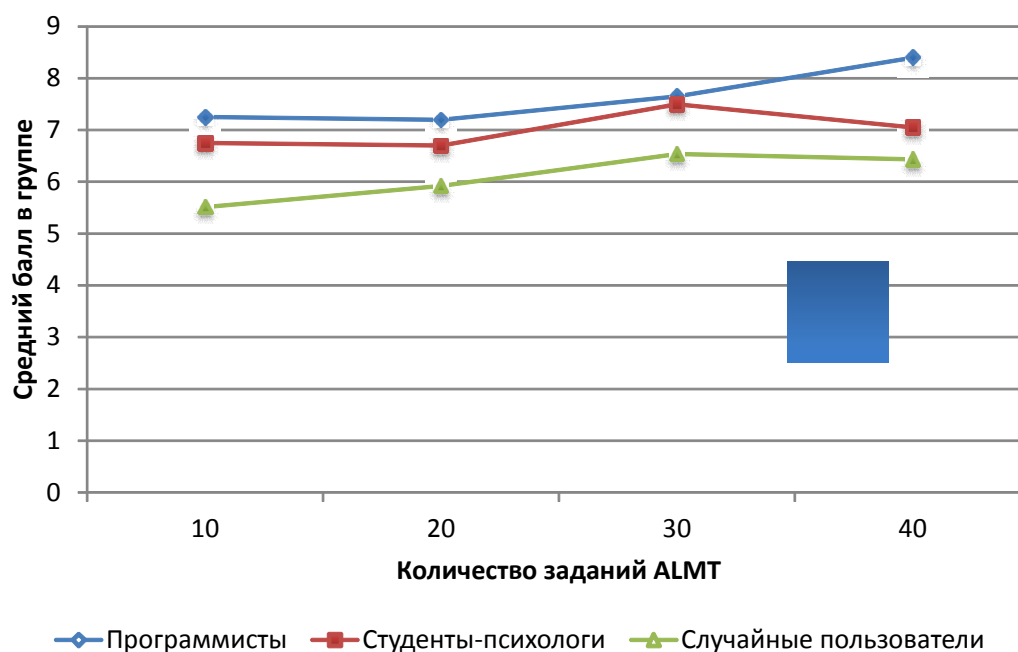
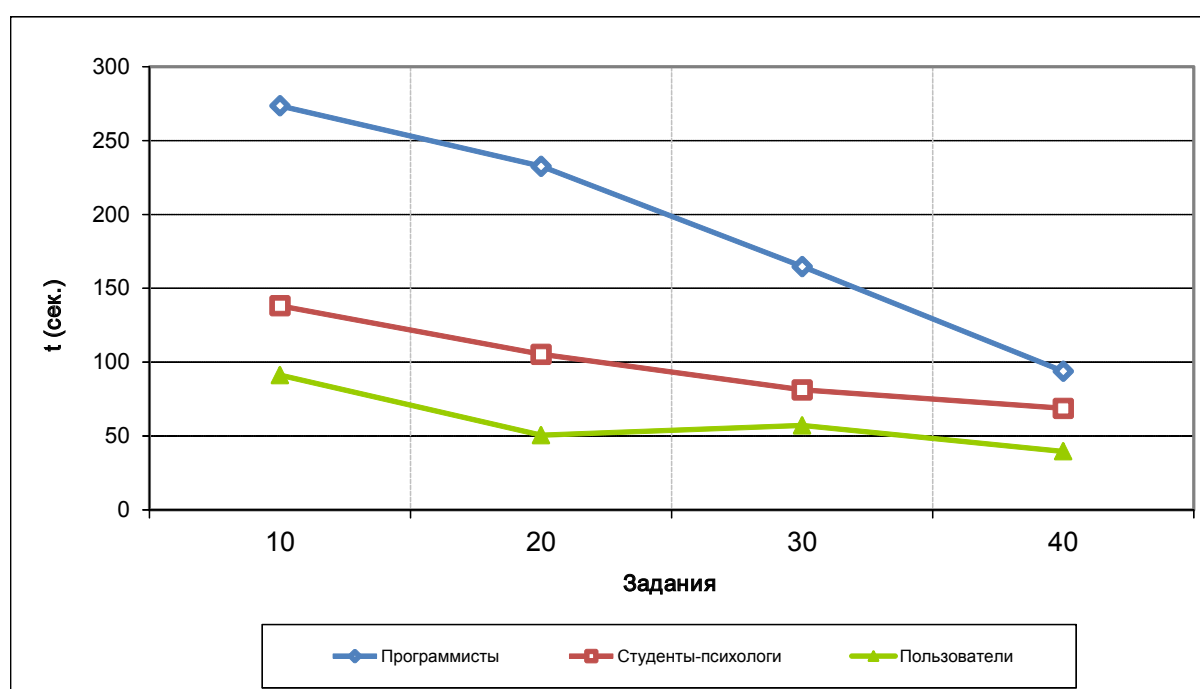


Рис. 2. Сравнительная динамика усвоения искусственного языка в основной и фоновых группах

**Таблица 1.** Результаты сравнения выборок по времени, затраченному на решение задач (критерий Краскела–Уоллиса)

Интервал	$\chi^2^*$	p-value
Задания 1–10	12.77	0.002
Задания 11–20	20.41	0.000
Задания 21–30	9.25	0.010
Задания 31–40	9.73	0.008

\* df = 2

**Рис. 3.** Временная динамика решения тестовых задач разными группами респондентов

Несмотря на то что значимых различий по степени усвоения искусственного языка между программистами и студентами-психологами обнаружить не удалось, динамика решения задач теста у этих групп различается, и, как в предыдущем случае, на последних 10 заданиях теста.

Еще один параметр, который нам удалось проследить, – это время, затраченное на решение задач. Различия во времени выполнения теста между разными группами были проанализированы с помощью критерия Краскела–Уоллиса, позволяющего сравнивать несколько независимых выборок. Поскольку нас интересовала динамика изучения языка, было проанализировано время решения каждых 10 задач теста в каждой из трех выборок. Результаты сравнения выборок для каждого из четырех временных интервалов представлены в табл. 1. Из нее видно, что различия по временным затратам на выполнение каждой части теста между изучаемыми группами являются значимыми. Графическое представление результатов (рис. 3) позволяет отметить, что программисты тратили на выполнение заданий теста существенно больше времени, чем представители других исследуемых групп.

Как видно из результатов анализа временной динамики, во всех группах существует тенденция к уменьшению времени, затрачиваемого на решение тестовых задач. Однако

каждая группа демонстрирует индивидуальные временные показатели на каждой десятке вопросов теста.

Основная группа программистов, которая успешнее остальных справилась с заданиями теста, работала в среднем медленнее представителей фоновых групп. Второй результат по времени показала группа студентов-психологов, которая, как описано выше, также занимает второе место по количеству решенных задач. И наконец, быстрее всего с методикой справилась группа случайных пользователей, которые хуже всего решали задачи теста. Таким образом, наблюдается обратная зависимость между успешностью решения тестовых задач и затраченным на это временем.

Помимо этого, у группы программистов время, затраченное на решение первых 10 задач теста, существенно больше времени, затраченного на последние задачи. Другие группы столь резкой динамики не показывают. Более того, из анализа графика прослеживается та же тенденция: чем менее успешно группа в целом справилась с решением тестовых задач, тем меньше разница между временем, затраченным ею на решение первых и последних задач теста.

### Обсуждение результатов

Итак, нам удалось зафиксировать следующее. Во-первых, программисты, хотя бы на уровне тенденции, более успешно справляются с задачей усвоения искусственного языка, и, во-вторых, динамика их работы с искусственно созданным языковым материалом существенно отличается от динамики, продемонстрированной в фоновых группах.

Таким образом, можно сделать вывод о том, что успешность работы с искусственными языками действительно является отличительной чертой для профессии программиста. Однако говорить о направлении этой связи мы, конечно, не можем. На материале проведенного нами исследования невозможно ответить на вопрос, приходят ли в эту профессию люди, изначально более способные к работе с искусственными языками, или эта способность развивается в ходе профессиональной деятельности.

Мы можем выделить две причины различий результатов между выборками. Прежде всего для программиста работа с искусственными языками является одним из аспектов профессиональной деятельности, и, следовательно, она для них более привычна. В предыдущих исследованиях по психологии программирования уже были описаны особенности деятельности программистов [Орел, 2007], повторим только, что работа с языковыми структурами является главной составляющей их профессиональной деятельности. К привычной деятельности можно применить готовые схемы решения, а если они не будут работать, то, скорее, возникнет желание решить задачу, чем бросить ее.

Еще один фактор, по-видимому, повлиявший на результат, – это мотивация респондентов. В одном из предыдущих исследований было показано, что программисты, в среднем, обладают более высоким уровнем эрудиции, чем представители того же класса профессий (Человек – Знаковые системы) [Орел, 2007]. Помимо демонстрации общего высокого уровня знаний в различных областях, эрудиция – это еще и показатель высокой познавательной активности респондента. А это, в свою очередь, означает желание узнать нечто новое, приобрести новые способы и средства решения еще нерешенных задач. То есть у программиста скорее сформируется мотивация решать новые задачи, тем более такие, которые по своей структуре согласуются с их профессиональной деятельностью. К сожалению, нами не проводилась оценка мотивации респондентов во время тестирования, и мы можем опираться только на устные отзывы. Однако наши результаты хорошо ложатся на гипотезу о том, что респонденты, обладающие более слабой мотивацией решить задачу (т.е. решить ее ради того,

чтобы получить решение) и более сильной установкой на самопознание, скорее потеряют интерес к тесту, если сразу не достигнут высоких результатов. Отсюда – и полученное нами расхождение в динамике выполнения теста основной и фоновыми группами.

В случае же выборки случайных пользователей этот фактор работает в меньшей степени просто за счет того, что в ней представлено меньше людей с выраженной познавательной потребностью: поскольку выборка составлена случайным образом, этот тезис работает просто за счет статистических законов. Конечно, все участники контрольной группы проявили достаточный уровень мотивации, чтобы принять участие в исследовании и дойти до конца теста. Однако, поскольку каждый участник тестирования получал обратную связь, мотивация эта, скорее, направлена на самопознание. Тогда как опрошенные программисты, принявшие участие в исследовании, отмечали, что главным фактором, заставившим их пройти исследование до конца, был именно интерес к задаче.

Задачи теста для всех групп были одинаково новые – в том смысле, что именно в таком виде сталкиваться с усвоением искусственного языка никому из участников эксперимента ранее не приходилось. Для программистов задания теста могут только отдаленно напоминать профессиональную деятельность – и тем более вряд ли на сознательном уровне. Таким образом, по-видимому, представители разных групп, участвовавших в тестировании, подходят к решению задач с разными мотивационными установками, что и отражается на времени решения.

Наиболее выраженный мотив разобраться в задаче и решить ее как раз представлен в стратегии, используемой программистами: потратить больше времени на первые задачи, понять принцип их решения. Студенты-психологи тоже демонстрируют такую тенденцию, однако у них она менее выражена, чем у программистов. И наконец, у пользователей интернета временные характеристики прохождения теста наиболее близки к тем, которые проявляются при использовании стратегии случайных ответов. Заметим также, что время решения последних задач у всех трех групп практически одинаковое.

Еще одно возможное объяснение, одновременно открывающее перспективу для будущих исследований, – это различия в когнитивных стилях. Каган [Холодная, 2004] предложил различать импульсивные и рефлексивные когнитивные стили, основное отличие между которыми как раз лежит в области динамики решения мыслительных задач. Те, кто обладает рефлексивным когнитивным стилем, сравнительно медленно вникают в задачу, но потом, разобравшись во всех ее нюансах, сравнительно быстро решают сходные задачи. Они способны посвятить довольно длительное время изучению условий и сознательным поискам метода решения, тогда как обладатели импульсивного когнитивного стиля, скорее, будут действовать методом проб и ошибок [Корнилова, Парамей, 1989]. Согласно некоторым исследованиям, у программистов преобладает рефлексивный стиль [Turkle, 1984], однако сравнительный анализ с другими профессиональными группами, возможно, еще предстоит проделать.

Разработанная экспериментальная методика может использоваться в профориентационных целях для диагностики способностей. Ее очевидные плюсы в том, что решение представленных в ней задач не требует никаких специальных знаний, она занимает в среднем 20 минут и позволяет сделать вывод о способности респондента работать с искусственными языковыми структурами. Методика может быть использована как часть профориентационного комплекса для выявления склонностей и способностей респондента в той или иной профессиональной сфере. Чтобы данную методику использовать в целях отбора, ее необходимо доработать и дополнить более комплексной диагностикой этой сложной способности.

## Литература

- Бабаева Ю.Д., Войскунский А.Е. Одаренный ребенок за компьютером М.: Сканрус, 2003.
- Вирт Н. Алгоритмы и структуры данных. СПб.: Невский диалект, 2001
- Иванова Г.С. Основы программирования. М.: Изд-во МГТУ им. Баумана, 2002.
- Корнилова Т.В., Парамей Г.В. Подходы к изучению когнитивных стилей: двадцать лет спустя // Вопросы психологии. 1989. № 6. С. 140–147.
- Корнилова Т.В., Тихомиров О.К. Принятие интеллектуальных решений в диалоге с компьютером. М.: Изд-во МГУ, 1990.
- Леонтьев А.Н. Деятельность. Сознание. Личность. М., 1975
- Орел Е.А. Вербальные способности как фактор успешности в программировании // Вестник МГУ. Сер. 14. Психология. 2007. № 2. С. 70–79.
- Орел Е.А. Психологическая структура деятельности по созданию программного обеспечения // Личностно развивающее профессиональное образование. Материалы V Международной научно-практической конференции. Часть III. Екатеринбург, 2006
- Тихомиров О.К., Бабанин Л.Н. ЭВМ и новые проблемы психологии. М., 1986.
- Тихомиров О.К., Белавина И.Г., Березанская Н.Б., Богданова Т.Г., Васильев И.А. Психология и практика автоматизации проектирования ЭВМ // Психологический журнал. 1982. № 5. С. 39–54.
- Тихомиров О.К., Белавина И.Г., Войскунский Е.А. Психология и практика программного обеспечения ЭВМ // Вестник МГУ. Сер. 14. Психология. 1981. № 1. С. 3–14.
- Тихомиров О.К., Гурьева Л.П. Психологический анализ трудовой деятельности, опосредованной компьютерами // Психологический журнал. 1986. Т. 7. № 5. С. 13–26.
- Холодная М.А. Когнитивные стили. О природе индивидуального ума. 2-е издание. СПб.: Питер, 2004
- Шнейдерман Б. Психология программирования. Человеческий фактор в вычислительных и информационных системах. Пер. с англ. А.И. Горлина, Ю.Б. Котова. М.: Радио и связь, 1984.
- Albrecht A.J. Measuring application development productivity // Proceedings of IBM Application Development Symp., 1979. P. 83–92.
- Atwood M.E., Ramsey H.R. Cognitive structures in the comprehension and memory of computer programs. Technical Memorandum SAI-77-O25-DEN, Science Applications, Inc. March 8. 1977.
- Bell D., Morrey I., Pugh J. Software Engineering A Programming Approach, Englewood Cliffs, NJ: Prentice-Hall, 1987.
- Blackwell A.F., Britton C., Cox A., Green T.R.G., Gurr C.A., Kadoda G.F., Kutar M., Loomes M., Nehaniv C.L., Petre M., Roast C., Roes C., Wong A., Young R.M. Cognitive dimensions of notations: design tools for cognitive technology // Beynon M., Nehaniv C.L., Dautenhahn K. (eds.) Cognitive Technology. Springer, Berlin, 2001. P. 325–341.
- Brooks R. Towards a theory of the comprehension of computer programs // International Journal of Man-Machine Studies. 1983. Vol. 18 (6). P. 543–554.
- Canas J.J., Bajo M.T., Gonzalvo P. Mental models and computer programming // International Journal of Human-Computer Studies. 1994. Vol. 40. P. 795–811.
- Carver D.L. Programmer variations in software debugging approaches // International Journal of Man-Machine Studies. 1989. Vol. 31(3). P. 315–322.
- Chaudhary B.D. Factors of program complexity and their effects on program comprehension. Thesis submitted to the Department of Electrical Engineering, Indian Institute of Technology Kanpur, 1979.
- Cheney P.H. Effects of individual characteristics, organizational factors and task characteristics on computer programmer productivity and job satisfaction // Information and Management. 1984. Vol. 7(4). P. 209–214.

- Curtis B., Sheppard S.B., Kruesi-Bailey E., Baley J., Boehm-Davies D.A.* Experimental evaluation of software documentation formats // *Journal of Systems and Software*. 1989. Vol. 9. P. 167–207.
- Di Persio T., Isbister D., Shneiderman B.* An experiment using memorization/reconstruction as a measure of programmer ability // *International Journal of Man-Machine Studies*. 1980. Vol. 13. P. 339–354.
- Dijkstra E.W.* A discipline of programming. Prentice-Hall, 1976
- Dunsmore H.E., Gannon J.D.* Analysis of the effects of programming factors on programming effort // *Journal of Systems and Software*. 1980. Vol. 1 (2). P. 141–153.
- Good J., Brna P.* Program comprehension and authentic measurement: a scheme for analyzing descriptions of programs // *International Journal of Human-Computer Studies*. 2004. Vol. 61. P. 169–185.
- McKeithen K.B., Reitman J.S., Rueter H.H., Hirtle S.C.* Knowledge organization and skill differences in computer programmers // *Cognitive Psychology*. 1981. Vol. 13. P. 307–325.
- Navarro-Prieto R., Canas J.J.* Are visual programming languages better? The role of imagery in program comprehension // *International Journal of Human-Computer Studies*. 2001. Vol. 54. P. 799–829.
- Norcio A.F., Kerst S.M.* Human memory organization for computer programs // *Journal of the American Society for Information Science*. 1983. Vol. 34. P. 109–114.
- O'Brien M.P., Shaft T.M., Buckley J.* An Open-Source Analysis Scheme for Identifying Software Comprehension Processes // Kadoda G. (ed.). *Proceedings of PPIG-13: 13th Annual Meeting of the Psychology of Programming Interest Group*, Bournemouth, UK. 2001. P. 129–146.
- Pane J.F.* Programming System for Children that is Designed for Usability. PhD Thesis, Computer Science Department Carnegie Mellon University, Pittsburgh, PA. Computer Science Technical Report CMU-CS-02-127. 2002.
- Pane J.F., Myers B.A.* Tabular and Textual Methods for Selecting Objects from a Group. in *Proceedings of VL 2000: IEEE International Symposium on Visual Languages*. Seattle, WA: IEEE Computer Society. 2000. P. 157–164.
- Pane J.F., Ratanamahatana C.A., Myers B.A.* Studying the Language and Structure in Non-Programmers' Solutions to Programming Problems // *International Journal of Human-Computer Studies*. 2001. Vol. 54 (2). P. 237–264.
- Shneiderman B., Mayer R.* Syntactic/semantic interactions in programmer behavior: a model and experimental results // *International Journal of Computer and Information Sciences*. 1979. Vol. 8(3). P. 219–238.
- Sommerville D.* *Software Engineering*, Wokingham: Addison-Wesley, 1985.
- Turkle S.* *The Second Self: Computers and the Human Spirit*. N.Y.: Simon and Shuster, 1984.
- Vessey I.* On matching programmers' chunks with program structures: An empirical investigation // *International Journal of Man-Machine Studies*. 1987. Vol. 27. P. 65–89.
- Von Mayrhauser A., Lang S.* A coding scheme to support systematic analysis of software comprehension // *IEEE Transactions on Software Engineering*. 1999. Vol. 25. P. 526–540.
- Walston C.E., Felix C.P.* A method of programming measurement and estimation // *IBM Systems Journal*. 1977. Vol. 16 (1). P. 54–73.

# The study of artificial language learning among programmers

**Ekaterina OREL**

*Ph.D., National Research University Higher School of Economics, Moscow, Russia, eorel@hse.ru*

## **Abstract**

Programmer's professional activity requires an amount of work with different artificial languages. Many studies report that effective programming is correlated with the high level of verbal intelligence. In this paper we study the dynamics of artificial language learning among programmers in comparison with psychologists and the group of non-professional users. We show that programmers learn artificial language in a different way, then the other groups, and this difference is based on their professional requirements.

**Keywords:** psychology of programmers; artificial language; intelligence.