Интернет-журнал «Науковедение» ISSN 2223-5167 http://naukovedenie.ru/
Том 7, №3 (2015) http://naukovedenie.ru/index.php?p=vol7-3
URL статьи: http://naukovedenie.ru/PDF/179EVN315.pdf
DOI: 10.15862/179EVN315 (http://dx.doi.org/10.15862/179EVN315)

**УДК 00.004.9**

**Каширина Надежда Валентиновна**
ФГБОУ ВО «Государственный Университет Управления»
Россия, Москва
Старший преподаватель
E-mail: Nadin.vk@mail.ru

**Маран Михкель Михкелевич**
ФГБОУ ВО НИУ «МЭИ»
Россия, Москва[1]
Доцент
Кандидат технических наук
E-mail: maranmm@mpei.ru

# Сопоставительный анализ подготовки специалистов по информационным технологиям в вузах России и за рубежом

[1] 111250, Москва, Красноказарменная ул., 14

**Аннотация.** В статье рассмотрен процесс становления информационных технологий и развитие подготовки кадров для этой отрасли. Вместе с развитием информационных технологий процесс подготовки кадров тоже претерпел существенные изменения. Проведен сопоставительный анализ подготовки специалистов по информационным технологиям в России, странах Евросоюза и по рекомендациям ACM/IEEE. Выделено два уровня подготовки специалистов: подготовка профессионалов по разработке программного обеспечения (в том числе и системного) и подготовка по направлениям, по которым требуется глубокое понимание процессов обработки информации на компьютерах. Для первой группы специалистов проведено сравнение рекомендации ACM/IEEE и ФГОС России по направлению «Информатика и вычислительная техника» и «Программная инженерия». На основе проведенного анализа можно сделать вывод, что существенных отличий между требованиями к подготовке специалистов нет. Для будущих специалистов-квалифицированных пользователей предусмотрено изучение алгоритмизации и программирования, в разных учебных заведениям предпочтение отдается разным языкам программирования, но преобладает Python. В итоге можно утверждать, что существенных отличий между подготовкой специалистов в России и за рубежом нет.

**Ключевые слова:** информационная революция; информационный взрыв; этапы развития программирования; программная инженерия; кризис программирования; развитие информационных технологий; подготовка специалистов по информационным технологиям; подготовка квалифицированных пользователей; Федеральные государственные образовательные стандарты; рекомендации ACM/IEEE.

**Ссылка для цитирования этой статьи:**

Каширина Н.В., Маран М.М. Сопоставительный анализ подготовки специалистов по информационным технологиям в вузах России и за рубежом // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 7, №3 (2015) http://naukovedenie.ru/PDF/179EVN315.pdf (доступ свободный). Загл. с экрана. Яз. рус., англ. DOI: 10.15862/179EVN315

Мы живем в быстро развивающемся информационном обществе, характеризующемся новой ролью информации в обществе, скоростью ее распространения и переработки. [Воронина Т.П. Философские проблемы образования в информационном обществе Автореферат диссертации на звание доктора философских наук. М.: 1995]. В начале жизни человека (в младенческом возрасте) человеку было достаточно элементарных знаний и простейших умений и навыков. По мере развития личности требуются уже не только индивидуальные, но и групповые (коллективные) информационные технологии, способствующие адекватной реакции на обстоятельства. Для этого человеку уже в начале жизни требуются различные технические средства. Другими словами личность должна пройти этапы филогенеза. Этапы появления средств и методов обработки потоков информации, вызывающих кардинальные изменения в развитии личности (филогенезные процессы отражённые в онтогенезе), обычно определяются как *информационные революции* трансформированные в опыте конкретной персоны. Обычно принято говорит о четырёх информационных революциях. Считается, что *первая информационная революция* связана с изобретением письменности. При этом, почему появление устной речи не считается как революция в информационной сфере. *Вторая информационная революция* (середина XVI века) это эпоха Возрождения и связана с изобретением книгопечатания. *Третья информационная революция* (конец XIX века) связана с появлением телеграфа, телефона и радио. *Четвертая информационная революция* (70-е годы XX столетия) связана с изобретением микропроцессорной технологии и появлением *персональных компьютеров.*

Представления о достижениях в области компьютерной техники представлены (табл. 1.).

**Таблица 1**

**Поколения компьютеров (согласно Емельяновой М.С. http://emelmarya.ucoz.ru/index/osnovnye_ehtapy_razvitija_informacionnogo_obshhestva/0-57)**

| Поколение | Элементная база | Характеристика |
|---|---|---|
| Первое поколение (середина 40-х -конец 50-х годов) | Электронные лампы | ЭВМ отличаются огромными габаритами, большим потреблением энергии, малым быстродействием, низкой надежностью. Программирование ведется в кодах. |
| Второе поколение (конец 50-х - середина 60-х годов) | Полупроводниковые элементы, транзисторы | Улучшены все технические характеристики. Для программирования используются алгоритмические языки. |
| Третье поколение (середина 60-х -конец 70-х годов) | Интегральные схемы, многослойный печатный монтаж | Резкое снижение габаритов ЭВМ, повышение их надежности, увеличение производительности. Возможность доступа с удаленных терминалов. |
| Четвертое поколение (конец 70-х годов по настоящее время) | Микропроцессоры, большие интегральные схемы | Улучшены технические характеристики. Массовый выпуск персональных компьютеров. |
| Перспективы развития | Мощные многопроцессорные вычислительные системы с высокой производительностью; создание дешевых микрокомпьютеров; разработка интеллектуальных компьютеров. Внедрение компьютерных сетей во все сферы и их объединение; распределенная обработка данных; повсеместное использование компьютерных информационных технологий. | |

Для того, чтобы понять как работали информационные технологии раньше один пример «Какие-нибудь 40 лет назад полеты наших первых спутников рассчитывались следующим замечательным способом. В центре, проектировавшем траектории полета, располагались два огромных зала, заполненных специалистами-расчетчиками. Каждый из них выполнял только одну арифметическую операцию в алгоритме расчета. Например, на стол определенного специалиста с двух соседних столов постоянно поступали две бумажки с числами, там они суммировались, и бумажка с суммой передавалась на следующий стол. Вот так, полностью имитируя функции компьютера, недели и месяцы работали сотни специалистов. Для повышения надежности во втором зале шел точно такой же расчетный процесс, и полученные результаты сравнивались» [http://textb.net/87/67.html]. С четвертой информационной революцией связано и понятие информационного взрыва, то есть резкого увеличения объемов информации. Современный человек получает за неделю столько информации, сколько житель средневековья – за всю жизнь. В этом нам помогают плееры, мобильные телефоны и, самое главное, развитие компьютерных сетей. Возможности человека по усвоению и обработке информации ограничены. Человек может усваивать информацию со скоростью не больше 25 бит/сек. В образовании половина полученных знаний становятся бесполезными уже через семь лет, а в области информационных технологий уже через год. Соответственно, при обучении надо не преподносить данные для запоминания, а стараться показать, как находить нужную информацию и ориентироваться в большом количестве разных сведений.

С появлением интернета масса людей, не являющихся специалистами в области информационных технологий, получили возможность использовать информацию и обмениваться ею через различные социальные сети и форумы. Недостатком этого является дублирование информации и ее невысокое качество.

Информационный взрыв характеризуется также следующими показателями:

- быстрое совершенствование компьютерной техники. Широкое использование мультимедийных средств;
- интенсивный рост индустрии информации.
- информация трансформируется в цифровую, становится обрабатываемой на компьютере;
- инструменты и приборы содержат все больше электроники, приобретают свойство интеллекта, в центре всего становится компьютер;
- в любой сфере деятельности становится невозможным обойтись без использования электронных приборов - компьютеров;
- государственные органы представлены в виртуальном пространстве;
- документооборот становится электронным
- расширяется беспроводной доступ к сети;
- все большая часть продаж, услуг осуществляется через интернет.

Из вышесказанного следует, что в современном мире информационные технологии приобретают все большее значение. Все возрастающее воздействие компьютеров в современном мире предъявляет важные требования к созданию программных систем. Эти требования состоят в необходимости такой разработки и сопровождения программных систем, которые гарантируют:

- Надежность системы.

- Удобство в использовании.
- Безопасной.
- Проверяемой.
- Оставляющей главную роль за человеком, а не за компьютером.

Совокупность методов и средств, используемых в процессе разработки программного обеспечения, называется технологией программирования. Рассмотрим развитие технологии программирования в историческом контексте, выделяя основные этапы развития программирования, как науки. [8,10].

Первый этап – стихийное программирование. Продолжался от момента появления первых вычислительных машин до середины 60-ых годов 20 века. Первые программы имели простейшую структуру. (рис. 1). Они состояли из собственно программы на машинном языке и обрабатываемых ею данных. Появление ассемблеров позволило вместо двоичных или шестнадцатеричных кодов использовать символические имена. Программы стали более читаемыми.

***Рис. 1.*** *Структура программы первого этапа развития программирования*

Создание языков программирования высокого уровня (Fortran, Algol) упростило программирование вычислений. Это позволило увеличить сложность программы. Революцией было появление в языках средств, позволяющих оперировать подпрограммами, которые можно было сохранять и использовать в других программах. В результате были созданы библиотеки подпрограмм, которые вызывались из разрабатываемой программы.

В конце 60-ых годов 20 века разразился кризис программирования. Технические средства развивались очень быстро, особенно после появления интегральных схем в начале 60-ых годов и существующие методы разработки программ не могли использовать возможности технических средств. Кризис выражался в том, что большие программные проекты стали выполняться с отставанием от графика или с превышением сметных расходов, разработанный программный продукт не обладал требуемыми функциональными возможностями, производительность его была низка, качество получаемых программных продуктов не устраивало потребителей. Аналитические исследования и обзоры, выполненные в последующие годы ведущими зарубежными аналитиками, показывали не слишком обнадеживающие результаты. Так, например, результаты исследований, проведенных компанией Standish Group, проанализировавшей работу 364 американских корпораций и итоги выполнения более 23 тысяч проектов, показали, что только 16,2% проектов завершились в срок, не превысили запланированный бюджет и реализовали все требуемые функции. С опозданием были завершены 52,7% проектов, расходы на их разработку превысили запланированный бюджет, а требуемые функции не были реализованы в полном объеме. Аннулированы до завершения 31,1% проектов. Причинами столь низких показателей, по мнению разработчиков, являются следующие:

- Нечеткая и неполная формулировка требований к программным продуктам.
- Недостаточное вовлечение пользователей в работу над проектом.
- Отсутствие необходимых ресурсов.

- Неудовлетворительное планирование и плохое управление проектом.
- Частые изменения требований и спецификаций.
- Новизна и несовершенство используемой технологии.
- Недостаточная поддержка со стороны высшего руководства.
- Недостаточно высокая квалификация разработчиков, отсутствие необходимого опыта.

Объективно это было вызвано несовершенством технологии программирования. Использовался подход к разработке программ снизу – вверх, когда вначале разрабатывали простые программы, из которых затем пытались построить сложную программу. При сборке программного продукта выявлялось большое количество ошибок согласования. В конечном итоге процесс тестирования и отладки программ занимал больше 80% времени разработки. Анализ причин возникновения ошибок позволил сформулировать новый подход к программированию, который был назван структурным.

Второй этап – структурный подход к программированию (60-70-е годы XX в.). В основе этого подхода лежит декомпозиция сложных систем с целью последующей реализации в виде отдельных подпрограмм (рис. 2). Задача представляется в виде иерархии подзадач простейшей структуры. Проектирование осуществляется сверху – вниз. Эти принципы были заложены в основу процедурных языков программирования. Это PL/1, Pascal,C.

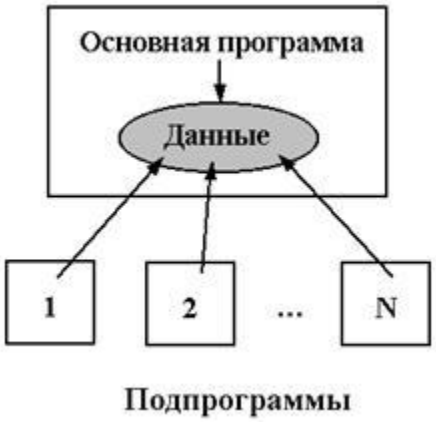


***Рис. 2.*** *Структура программы второго этапа развития программирования*

Потребность контролировать процесс разработки программного обеспечения, прогнозировать и гарантировать стоимость разработки, сроки и качество результатов привели в конце 70-ых годов к необходимости перехода от кустарных к индустриальным способам создания программ. Начала развиваться совокупность инженерных методов и средств создания программного обеспечения, объединенных общим названием «программная инженерия» (software engineering). Впервые этот термин был использован как тема конференции, проведенной под эгидой НАТО в 1968 году. В 1975 году в Вашингтоне была проведена первая международная конференция, посвященная программной инженерии. Этот период рассматривается как первый этап развития программной инженерии – систематизация и стандартизация процессов создания программного обеспечения.

Дальнейший рост сложности и размеров разрабатываемого программного обеспечения потребовал развития структурирования данных. Начала развиваться технология модульного программирования, связанная с созданием библиотек подпрограмм (рис. 3). Она предполагает выделение групп подпрограмм, использующих одни и те же глобальные переменные, в отдельно компилируемые модули. Они представляют из себя библиотеки подпрограмм. Это

новые версии языков Pascal и C (C++), языки Ада и Modula. Используя этот подход, можно получать достаточно надежные программы, размер которых не превышает 100 тысяч операторов.

При увеличении размера программы возрастает сложность межмодульных интерфейсов и трудно обеспечить взаимное влияние отдельных частей программы. Для разработки программного обеспечения большого объема было предложено использовать объектный подход.

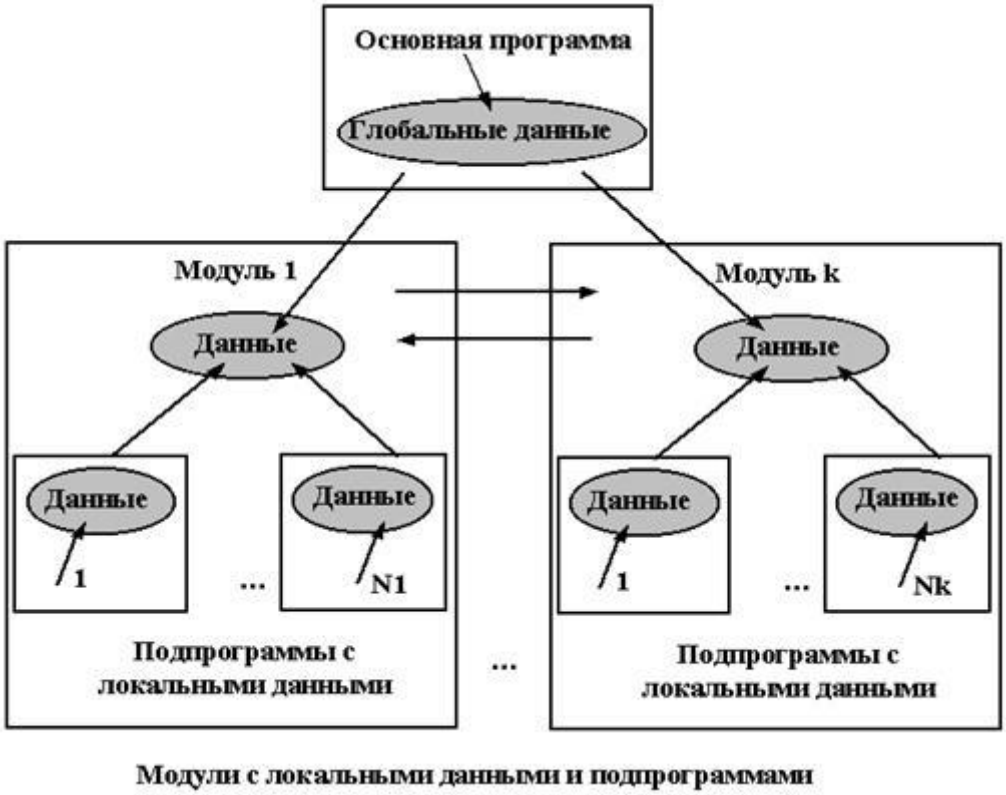


**Рис. 3.** Структура программы, реализующей технологию модульного программирования

Третий этап – объектный подход к программированию (с середины 80-х до конца 90-х годов XX века). Это второй этап развития программной инженерии – начало перехода к сборочному, индустриальному способу создания программного обеспечения на основе объектно-ориентированного подхода. Объектно-ориентированное программирование - технология создания сложного программного обеспечения, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определенного типа (класса), а классы образуют иерархию с наследованием свойств. Взаимодействие программных объектов в такой системе осуществляется путем передачи сообщений (рис. 4). Основное достоинство объектно-ориентированного программирования по сравнению с модульным программированием – более естественная декомпозиция программного обеспечения, которая существенно облегчает его разработку, а также более полная локализация данных и интегрирование их с подпрограммами обработки, что позволяет вести практически независимую разработку отдельных частей программы и создавать библиотеки классов для различных применений. Впервые объектная структура программы была использована в языке имитационного моделирования сложных систем Simula (60-е годы XX в.), в специализированном языке моделирования Smalltalk (70-е годы XX в.), а затем в новых версиях универсальных языков программирования, таких, как Pascal, C++, Modula, Java. Бурное развитие технологий программирования, основанных на объектном подходе, позволило решить многие проблемы. Так были созданы среды, поддерживающие визуальное

программирование, например, Delphi, C++ Builder, Visual C++ и т.д. При использовании визуальной среды у программиста появляется возможность проектировать некоторую часть, например, интерфейсы будущего продукта с применением визуальных средств добавления и настройки специальных библиотечных компонентов. Результатом визуального проектирования является заготовка будущей программы, в которую уже внесены соответствующие коды.

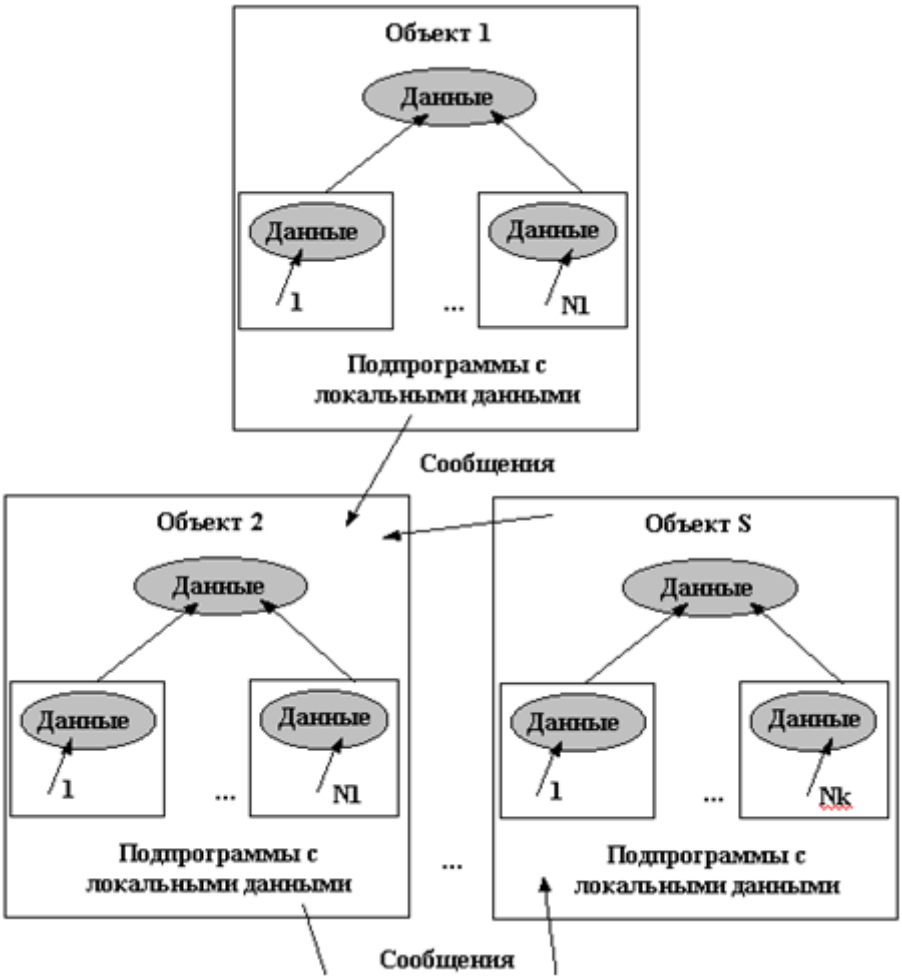


***Рис. 4.*** *Структура программы, реализующей объектно-ориентированный подход*

Использование объектного подхода имеет много преимуществ, однако его конкретная реализация в объектно-ориентированных языках программирования, таких, как Pascal и C++, имеет существенные недостатки: при использовании этих языков программирования сохраняется зависимость модулей программного обеспечения от адресов экспортируемых полей и методов, а также структур и форматов данных. Эта зависимость объективна, так как модули должны взаимодействовать между собой, обращаясь к ресурсам друг друга. Связи модулей нельзя разорвать, но можно попробовать стандартизировать их взаимодействие, на чем и основан компонентный подход к программированию.

Четвертый этап – компонентный подход и CASE-технологии (с середины 90-х годов XX до нашего времени). Компонентный подход предполагает построение программного обеспечения из отдельных компонентов физически отдельно существующих частей программного обеспечения, которые взаимодействуют между собой через стандартизованные двоичные интерфейсы. В отличие от обычных объектов объекты-компоненты можно собрать в динамически вызываемые библиотеки или исполняемые файлы, распространять в двоичном виде (без исходных текстов) и использовать в любом языке программирования,

поддерживающем соответствующую технологию. Отличительной особенностью современного этапа развития технологии программирования, кроме изменения подхода, является создание и внедрение автоматизированных технологий разработки и сопровождения программного обеспечения, которые были названы CASE-технологиями (Computer-Aided Software/System Engineering – разработка программного обеспечения / программных систем с использованием компьютерной поддержки). Без средств автоматизации разработка достаточно сложного программного обеспечения на настоящий момент становится трудно осуществимой: память человека уже не в состоянии фиксировать все детали, которые необходимо учитывать при разработке программного обеспечения. На сегодня существуют CASE-технологии, поддерживающие как структурный, так и объектный (в том числе и компонентный) подходы к программированию.

В основе программной инженерии лежит фундаментальная идея: проектирование программного обеспечения является формальным процессом, который можно изучать и совершенствовать. Однако, как показывает опыт более тридцати прошедших лет, использовать математические модели для формализации процесса проектирования программного обеспечения не столь просто, и результаты достаточно скромные. Ф. Брукс, руководитель проекта разработки операционной системы OS/360, отмечал, что самым существенным свойством программных систем является их сложность. Для успешной реализации проекта должны быть построены модели архитектуры программной системы, определяющей совокупность структурных элементов системы и связей между ними, поведение элементов системы в процессе их взаимодействия, а также иерархию подсистем, объединяющих структурные элементы.

Как уже упоминалось, в современном мире информационные технологии приобретают все большее значение, поэтому надо обеспечить должный уровень подготовки профессионалов в этой области, в частности в области разработки программного обеспечения.

Наблюдаются следующие тенденции в системе подготовки специалистов в области информационных технологий:

- Снизилась доля подготавливаемых специалистов в области фундаментальной науки.
- Появились новые направления подготовки: прикладная информатика, информационный менеджмент, бизнес-информатика, информационная безопасность, программная инженерия.

Кроме этого, изменился статус специалиста по информационным технологиям. Если раньше такие специалисты не являлись ведущими специалистами в организации, а лишь отвечали за бесперебойную работу компьютеров, то со временем, когда процесс информатизации привел к реорганизации структуры организации, изменению форм документооборота, то статус специалиста по информационным технологиям существенно повысился. [9].

Следует отметить, что количество специалистов по информационным технологиям явно недостаточно, как у нас в стране, так и за рубежом. Согласно данным Ассоциации информационных технологий США (Information Technology Association of America) в области информационных технологий работают 10,4 млн. человек и имеется еще 900 тысяч вакансий. Аналогичное положение сложилось и в России.

Согласно новому ФГОС ВО предусматривается подготовка бакалавров по двум видам деятельности: академические бакалавры, при подготовке которых рассматриваются научные аспекты программирования (Computer Science) и прикладные бакалавры, когда

программирование понимается как инженерная дисциплина (Software Engineering). Однако проблемы, возникающие при подготовке как академических, так и прикладных бакалавров, имеют много общего. Следует признать, что в России используется импортное техническое обеспечение и, в большой степени - программное, так что очень актуален вопрос подготовки отечественных профессионалов для разработки нового программного обеспечения (в том числе и системного) или для администрирования сложных программных комплексов. Эти специалисты должны обладать хорошими знаниями по математике (математическая логика, дискретная математика, алгебра, исследование операций), поэтому их подготовка осуществляется, в основном, в технических ВУЗах. В России существуют следующие направления подготовки и специальности высшего профессионального образования (таблица 2):

**Таблица 2**

**Количество ВУЗов, осуществляющих подготовку специалистов по информационным технологиям (сделана авторами)**

| код направления | название | квалификация | количество ВУЗов, осуществляющих подготовку | из них в Москве и Петербурге |
|---|---|---|---|---|
| 010200 | Математические и компьютерные науки | магистр<br>бакалавр | 8<br>18 | 1<br>4 |
| 010300 | Фундаментальные информатика и информационные технологии | магистр<br>бакалавр | 8<br>46 | 4<br>21 |
| 010400 | Прикладная математика и информатика | магистр<br>бакалавр | 60<br>90 | 20<br>38 |
| 010500 | Математическое обеспечение и администрирование информационных систем | магистр<br>бакалавр | 1<br>55 | 1<br>9 |
| 090301 | Компьютерная безопасность | специалист | 28 | 7 |
| 090302 | Информационная безопасность телекоммуникационных систем | специалист | 19 | 5 |
| 090303 | Информационная безопасность автоматизированных систем | специалист | 41 | 6 |
| 090305 | Информационно-аналитические системы безопасности | специалист | 4 | 2 |
| 090900 | Информационная безопасность | Бакалавр<br>магистр | 13<br>109 | 7<br>30 |

| код направления | название | квалификация | количество ВУЗов, осуществляющих подготовку | из них в Москве и Петербурге |
|---|---|---|---|---|
| 090915 | Безопасность информационных технологий в правоохранительной сфере | специалист | 13 | 6 |
| 230100 | Информатика и вычислительная техника | магистр<br>бакалавр | 116<br>366 | 36<br>49 |
| 230400 | Информационные системы и технологии | магистр<br>бакалавр | 75<br>230 | 20<br>44 |
| 230700 | Прикладная информатика | магистр<br>бакалавр | 92<br>491 | 27<br>72 |
| 231000 | Программная инженерия | магистр<br>бакалавр | 33<br>86 | 9<br>18 |
| 231300 | Прикладная математика | магистр<br>бакалавр | 10<br>15 | 2<br>23 |

Лидерами по подготовке специалистов по информационным технологиям являются МФТИ, ВМК МГУ, МИФИ, МГТУ имени Баумана. По информационным технологиям нужно много разных специалистов: системные администраторы, администраторы баз данных, специалисты по интерфейсам, архитекторы информационных систем, системные аналитики. По мере развития информационных технологий изменяется круг требуемых специалистов, но основным умением для них остается владение программированием.

Какими же качествами должен обладать человек, чтобы стать хорошим программистом? [http://*forum.vingrad.ru/forum/topic-18296/60.html]* Психологи еще не нашли ответ на вопрос: человек рождается со способностями к программированию или они формируются позднее? Среди множества тестов нет таких, которые надежно устанавливали способности к программированию. Мало того, среди успешных разработчиков есть люди с разными типами личности. Однако, можно отметить следующее: написанию программы предшествует разработка алгоритма, то есть по существу программа – это алгоритм, оформленный согласно требованиям применяемого языка программирования. Нельзя составить хороший алгоритм без способности к логическому мышлению и абстрагированию, то есть без умения выделять взаимосвязи, находить зависимости и делать выводы. Итак, первое требование – способность к логическому мышлению. Следующее требование, которое показывает способность к этой работе - готовность программиста обновлять знания и видоизменять навыки, не пытаясь зафиксироваться на уже достигнутом и уже приобретенном опыте. Это объясняется тем, что для информационных технологий характерно очень быстрое обновление знаний, что актуально как для специалистов-исследователей, так и для специалистов-разработчиков. Им приходится учиться всю жизнь. Этим объясняется большое количество тренингов, курсов и программ переподготовки для программистов. Еще программисту требуются:

- Математические способности.
- Интуитивное мышление.
- Аккуратность.
- Усидчивость.

- Внимательность.
- Самодостаточность.
- Способность к созданию образа по словесному описанию.
- Умственная работоспособность.
- Способность заниматься длительное время кропотливой работой.
- Целеустремленность.
- Умение работать в команде.
- Исполнительность.
- Память на условные обозначения, числа и символы.

Подготовка профессионалов в области разработки и эксплуатации программного обеспечения включает прохождение целого цикла дисциплин о поддержке всех этапов жизненного цикла и теоретическим основам разработки. Согласно посвященному этому вопросу отчету совместного комитета ACM (Association for Computing Machinery) и IEEE Computer Society [4] подготовка бакалавров по направлению программная инженерия должно включать прохождение следующих дисциплин:

1. **Основы программирования.** Дисциплина не требует предварительной подготовки и должна ознакомить студентов с базовыми алгоритмами, принципами структурного программирования, основными конструкциями языка программирования, представлением данных в памяти компьютера, а также основами методики разработки и тестирования программ. В качестве языка программирования повсеместно используется C++, ограничиваясь средствами процедурного программирования. Выбор языка объясняется его почти безраздельным распространением для профессионального программирования, а также тем, что на базе C++ разработаны более новые языки (C#, Java) и освоившему C++ не составляет никаких трудностей их изучение.

2. **Объектно-ориентированное программирование (ООП)**. Дисциплина базируется на предыдущей и знакомит студентов с парадигмой ООП, принципами разработки программ по объектно-ориентированной методике. Важно понимать, что ООП не отрицает структурное программирование, а является его развитием. используемыми языками являются C++, в меньшей степени C#, Java.

3. **Алгоритмы и структуры данных**. Дисциплина включает изучение абстрактных структур данных (стек, очередь, список, дерево, сеть), алгоритмов их создания, изменения и обработки. В зависимости от объема дисциплины можно подробно рассматривать названные алгоритмы или ограничиться рассмотрением контейнерных классов C++ и стандартных классов для работы с динамическими структурами данных на C#.

4. **Архитектура ЭВМ**. Дисциплина познакомит студентов с устройством и принципами работы компьютеров.

5. **Операционные системы и сети**. Студенты узнают принципы функционирования операционных систем, коротко рассмотрят алгоритмы распределения ресурсов компьютера и управления вычислительным процессом. В разделе сетевых технологий будут изложены типовые структуры сетей и их функционирование.

6. **Базы данных**. Дисциплина посвящена изучению проблем хранения обработки больших массивов данных. Рассматриваются сбор исходных данных для проектирования баз данных. Основы теории и проектирование структуры реляционной модели данных.

Реализация базы данных на одной системе управления базами данных. Язык запросов SQL. Особенности распределенных и объектно-ориентированных баз данных.

7. **Дискретная математика.** Цель этого курса – создание теоретического фундамента. Круг рассматриваемых вопросов включает основы математической логики (алгебра высказываний, логика предикатов, теория логического вывода), элементы теории графов и комбинаторики (с рассмотрением основных алгоритмов задач на графах и комбинаторики), понятия вычислимости и вычислительной сложности алгоритмов.

8. **Статистические методы вычислений.** Основные методы статистического анализа данных, проверка статистических гипотез, применение статистики для анализа производительности. Работа со статистическими пакетами и/или статистическими функциями электронных таблиц.

9. **Программная инженерия.** После прохождения этой дисциплины студенты должны уметь провести анализ, формулировать требования к разработке или усовершенствованию программных средств. Составить UML-диаграммы и проектировать архитектуру программных средств. Выполнять их реализацию с разработкой интерфейсов пользователей и тестирование. Знать методы управления программными проектами и обеспечения их качества. Предусмотрены следующие разделы этой дисциплины:

- Конструирование программного обеспечения.
- Разработка человеко-машинных интерфейсов.
- Архитектура больших систем программного обеспечения.
- Тестирование программного обеспечения.
- Детальное проектирование программного обеспечения.
- Формальные методы программной инженерии.
- Обеспечение качества программного обеспечения.
- Анализ требований к программному обеспечению.
- Управление программными проектами.

10. **Экономика разработки программных средств.** Студенты должны научиться выполнять анализ соотношения затрат и результатов, анализировать экономический эффект альтернативных решений.

11. **Методы групповой коммуникации.** Студенты должны научиться грамотно написать техническую документацию программных средств, рецензировать документацию, создать качественные презентации программных продуктов и участвовать в общении с другими участниками разработки на лаконичном и точном профессиональном языке.

Для сравнения рассмотрим требования, предъявляемые будущим бакалаврам по направлению 231000 «Программная инженерия» действующим на сегодняшний день ФГОС ВПО. Ограничимся рассмотрением математического и естественнонаучного и профессионального циклов.

В естественнонаучном цикле предусмотрено изучение следующих дисциплин:

1. Математический анализ.
2. Алгебра и геометрия.
3. Математическая логика и теория алгоритмов.

4. Дискретная математика.
5. Теория автоматов и формальных языков.
6. Теория вероятностей и математическая статистика.

В базовой части профессионального цикла предусмотрено изучение следующих дисциплин:

1. Информатика и программирование.
2. Алгоритмы и структуры данных.
3. Введение в программную инженерию.
4. Архитектура вычислительных систем.
5. Операционные системы и сети.
6. Базы данных.
7. Конструирование программного обеспечения.
8. Проектирование человеко-машинного интерфейса.
9. Проектирование и архитектура программных систем.
10. Тестирование программного обеспечения.
11. Разработка и анализ требований.
12. Управление программными проектами.
13. Экономика программной инженерии.
14. Безопасность жизнедеятельности.

При сравнении предлагаемых для изучения дисциплин можно сделать вывод, что они практически совпадают. В образовательном стандарте Российской Федерации больше внимания уделено фундаментальной подготовке (математический анализ, аналитическая геометрия). Но отсутствует (по меньшей мере, в обязательной части) изучение технического языка.

Для наглядности изложенный материал приведен в таблице 3.

**Таблица 3**

**Сравнение предлагаемых для изучения дисциплин (сделана авторами)**

| | **Рекомендации ACM/ IEEE** | **Требования ФГОС РФ** |
|---|---|---|
| 1 | Основы программирования | Информатика и программирование |
| 2 | Объектно-ориентированное программирование | |
| 3 | Алгоритмы и структуры данных | Алгоритмы и структуры данных |
| 4 | Архитектура ЭВМ | Архитектура вычислительных систем |
| 5 | Операционные системы и сети | Операционные системы и сети |
| 6 | Базы данных | Базы данных |
| 7 | Дискретная математика | Дискретная математика |
| 8 | | Математическая логика и теория алгоритмов |
| 9 | | Теория автоматов и формальных языков |
| 10 | Статистические методы вычислений | Теория вероятностей и математическая статистика |

| | Рекомендации ACM/ IEEE | Требования ФГОС РФ |
|---|---|---|
| 11 | Программная инженерия | Введение в программную инженерию |
| 12 | | Конструирование программного обеспечения |
| 13 | | Проектирование человеко-машинного интерфейса |
| 14 | | Проектирование и архитектура программных систем |
| 15 | | Тестирование программного обеспечения |
| 16 | | Разработка и анализ требований |
| 17 | | Управление программными проектами |
| 18 | Экономика разработки программных средств | Экономика программной инженерии |
| 19 | Методы групповой коммуникации | |
| 20 | | |
| 21 | | Математический анализ |
| 22 | | Алгебра и геометрия |
| 23 | | Безопасность жизнедеятельности |

Можно провести аналогичный сопоставительный анализ между подготовкой бакалавров по рекомендациям ACM - IEEE Computer Society по направлению «Информатика» и отечественным образовательным стандартом направления 230100 «Информатика и вычислительная техника». Наиболее существенным отличием является наличие в ФГОС среди обязательных дисциплины «Электротехника, электроника и схемотехника», отсутствующей в их рекомендациях и наличие в рекомендациях ACM - IEEE Computer Society раздела «Социальные и профессиональные вопросы», в рамках которого предусмотрено рассмотрение следующих вопросов: профессиональная и этическая ответственность, интеллектуальная собственность, социальный контекст информатики.

В целом можем констатировать, что основные требования к подготовке бакалавров в области информатики и вычислительной техники и программной инженерии, установленные ФГОС ВПО в Российской Федерации и рекомендациями ACM IEEE Computer Society практически совпадают.

Рассмотрим в качестве конкретного примера сравнение учебных планов подготовки бакалавров по направлению 010300 «Фундаментальная информатика и информационные технологии» Пермского государственного национального исследовательского университета (ПГНИУ) и программы «Computer Science» Университета штата Северная Каролина (NCSU) [7]. Ограничимся сравнением циклов математического и естественнонаучного дисциплин и профессиональных дисциплин, так как именно эти циклы в основном определяют уровень компетентности будущего бакалавра. В таблице 4 дисциплины разделены на группы.

**Таблица 4**

**Сравнение учебных планов подготовки бакалавров (*сделана авторами)***

| Дисциплина | ПГНИУ | NCSU |
|---|---|---|
| Химия | - | + |
| Физика | - | + |
| | | |
| Математический анализ | + | + |
| *Кратные интегралы и ряды* | + | - |
| *Дифференциальные уравнения* | + | - |
| | | |
| Линейная алгебра | - | + |
| *Алгебра и геометрия* | + | - |
| | | |
| Теория вероятностей и математическая статистика | + | + |
| | | |
| Введение в вычислительные среды | - | + |
| Введение в компьютинг (Java) | - | + |
| Концепции программирования | + | - |
| *Основы программирования* | + | - |
| *Практикум по программированию* | + | - |
| | | |
| Дискретная математика | + | + |
| Теория автоматов | - | + |
| *Математическая логика и теория алгоритмов* | + | - |
| *Теория автоматов и формальных языков* | + | - |
| *Неклассические логики* | + | - |
| *Теория конечных графов и ее приложения* | + | - |
| | | |
| Программирование на С | - | + |
| *Языки программирования* | + | - |
| Структуры данных и алгоритмы | - | + |
| | | |
| Организация компьютера и язык ассемблера | - | + |
| *Архитектура вычислительных систем* | + | - |
| Операционные системы | + | + |
| | | |
| Программная инженерия | + | + |
| *Моделирование информационных процессов* | + | - |
| *Основы проектирования и реализации информационных систем* | + | - |
| *Управление проектами* | + | - |
| | | |
| Этика в компьютинге | - | + |
| Техническое письмо | - | + |
| | | |
| Компьютерная графика | + | - |
| Методы оптимизации и исследование операций | + | - |
| Администрирование информационных систем | + | - |

Из фундаментальных дисциплин в учебном плане ПГНИУ отсутствуют физика и химия, а также этика и техническое письмо. В учебном плане NCSU отсутствуют компьютерная графика, методы оптимизации и администрирование информационных систем. Кстати, и в приведенных выше рекомендациях ACM тоже содержится технический язык. Надо, наверно, согласиться, что обучение студентов грамотно писать техническую документацию на свои разработки совсем не лишнее. Изучение физики и особенно химии для специалистов по информационным технологиям кажется лишним. Дисциплины компьютерная графика и администрирование информационных систем, безусловно, полезны.

Что касается других групп дисциплин, то они практически совпадают и для определения их отличии следовало бы ознакомиться не только с учебными планами, но уже с программами дисциплин.

В образовательные стандарты подготовки специалистов, в профессиональной деятельности которых необходимо глубокое понимание принципов разработки и использования программного обеспечения и умение разрабатывать программы для решения прикладных задач входит дисциплина «Информатика» и в учебные программы включено изучение программирования. [5,6]. Следует обратить внимание, что первостепенное значение при этом имеет не изучение элементов того или иного языка программирования, а освоение базовых алгоритмов и методики решения задач на ЭВМ, начиная с постановки задачи и заканчивая тестированием. Поэтому выбор языка имеет второстепенное значение. Но язык программирования, используемый на начальном этапе обучения, должен обеспечить написание работающих программ, свободных от лишних технических деталей. В последние годы таким языком стал *Python*. В качестве основных его преимуществ можно назвать:

- Динамическое определение типов данных и как результат: нет необходимости в объявлениях.
- Наличие достаточно развитых структур данных (список, очередь, словарь) с операциями работы с ними, что освобождает программиста от их написания.
- Традиционный, но достаточно простой синтаксис базовых операторов.
- Наличие свободно-распространяемых реализаций.

В то же время *Python* поддерживает структурное, объектно-ориентированное и функциональное программирование. Правда, последние две разновидности – с ограничениями. Имеются большой набор функций, средства работы с базами данных, с интернетом и с данными в XML-формате.

Называть среди преимуществ *Python* наличие упомянутых выше сложных структур данных по сравнению с C++ весьма спорно. Контейнерные классы C++ и стандартные классы C# обладают значительно более широкими возможностями при простоте использования. Средства функционального программирования C# через язык LINQ также обладают более широкими возможностями, чем аналогичные средства *Python*.

Поэтому можно сделать вывод, что использование *Python* в качестве языка для начальной программистской подготовки оправдано для направлений подготовки прикладных программистов. Этот язык успешно применяется у нас и в зарубежных вузах.

## ЛИТЕРАТУРА

1. Воронина Т.П. Философские проблемы образования в информационном обществе Автореферат диссертации на звание доктора философских наук. М.: 1995.
2. http://emelmarya.ucoz.ru/index/osnovnye_ehtapy_razvitija_informacionnogo_obshhestva/0-57.
3. http://textb.net/87/67.html.
4. Computer Science Curricula 2013 Curriculum Guidelines for Under-graduate Degree Programs in Computer Science December 20, 2013. The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society. На сайте http://www.acm.org/education/CS2013-final-report.pdf.
5. Аннотация дисциплины «Информатика» на экономическом факультете Таллиннского технического университета. На сайте http://www.ttu.ee/.
6. Аннотация дисциплины «Информатика» на экономическом факультете Тартуского университета. На сайте http://www.ut.ee/.
7. Хеннер Е., Стэллманн М. Подготовка специалистов по ИТ: Россия и США. Открытые системы, 2013, №3, с. 23.
8. Назаров С.В. Архитектура и проектирование программных систем: Монография. – М.: ИНФРА-М, 2013.-351 с.
9. Иванова Е.В. Подготовка специалиста в области информационных технологий на уровне CIO. Задачи высшей школы и бизнеса // Успехи современного естествознания. – 2008 – №4 – С. 77-79. На сайте: http://www.rae.ru/use/?section=content&op=show_article&article_id=7782858.
10. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения: учебное пособие / под ред. Л.Г. Гагариной. - М.: ИД «ФОРУМ»: ИНФРА – М., 2008-400 с.

**Рецензент:** Кирсанов Константин Александрович, главный редактор интернет-журнала «Науковедение», доктор экономических наук, профессор.

**Kashirina Nadezhda Valentinovna**
State University of Management
Russia, Moscow
E-mail: Nadin.vk@mail.ru

**Maran Mihkel Mihkelevich**
Nacional Research University MPEI
Russia, Moscow
E-mail: maranmm@mpei.ru

# Comparative analysis of training for the information technology specialists in the higher education institutes of Russia and abroad

**Abstract**. The article describes the process of information technology formation and development of personnel training for this industry. Along with the development of information technology the process of personnel training also undergone significant changes. The comparative analysis of the specialists training for information technology in Russia, the European Union countries and on the recommendations of ACM/IEEE was carried out. Two levels of specialists training were selected: training of professionals for software development (including system one) and training for areas that require a deep understanding of computer information processing. Comparison of recommendations of ACM/IEEE and FSES (Federal State Education Standards) of Russia was carried out for the first group of experts in such areas as "Information and Computer Science" and "Software Engineering". Based on this analysis, we can conclude that the significant differences between the requirements for training of specialists are absent. The study of algorithmization and programming is provided for the future specialists - qualified users. In the different educational institutions the various programming languages have been preferred, but Python dominates. As a result, one may state that the substantial differences between the training of specialists in Russia and abroad are absent.

**Keywords:** information revolution; information blow-up; stages of programming development; software engineering; computer programming crisis; development of information technology; training of information technology specialists; training of skilled users; Federal State Educational Standards.

## REFERENCES

1. Voronina T.P. Filosofskie problemy obrazovaniya v informatsionnom obshchestve Avtoreferat dissertatsii na zvanie doktora filosofskikh nauk. M.: 1995.
2. http://emelmarya.ucoz.ru/index/osnovnye_ehtapy_razvitija_informacionnogo_obshhestva/0-57.
3. http://textb.net/87/67.html.
4. Computer Science Curricula 2013 Curriculum Guidelines for Under-graduate Degree Programs in Computer Science December 20, 2013. The Joint Task Force on Computing Curricula Association for Computing Machinery (ACM) IEEE Computer Society. Na sayte http://www.acm.org/education/CS2013-final-report.pdf.
5. Annotatsiya distsipliny «Informatika» na ekonomicheskom fakul'tete Tallinnskogo tekhnicheskogo universiteta. Na sayte http://www.ttu.ee/.
6. Annotatsiya distsipliny «Informatika» na ekonomicheskom fakul'tete Tartuskogo universiteta. Na sayte http://www.ut.ee/.
7. Khenner E., Stellmann M. Podgotovka spetsialistov po IT: Rossiya i SShA. Otkrytye sistemy, 2013, №3, s. 23.
8. Nazarov S.V. Arkhitektura i proektirovanie programmnykh sistem: Monografiya. – M.: INFRA-M, 2013.-351 s.
9. Ivanova E.V. Podgotovka spetsialista v oblasti informatsionnykh tekhnologiy na urovne CIO. Zadachi vysshey shkoly i biznesa // Uspekhi sovremennogo estestvoznaniya. – 2008 – №4 – S. 77-79. Na sayte: http://www.rae.ru/use/?section=content&op=show_article&article_id=7782858.
10. Gagarina L.G., Kokoreva E.V., Visnadul B.D. Tekhnologiya razrabotki programmnogo obespecheniya: uchebnoe posobie / pod red. L.G. Gagarinoy. - M.: ID «FORUM»: INFRA – M., 2008-400 s.