

Вестник Череповецкого государственного университета. 2024. № 3 (120). С. 213–230.
Cherepovets State University Bulletin, 2024, no. 3 (120), pp. 213–230.

Научная статья

УДК 373.8

<https://doi.org/10.23859/1994-0637-2024-3-120-18>

EDN: DOBFDC

Необходимые компетенции выпускника школы в области информационных технологий в условиях острого дефицита IT-специалистов

Олег Александрович Козлов¹, Ирина Викторовна Барышева^{2✉}

ФГБНУ «Институт стратегии развития образования»,
Москва, Россия

¹ole-kozlov@yandex.ru, <https://orcid.org/0000-0002-4148-2726>

^{2✉}ibar1950@yandex.ru

Аннотация. В работе рассматривается возможный вариант школьного образования в области информационных технологий и программирования с точки зрения оценки ситуации преподавателями высшей школы, ведущих подготовку специалистов в области информационных технологий. Приводятся результаты опросов выпускников, поступивших на специальности «Прикладная математика и информатика», «Фундаментальная информатика и информационные технологии» и «Программная инженерия». Целью исследования является определение реальных базовых знаний в области программирования, с которыми пришли новые студенты. Анализируются причины падения уровня знания программирования среди школьников, а также изменения восприятия динамики развития науки. Исследуется история сверхбыстрого и хаотичного развития школьного предмета от «Основы вычислительной техники и КТ», появившегося в школьной программе в 1985 г., до «Информатика и КТ» в современных реалиях, которая состоит в том, что все новое и привлекательное, появившееся в области информационных технологий, пытались в том или ином виде включить в школьную программу. Особенно быстрые и заметные (не только для профессиональной среды) изменения в IT-индустрии наблюдались с начала 2000-х гг. с появлением офисных приложений, интернета, удобного интерфейса, с развитием технологической базы, расширением доступности разных вариантов компьютеров – от телефонов, ноутбуков до серьезных высокопроизводительных компьютеров в личной собственности. Использование различных приложений – от игр и чатов общения до готовых рефератов и домашних заданий – превратили детей в самых активных пользователей достижений в области информационных технологий. Разработчики школьной программы по информатике в попытках не отстать от современных достижений в сфере IT включали и включали, как заплатки, все новые и новые разделы, без заметного увеличения часов, без должной подготовки учителей, без разработки методики преподавания.

В качестве вывода проведенного исследования предлагается концептуально пересмотреть всю программу школьного предмета, определить основные цели и приоритеты для каждого возраста, очертить круг проблем и вопросов, решение которых обеспечивает достижение поставленных задач. Особенно важным результатом обсуждений должно быть выделение

программирования в отдельный предмет. Кардинальным изменениям необходимо подвергнуть содержательную часть выпускного экзамена по информатике: первое – определиться, это экзамен по информатике или по программированию, второе – если все-таки это экзамен по программированию, чем он де-факто стал в последние два года, тогда и число задач необходимо привести в соответствие, и содержание должно иметь другой вид, и система оценок должна быть не двузначной, но учитывать степень выполнения задачи. В работе обсуждается выбор первого языка программирования, предлагается методика обучения программированию, сделана попытка определения возраста, с которого целесообразно изучение программирования, определяется круг вопросов, которые могут быть включены в школьную программу по программированию.

Ключевые слова: информатика в школе, методика обучения программированию, Python в школе, первый язык программирования

Для цитирования: Козлов О. А., Барышева И. В. Необходимые компетенции выпускника школы в области информационных технологий в условиях острого дефицита IT-специалистов // Вестник Череповецкого государственного университета. 2024. № 3 (120). С. 213–230. <https://doi.org/10.23859/1994-0637-2024-3-120-18>; EDN: DOBFDC

School graduate's required competences in the field of information technologies in the context of an acute shortage of IT specialists

Oleg A. Kozlov^{1✉}, Irina V. Barysheva²

^{1,2} The Federal State Budget Scientific Institution
“Institute For Strategy of Education Development”,
Moscow, Russia

¹ole-kozlov@yandex.ru, <https://orcid.org/0000-0002-4148-2726>

^{2✉}ibar1950@yandex.ru

Abstract. The paper considers a possible variant of school education in the field of information technologies and programming from the point of view of the situation assessment by higher school teachers who train specialists in the field of IT. The authors present the findings of the surveys distributed to the graduates who applied to the programmes “Applied Mathematics and Informatics”, “Fundamental Informatics and Information Technologies” and “Software Engineering”. The aim of the research is to determine the actual basic knowledge in the field of programming among first-year students. The authors analyse the reasons for the decline in the level of programming knowledge among schoolchildren, as well as changes in the dynamics of programming science development. The authors study super-fast and chaotic development of the school subject from “Basics of Computer Science and CT”, which appeared in the school curriculum in 1985, to “Informatics and CT” in modern realities, which consists in the fact that everything new and attractive that appeared in the field of IT was in this or that form included in the school curriculum. The changes in the IT industry have been particularly rapid and noticeable to everyone since the early two thousand caused by the use of office applications, the Internet, user-friendly interface, technological base development, and availability of different computers – from phones, laptops to serious high-performance personal computers. Instructional designers of school informatics curriculum, in an attempt to keep up, included more new sections without extra learning hours, without proper training of teachers, without development of teaching methods.

As a conclusion, the authors propose to conceptually review the subject curriculum, to define the main objectives and priorities for each age, to outline the range of problems and issues, the solution of which ensures the achievement of the objectives. A particularly important outcome of the

discussions is to identify programming as a separate subject. Fundamental changes should be made in the content of the final exam in informatics: firstly, it is necessary to decide whether it is an exam in informatics or in programming; secondly, if it is still an exam in programming, as it has become *de facto* in the last two years, then the number of tasks should be brought into line, the content should have a different form, and the grading system should take into account the degree of task fulfilment. The paper discusses the choice of the first programming language, proposes a methodology for teaching programming, attempts to determine the age at which it is advisable to study programming, and defines the range of issues that can be included in the school curriculum on programming.

Keywords: computer science at school, programming teaching methodology, Python at school, first programming language

For citation: Kozlov O. A., Barysheva I. V. School graduate's required competences in the field of information technologies in the context of an acute shortage of IT specialists. *Cherepovets State University Bulletin*, 2024, no. 3 (120), pp. 213–230. (In Russ.) <https://doi.org/10.23859/1994-0637-2024-3-120-18>; EDN: DOBFDC

Введение

История изучения информатики в школе насчитывает уже почти сорок лет, начиная с 1985 года (когда даже слово *информатика* не существовало) при введении предмета «Основы вычислительной техники и КТ», до нынешнего времени со спорным названием «Информатика и КТ». За эти почти сорок лет школьная программа, как и все, что связано с компьютерами и математическим обеспечением, изменялась практически с каждым годом. Первоначально это было программирование, просто ничего другого не было, такая ситуация продолжалась практически до начала 2000-х гг. Менялись языки программирования, начиная от простого варианта Basic, расшифровка букв которого в английском варианте означает примерно «для начинающих и мало знающих». Затем и Basic стал усложняться до самого Visual Basic. В конце 1990-х гг. в нашей стране появился Pascal, редко Fortran. Все это так или иначе отражалось в школьной программе. Сложности создания единой школьной программы усиливались разнообразием технической базы, в разных школах стояли разные компьютеры, не очень надежные, с различным математическим обеспечением. Достаточно часто обучение сводилось лишь к знакомству с компьютером или даже экскурсиям в тот или иной вычислительный центр. Но самое главное заключалось в том, что не была сформулирована концепция для создания задач, которые можно отнести к программированию. Например, на экзамене могла быть предложена задача нахождения объема шара с заданным радиусом, которая относится к геометрии или, если в качестве шара рассматривается планета, то – к астрономии или физике. Достаточно часто задачи по комбинаторике, которая является частью математики, считаются задачами сугубо программистскими. В этом плане интересными являются задачи вступительных экзаменов по информатике в Нижегородском государственном университете¹ в период с 1995 по 2006 гг. Каждая задача имела легенду или со-

¹ Барышева И. В., Гергель В. П., Городецкий С. Ю. и др. Информатика для абитуриентов. Задачи и решения. Нижний Новгород: Нижегородский госуниверситет, 2007, 360 с.; Барышева И. В. Подготовка абитуриентов математических факультетов к ЕГЭ по информатике. Кон-

держательную часть, для решения которой необходимо создать модель, именовать объекты, встречающиеся в задаче, составить словесное описание алгоритма и только потом написать программу, являющуюся композицией базовых алгоритмов, изучаемых во время подготовки к экзаменам. В зависимости от аудитории, которой предлагалась та или иная задача (а экзамены были на различных факультетах – от юридического до ВМК), сложность композиции могла быть разной. В набор из десяти базовых алгоритмов разной сложности входили: суммирование, поиск экстремума, отбор элементов по признаку, линейный поиск, кратность с вариантами, одна из сортировок, и два алгоритма работы со строками – деление строки с одним разделителем на слова и случай нескольких разделителей. Этот набор позднее в полном объеме, кроме последнего алгоритма, был реализован в языке Python. Для усложнения задач для абитуриентов факультета ВМК использовался еще алгоритм форматирования и работа с текстом, который определялся как массив строк. Следует заметить, что и система оценок не была двузначной: получил правильный ответ – 1 балл, ответ неверный – 0 баллов вне зависимости от степени решения задачи. Учитывался каждый элемент решения, каждый этап имел свою ценность, выраженную в баллах, система экзаменационных оценок за четыре задания формировалась из суммарного балла и была семизначной от 1, 2, 3, 3.5, 4, 4.5, 5. Время экзамена составляло три часа для четырех содержательных задач. Это совершенно разумное время. Увеличение количества задач ведет к их упрощению, стандартизации, потере содержательности, к сведению методики решения к схеме «запомни и повтори», как например, задания № 2, 5, 6, 8, 16, 17 в ЕГЭ¹ предлагается решать с помощью заученных программ с небольшими вариациями.

С появлением персональных компьютеров в конце 1990-х гг., имеющих операционную систему, например Windows с офисными приложениями, чуть позднее хорошего и доступного интернета, с развитием технической базы школьная программа пыталась догнать снежный ком всего того, что на данном этапе входит в понятие «информатика». Практически без увеличения количества часов в программу предмета включались все новые и новые разделы, которые по содержанию все более мельчали, становились номинальными, особенно пострадало программирование. Наибольшее число часов на программирование уделили в 7–9 классах, на три года всего 34 учебных часа. В этом возрасте ученики еще не достигли уровня общего развития и понимания задач, относящихся к программированию, на изучение любого предмета такого количества часов недостаточно, а с программированием, где само понятие «решить задачу» имеет совершенно иной, чем в других предметах – алгебре, геометрии, физике, биологии, химии – смысл, даже обзорно познакомиться не получится. В рамках школьных предметов практически решение задачи есть выполнение учеником некоторых действий, результат которых является решением. В про-

трольные работы. Сайт электронного обучения ННГУ. URL: <http://www.unn.ru/t-ltaning/> (дата обращения: 16.06.2023).

¹ Барышева И. В. Подготовка абитуриентов математических факультетов к ЕГЭ по информатике. Контрольные работы. Сайт электронного обучения ННГУ. URL: <http://www.unn.ru/t-ltaning/> (дата обращения: 16.06.2023).

граммировании результатом является описание действий, которые должен выполнить компьютер. Отсюда вытекает критерий принадлежности задач к программированию: если время составления описания для компьютера меньше времени выполнения самой задачи человеком, тогда имеет смысл считать данную задачу задачей по программированию. Попытка упрощения задач, введение школьных игрушечных языков типа Кумира делает предмет неинтересным для учащихся, слишком заметна игрушечность и неестественность ситуации. Обострило положение программирования в школе и развитие самого программирования, появление объектно-ориентированной идеологии, реализованной в разных языковых средах, желание сохранить накопленный багаж алгоритмов в виде готовых библиотек, а затем и включение этих библиотек в основу языка, как например, в Python. Представляется желанной замена изучения основополагающих элементов программирования на использование готовых конструкций, как кубиков, без понимания схемы их устройства.

Основная часть

1. Анализ знаний по программированию выпускников школ, поступивших в профильный вуз по информационным технологиям

В любое высшее учебное заведение приходят выпускники из разных школ, из разных городов и населенных пунктов с разными личностными характеристиками, с разным уровнем знаний, даже если они имеют примерно одинаковый суммарный балл ЕГЭ. Разброс наиболее ощутим в таком сложном для организации предмете, как программирование, для всех направлений, связанных IT-индустрией, поскольку именно программирование – наиболее сложная составляющая IT-образования. Для организации занятий, более направленных на конкретный уровень конкретной аудитории, уже много лет проводится опрос вновь поступивших студентов. При возникновении опроса актуальной проблемой было наличие домашнего компьютера, затем, когда практически у всех уже был тот или иной вариант, появился вопрос о наличии ноутбука, который позволял аккуратней создавать архив решенных задач, сглаживая разницу между матобеспечением и другими характеристиками домашнего компьютера и компьютера в учебном терминал-классе, в котором проводятся занятия по программированию. В период пандемии этот вопрос отпал, ноутбук был у всех, выбравших IT-профессию. Но все большую значимость приобретал вопрос «Что знаете в области программирования?» Первоначально ответы варьировались между “Basic” – “Pascal” и «не знаю» – «знаю немного», очень редко было что-то другое (см. табл. 1). Флуктуация данных, полученных при опросах, наблюдается в 2021–2023 пандемийных годах, когда дети мало зависели от школы, занимались, чем хотели и что смогли организовать.

Таблица 1

Знакомство с различными языками программирования студентов первого курса до начала обучения в ВУЗе

		2017 – 2020 г.		2021 г.		2022 г.		2023 г.	
Количество респондентов		Всего 87 чел.	Всего 100 %	Всего 115 чел.	Всего 100%	Всего 57 чел.	Всего 100%	Всего 30 чел.	Всего 100%
Языки программирования	Basic	40	46	–	–	–	0	–	–
	C#	–	–	7	7	3	5	–	–
	C++	5	6	22	19	14	18	8	27
	CSS	–	–	3	3	1	5	–	–
	HTML	–	–	7	7	3	5	–	–
	Java	–	–	6	6	2	4	–	–
	PHP	–	–	1	1	–	–	–	–
	Python	–	–	55	56	46	81	28	93
	SQL	–	–	1	1	–	–	–	–
	Pascal	40	46	34	34	19	33	10	33
	Другое	2	2	4	4	4	7	–	–

2023 год преподнес много неожиданностей. Стало очевидным, что ЕГЭ не только является подведением итогов многолетней учебы школьника, но и по сути управляет содержимым программы школьного предмета. Особенно данный факт сказался на программировании. Изменение формы сдачи ЕГЭ переносом на компьютеры в пандемийные годы привело к некоторому изменению тематики вопросов, но не их количества. Далее оказалось, что решение многих вопросов упрощается с помощью компьютера: сложные формулы, где школьники обычно ошибались, считаются на калькуляторе, который не убрать, а задачи на логику с выбором лучшего варианта можно решить полным перебором и т. д. Следующий шаг: усложнить задачи, тогда 27 задач, составляющих ЕГЭ (при этом восемнадцать заданий настоятельно рекомендуется решать, используя программы на Python)¹, просто не успеть написать, даже на компьютерах. Следующий шаг: в школьную программу активно включается Python – язык, обладающий развитыми библиотеками. Программы на Python самые короткие, их делают однообразными и раздают школьникам готовые – «выучи и повтори». Именно по этой причине Python доминирует над всеми другими языками в школьной программе за 2022–2023 гг.

Опрос осени 2023 года проводился в группе студентов первого курса, поступивших на направление «Прикладная математика и информатика», средний суммарный балл ЕГЭ при поступлении – 239 практически без дисперсии, при этом средний балл по информатике – 76,4 с вариациями от 64 до 95. Иными словами, опрашивали мотивированных выпускников с хорошими баллами по остальным предметам, таких

¹ Барышева И. В. Подготовка абитуриентов математических факультетов к ЕГЭ по информатике. Контрольные работы. Сайт электронного обучения ННГУ. URL: <http://www.unn.ru/t-ltaning/> (дата обращения: 16.06.2023).

учеников немного даже в приличных школах. Результаты опроса отображены в табл. 1. Создается ощущение, что Python вытеснил остальные языки из школьной программы. Естественно, в опросе появился еще один вопрос: «Где изучали?». Массовая переподготовка на использование нового языка программирования учителей, среди которых явно не очень много профессиональных программистов, многие из которых только-только научились работать с Pascal, с простыми логическими конструкциями, вызывает сомнения. «Где изучали?» – данный вопрос пришлось задать детям, по которым жестко прошла пандемия. Осенью 2021 г. разброс в названиях языков вызывал изумление, но свыше 83 % отвечали: «Изучал дома сам». Ответы в 2022 г. показали рост влияния Python, но свыше 80 % респондентов – отметили самостоятельный режим изучения.

Опрос текущего года пришлось адаптировать под ответы новых студентов и детализировать допустимые. Результат приведен в табл. 2.

Таблица 2

Место изучения программирования

где изучал	Pascal	группа С	Python
сам	3,57 %	21,43 %	57,14 %
с репетитором	–	–	14,29 %
в школе на уроках	21,43 %	–	25,00 %
в школе на факультативе	–	–	–
на платных курсах	–	7,14 %	32,14 %
на бесплатных курсах	–	–	–

В ответах необходимо было указать все варианты, например ответ мог быть: «в школе, на курсах и самостоятельно». При проведении опроса выяснялась самооценка первокурсников в отношении их знаний того или иного языка. Получилось, что в некоторых школах еще остался Pascal, но его не используют при подготовке к ЕГЭ, к экзамену 93 % учат Python, но только для выполнения заданий, а лучше всех знают варианты языка С, С++ или С#, которым занимаются с интересом. Необходимо отметить, что в опрашиваемой группе оказались выпускники лицеев, гимназий и школ с хорошими математическими традициями, которые исторически сложились и существуют в каждом большом городе. Принцип подбора групп произведен по суммарному баллу ЕГЭ, данная группа имела баллы чуть выше проходных или равные проходному.

Таким образом, в школе примерно поровну занимались и Pascal, и Python. Однако как только дело доходит до экзамена, то практически в ответах фигурирует один Python. Более того, экзамен по информатике совершенно естественно (с переходом его проведения на компьютере) стал экзаменом по программированию, а программирования в школьной программе крайне недостаточно: например, в 11-м классе

всего 6–8 часов за год, чуть больше в 10-м¹. Отдельно следует заметить, что знакомство и изучение языков группы С осуществляется вне школы и никогда не используется для сдачи экзамена, при этом интерес школьников примерно одинаков на протяжении времени наблюдения, что определяется их сложностью (особенно просто С), тематикой задач, которые пытаются решать в этом возрасте. Результат приведен в табл. 3.

Таблица 3

Самостоятельная оценка знаний по программированию

знания по программированию	Pascal	группа С	Python
не знаю	–	–	–
слышал	–	3 %	–
в рамках школьной программы	33 %	–	–
только для сдачи ЕГЭ	–	–	87 %
хорошо	–	23 %	7 %
всего	33 %	26 %	94 %

Результаты многолетнего анализа знаний выпускников школ в области программирования позволяют говорить о недостатках школьной программы по информатике, частью которой является программирование. При всей привлекательности профессий IT-индустрии только самые упорные и целеустремленные выпускники из успешных учебных заведений могут преодолеть барьер перехода в вуз.

2. Несоответствие школьной программы по информатике и вопросов единого государственного экзамена

Хаотичность изменения программы по информатике в гонке за взрывным развитием информационных технологий в условиях ограниченности часов школьной программы, которые выделяются под этот предмет, привела к потере качества во всех компонентах. Как в самой программе, так и в формулировке типов вопросов ЕГЭ, много громких, умных слов и названий тем, с которыми в условиях, отведенных на их изучение часов, даже обзорно познакомиться нереально².

Например, задание под номером 5 имеет название «Анализ и построение алгоритмов для исполнителей», пункт 2 «Посимвольное десятичное преобразование» имеет вид:

Тип 5 № 7751

Автомат получает на вход четырёхзначное число. По этому числу строится новое число по следующим правилам:

¹ Каталог заданий по темам. URL: <https://inf-ege.sdamgia.ru> (дата обращения: 16.06.2023).

² Образовательный портал для подготовки к экзаменам. Информатика. URL: <https://inf-ege.sdamgia.ru/> (дата обращения: 16.06.2023).

1. Складываются первая и вторая, а также третья и четвёртая цифры исходного числа.

2. Полученные два числа записываются друг за другом в порядке возрастания (без разделителей).

Пример. Исходное число: 2366. Суммы: $2 + 3 = 5$; $6 + 6 = 12$. Результат: 512. Укажите наибольшее число, в результате обработки которого автомат выдаст число 117.

В качестве решения предлагается программа на Python¹:

```
for i in range(10000, 1000, -1):  
    s = str(i)  
    k1 = int(s[0]) + int(s[1])  
    k2 = int(s[2]) + int(s[3])  
    first = str(min(k1, k2))  
    second = str((max(k1, k2)))  
    s1 = first + second  
    if s1 == '117':  
        print(i)  
        break
```

Программа содержит полный перебор вариантов с выбором подходящего, а где анализ? Для выполнения в таком виде задания необходимо уже быть знакомым с Python.

В программе по информатике для 9 класса присутствует тема, к которой можно отнести данное задание и на которую отведено в сумме 19 часов. Она имеет формулировку: «Основы алгоритмизации и объектно-ориентированного программирования», раздел «Алгоритм и его формальное исполнение», в задачи которого входит определение алгоритма, способы его представления, возможные исполнители алгоритмов. Более сложным является второй раздел данной темы: «Основы объектно-ориентированного визуального программирования», в котором предлагается обсудить все разделы программирования как такового – от типа данных до функций. Однако сюда не включены вопросы непосредственно объектно-ориентированного программирования, да они и не могут быть включены без достижения некоторого уровня знаний и компетенций модульного программирования.

Другой пример. Задание № 14 «Кодирование чисел. Системы счисления» имеет вид:

Тип 14 № 48387

Операнды арифметического выражения записаны в системах счисления с основаниями 11 и 19:

$$x341y_{11} + 56x1y_{19}$$

В записи чисел переменными x и y обозначены допустимые в данных системах счисления неизвестные цифры. Определите значения x и y , при которых значение данного арифметического выражения будет наименьшим и кратно 305. Для

¹ Каталог заданий по темам. URL: <https://inf-ege.sdangia.ru> (дата обращения: 16.06.2023).

найденных значений x и y вычислите частное от деления значения арифметического выражения на 305 и укажите его в ответе в десятичной системе счисления.

В качестве решения предлагается программа на Python¹:

```
result_search = []
for x in '0123456789A':
    for y in '0123456789A':
        t = int(x + '341' + y, 11) + int('56' + x + '1' + y, 19)
        if t % 305 == 0:
            result_search.append(t)
if result_search:
    print(min(result_search) // 305)
```

Изменяемые строки

В данной программе, в отличие от предыдущей, уже есть двойной цикл, преобразование типов, работа со строками, использование средств Python для перевода из одной системы в другую. Невозможно проследить навыки и понимание школьниками систем счисления, поскольку все скрыто внутри Python. Опять полный перебор. Необходимо отметить, что задания данного типа отличаются только видом исходного арифметического выражения и числом, которому должно быть кратно значение выражения. Вся подготовка по решению задания сводится к запоминанию программы и умению менять запись одной единственной строки и числа 305.

Наконец, последний пример. Задание № 17 «Обработка числовой последовательности» имеет вид:

Тип 17 № 37340

В файле содержится последовательность из 10 000 целых положительных чисел. Каждое число не превышает 10 000. Определите и запишите в ответе сначала количество пар элементов последовательности, разность которых четна и хотя бы одно из чисел делится на 31, затем максимальную из сумм элементов таких пар. В данной задаче под парой подразумевается два различных элемента последовательности. Порядок элементов в паре не важен.

Задания данного типа отличаются друг от друга понятием пары элементов. Для решения данного типа задач требуется считать данные из файла, преобразовать их к целому типу, правильно вести счет и определять экстремум. Сама программа содержит 9–10 строк, их можно запомнить, но для правильности получаемого ответа необходимо выполнить следующие действия:

- организовать директорию;
- скачать заданный в условиях задачи файл данных;
- сформировать тестовый файл данных, на котором проверяется алгоритм;
- написать программу;
- сохранить программу в той же директории;
- протестировать алгоритм на тестовом файле;

¹ Каталог заданий по темам. URL: <https://inf-ege.sdamgia.ru> (дата обращения: 16.06.2023).

– провести решение на заданном файле.

Для выполнения всех действий необходимо время. Приведенные примеры позволяют сделать вывод о фактическом преобразовании экзамена по информатике в экзамен по программированию. Напомним, что часов на программирование в школьной программе отводится слишком мало, заявленные цели не достигаются, а методика изучения программирования сводится к заучиванию готовых программ. В то же время следует отметить, что количество заданий в ЕГЭ увеличивается (табл. 4).

Отдельно следует обсудить задания 19, 20 и 21, в которых необходимо разработать стратегию игры. Сами по себе задания интересные, после переноса экзамена на компьютеры стало возможным для решения применение электронных таблиц типа Excel. Это сделало задачи гарантированно решаемыми, остался вопрос: сколько времени потребуется для выполнения всех действий?

Таблица 4

Рост числа заданий в ЕГЭ, требующих написание программ

	2015–2020	2021–2022	2023
Самостоятельное написание программ	2 задания	5 заданий	18 заданий
Язык программирования	Basic, Pascal, C, C++, школьный алгоритмический язык	Pascal, C++, Python	Python, редко Pascal

Использование электронных таблиц в задании № 3 совершенно справедливо, так как задание направлено на проверку умения работать с базами данных, организованных на базе Excel, хотя для выпускного экзамена такая задача слишком элементарна. В 2023 г. появился вопрос № 22 «Многопроцессорные системы». Однако задание, которое должно отражать проблему, сводится к ручному счету по заданной таблице, в лучшем случае приходится использовать готовые функции «текст по столбцам» или «ВПР», которые есть в меню Excel. Еще более дорогостоящий по времени и плохо соответствующий программе по информатике вопрос № 18 «Робот – сборщик монет». Времени требует много, особенно если поле, по которому ходит сборщик, имеет границы непроходимости или особые точки. Еще один неутешительный вывод: время проведения экзамена перестало соответствовать времени, необходимому для решения двадцати семи задач в том виде, который они приобрели за последние два года.

3. Определение целей и мотивации школьников разных возрастных категорий при разработке концепции общей программы по информатике

Время «набора эмпирических данных» для формирования школьной программы в области информационных технологий прошло. Наступило время создания стройной системы взаимосвязанных элементов, соответствующих каждому возрасту и уровню развития школьников. Недопустимо включать все модное, громко звучащее без должного анализа необходимости с целью формирования нужных компетенций

современного человека, с одной стороны, с другой – для формирования базовых знаний, составляющих фундамент развития профессиональных качеств для IT-индустрии, которая в современном мире является определяющей составляющей развития общества. В табл. 5 представлены данные, демонстрирующие компетенции, необходимые современному школьнику.

Таблица 5

Таблица необходимых компетенций школьников по информационным технологиям

Информатика для пользователей	Современные разделы математики
Начальная школа	9-й класс
Игры и техника безопасности, безопасность при работе в интернет, презентации	алгебра логики, системы счисления, принципиальная схема устройства компьютера
Среднее звено	10 – 11-й классы
Презентации, электронные таблицы, базы данных, редакторы	программирование

В начальной школе, хотим мы этого или не хотим, дети добрались до интернета. Это факт, который необходимо принимать во внимание при формировании программы. Главная задача – безопасность детей, информационная безопасность, безопасность общения, безопасность для семьи и т. д. Программа должна разбирать круг, образованный обеспечением безопасности. Компьютерные игры также выставляют свои требования: и дозирование, и соответствие возрасту, и отсутствие элементов агрессии. При этом нельзя забывать о привлекательной стороне использования компьютера для школьников начальных классов. С точки зрения реальных приложений, для формирования интереса можно рассматривать, например, систему «Кумир». Разработка презентаций как задания по различным школьным предметам вполне доступна для понимания детей. Количество часов может быть небольшим.

Среднее звено – это дети переходного возраста, с ними можно разговаривать как со взрослыми, и цели обучения ставить серьезные: формирование компетенций образованного молодого человека XXI века. К таким компетенциям необходимо отнести: умение грамотно составить документы, используя редакторы, умение работать с базами данных, разработка простейших баз данных с помощью электронных таблиц, умение вести статистику своей деятельности. Иными словами, использование возможностей современных достижений IT-индустрии в повседневной жизни: электронный дневник, школьный чат, запись к врачу, электронная покупка билетов, онлайн-магазины. В этот период уже можно объяснить, что и интернет, и электронная почта, и искусственный интеллект, и облачные хранилища, и различные чаты для общения – продукт человеческого разума, и у каждого есть собственник, в руках которого находится вся информация, которая создается пользователем различных приложений. В школьную программу по информатике необходимо ввести знакомство с принципами организации общеизвестных пользовательских приложений.

Изучение программирования в рамках школьной программы – сверхсложная задача, причин сложности достаточно много:

- программирование как часть современной математики – сложная наука;
- бурное развитие информационных технологий и программирования (например, частые изменения школьной программы) не позволяют поддерживать массовую подготовку учителей на должном уровне;
- школьники обладают разными способностями в освоении сложных предметов;
- дифференциация интересов и умений школьников в рамках одного класса создают дополнительные трудности, падает мотивация.

К таким выводам приводит анализ статистики данных ЕГЭ по школам, единый государственный экзамен по информатике сдают выпускники гимназий, лицеев и школ со сформированными математическими традициями, в обычных школах редко выбор учеников попадает на информатику. Данные о школе подтверждаются и результатами опроса первокурсников.

В условиях дефицита кадров для IT-индустрии необходимо расширить круг школьников, интересующихся информационными технологиями, а в рамках школьной программы сформировать фундамент для дальнейшего изучения всех аспектов и сложных разделов программирования. Немаловажной задачей школьной программы по информатике и программированию является задача показать школьникам и их родителям, что факт легкого общения с компьютером определяется грамотным интерфейсом программного обеспечения, выполненного профессиональными программистами. Те из них, что выберут IT-индустрию как область своей профессиональной деятельности, должны разрабатывать, создавать, тестировать, сопровождать различные приложения и программное обеспечение.

В рамках поставленных целей необходимо пересмотреть содержание школьной программы по программированию не только с точки зрения часов, отводимых для данного раздела, но и с точки зрения содержания и методики преподавания.

В 9 классе следует знакомить с принципиальной схемой компьютера: как техническим прибором, устроенном по законам физики, и устройством, способным работать по математическим законам. Связующим звеном между физикой и математикой является алгебра логики, операции которой реализуются определенными электрическими схемами. Соответственно, алгебра логики должна быть темой для изучения. Но алгебра логики не работает с числами, тогда следующий раздел для изучения – системы счисления, рассматривать которые необходимо с точки зрения истории появления потребности в счете, формирования систем счисления, позиционные и непозиционные системы, появление нуля, другими словами, математической теории систем счисления и применение результатов этой теории в компьютерах. Трех тем вполне достаточно для девятого класса.

К сожалению, методика преподавания программирования достаточно часто сводится к изучению элементов одного языка с решением мелких задач, начало которых звучит примерно так: «есть числа...» или «задан файл, содержащий числа...». В этом случае школьники не видят связи между решаемыми задачами и всем тем прекрасным, многоликим и разнонаправленным, к чему они уже привыкли при использова-

нии планшета, домашнего компьютера или ноутбука. Неограниченность круга вопросов, которые пытаются включить в школьную программу, – это рекурсивные функции и объектно-ориентированное программирование, многопроцессорные системы и много всего другого при незначительном объеме выделенных часов превращает изучение данных тем в ознакомление с названиями.

Необходимо четко сформулировать цели школьного курса программирования. Их две: сформировать фундамент для изучения сложной науки программирование для тех, кто может и хочет заниматься в жизни информационными технологиями, максимально расширив круг интересов; показать всю сложность работы в IT-индустрии для тех, кто по тем или иным причинам не сможет изучать столь сложный раздел достижений человечества.

Изучение программирования можно представить в виде совокупности шагов¹:

- I) базовые типы данных, поддерживаемых компьютером;
- II) базовые структуры хранения;
- III) именованное объектов;
- IV) определение алгоритма и структура программы;
- V) базовые конструкции программирования:
 - 1) интерфейсные операции (ввод\вывод),
 - 2) оператор присваивания,
 - 3) реализация ветвления,
 - 4) организация повторений;
- VI) базовые алгоритмы:
 - 1) суммирование,
 - 2) поиск экстремума,
 - 3) отбор по признаку,
 - 4) линейный поиск,
 - 5) сортировка,
 - 6) кратность:
 - список без повторений,
 - определение рейтингов,
 - подсчет суммарной характеристики;
 - 7) деление строки с одним разделителем на слова,
 - 8) деление строки с несколькими разделителями на слова,
- VII) содержательные задачи окружающего мира.

При этом пункты V и VI должны разбираться вместе, по мере обсуждения очередного базового алгоритма появляется возможность знакомиться с новыми базовыми конструкциями. После освоения первых трех базовых алгоритмов можно начи-

¹ Барышева И. В. Подготовка абитуриентов математических факультетов к ЕГЭ по информатике. Контрольные работы. Сайт электронного обучения ННГУ. URL: <http://www.unn.ru/t-ltaning/> (дата обращения: 04.03.2024); Рабочая учебная программа по информатике (в соответствии с ФГОС). URL: https://karaul-school.astr.eduru.ru/media/2019/10/22/1265929801/inf_7-9.pdf (дата обращения: 04.03.2024)

нать решать содержательные задачи¹, которые возникают из всего того, что окружает современного человека: касса магазина даст задачи по маркетингу, простой чек из магазина также может быть решением соответствующей задачи, статистика спортивных соревнований, формирование списка поступивших, проверка грамматики в редакторе и т. д. Задачи должны иметь содержательную формулировку, которая отражает связь программирования с реальной действительностью, развивает интерес школьников к изучению программирования, формирует фундамент для изучения сложных конструкций программирования.

Опасным явлением при изучении программирования оказалась дань моде – срочный перевод школьного программирования на Python (см. табл. 2, 3). Python на фоне других языков выглядит проще, имеет богатые библиотеки и встроенные структуры хранения данных. Python как ни один другой язык подходит для сдачи ЕГЭ: можно использовать как калькулятор, декларации типов не требует, вместо алгоритмов – сортировки, определение суммы или поиска по признаку, можно одной строкой вызвать соответствующую функцию из библиотеки, работа со строковыми данными спрятана в готовых разработках, программы получаются иногда короче вдвое их аналогов на других языках. Но, если главная цель обучения в школе – не сдача ЕГЭ, а формирование думающего человека, готового к развитию и принятию быстро меняющегося мира, когда образование – это не сумма заученных инструкций, а возможность понимания всего того нового, что с огромной скоростью появляется в сфере информационных технологий, тогда необходимо ориентироваться на другие характеристики и свойства языка программирования, с которого начинается обучение. Инкапсуляция многих основополагающих элементов программирования в Python не позволяет сформировать компетенции обучающихся, связанные с научной стороной программирования². Python – это технология программирования, а технологии устаревают по мере развития науки. Целью создания языка Python является решение прикладных задач опытными программистами, для которых переход на Python связан с упрощением внешнего вида программ и для которых эффективность по памяти и времени получаемых проектов не носит критический характер.

Язык Pascal в свое время был создан специально для изучения программирования, его появление было этапом эволюционного развития программирования как науки. Практическое использование Pascal на данном этапе в основном пережило себя, но значимость Pascal как первого языка обучения программированию трудно переоценить.

В Pascal четко определены важные научные понятия программирования:

- типы данных, как базовые, так и конструируемые;
- переменные / объекты, обладающие именем, адресом и типом;

¹ Барышева И. В., Гергель В. П., Городецкий С. Ю. и др. Информатика для абитуриентов. Задачи и решения. Нижний Новгород: Издательство Нижегородского госуниверситета, 2007. 360 с.

² Образовательный портал для подготовки к экзаменам. Информатика. URL: <https://inf-ege.sdamgia.ru/> (дата обращения: 16.06.2023).

– модульное программирование с «передачей параметров по значению» и «передачей параметров по адресу», формальные и фактические параметры; распределением проекта файлах и т. д.

Более того, организация библиотек с помощью специальных модулей `unit` в Pascal может быть рассмотрена как прообраз появления объектно-ориентированного программирования (ООП): управление доступом к элементам в виде интерфейсной секции и секции реализации, подключение к основному файлу и даже секция инициализации может быть трактована как прообраз конструктора. Таким образом, Pascal, с одной стороны, – язык практического программирования для начинающих, с другой стороны, – язык, формирующий фундаментальные понятия программирования как быстроразвивающейся науки. Однако из табл. 2 и 3 видно, как в последние годы Pascal исчезает из школьной программы.

Языки C/C++ в определенном смысле еще лучше, чем Pascal, но для школьного программирования они сложны. Например, алгоритмический язык C имеет сложный интерфейс, операторы ввода/вывода требуют спецификаторы, связанные с типами данных, и адресацию объектов. Формирование C происходило в период, когда информация, обрабатываемая на компьютере, была в основном числовой, поэтому у C весьма скромная библиотека работы со строками. При этом реализация строк как массивов символов требовала работы с указателями, наиболее эффективный вариант – динамический массив, который даже для студентов профильных специальностей представляет сложности. Главной прерогативой C++ является объектно-ориентированное программирование (ООП), но изучение данной технологии точно не для школы – и по уровню сложности данной темы и по времени, необходимом для понимания, да и уровень школьных задач не предполагает использование ООП. C++ без объектов по своей идеологии, по реализации обязательных элементов программирования имеет много общего с Pascal, поэтому разумно разделение: в школе изучается Pascal, в вузе – C и C++. Такой вариант позволяет выявить закономерности программирования как такового, понять тенденции развития языков программирования, принять мысль о том, что языки программирования имеют несколько этапов: возникновение, расцвет, зрелый период и закат, а программирование как наука, как технология при этом получает новые стороны развития. Необходимо понимать быстротечный характер «жизни» языков программирования. Важнейшей компетенцией в области программирования является знание внутренней структуры самого программирования, которая проявляется и получает реализацию в каждом конкретном языке, умение безболезненно за разумное время переходить с одного языка на другой. Переход с Pascal, C, C++ на более простой по форме языка Python, особенно для специалистов, знакомых со структурами данных не представляет труда, обратный переход начинается с изучения программирования практически с нуля.

Выводы

Для начинающих изучать программирование необходимо знакомство с основополагающими понятиями. Pascal, как академический язык, как наследник Алгола, созданного международной группой ученых, язык, имеющий в основе формальную грамматику, теоретически грамотно и последовательно формирует профессиональ-

ное мировоззрение, понимание ключевых положений программистской культуры. Этот язык, в котором заложены идеи дальнейшего развития программирования, может использоваться как первый язык для начинающих программистов.

Назрела необходимость введения в школьную программу самостоятельного предмета «программирование», она обусловлена следующими факторами:

- высокая потребность в IT-специалистах,
- большой интерес к IT-индустрии у школьников,
- наличие большого числа различных курсов, часто платных, восполняющих недостаток школьной программы,
- ранняя профориентация школьников, позволяющая правильно определить свои возможности,
- несоответствие школьной программы и содержимого заданий ЕГЭ, в которых де-факто подразумевается написание программ.

Изучение языков программирования, алгоритмизация должны присутствовать в школьной программе не только с целью освоения предмета, но и как фактор формирования логического и математического мышления. Умение разбить задачу на простые шаги, оптимизировать выполнение каждого шага и всей задачи в целом, правильно определить все ответвления – эти навыки в школе может дать только программирование, а пригодятся они отнюдь не только программистам.

Список источников

Барышева И. В., Гергель В. П., Городецкий С. Ю. и др. Информатика для абитуриентов. Задачи и решения. Нижний Новгород: Издательство Нижегородского госуниверситета, 2007. 360 с.

Барышева И. В., Козлов О. А. Формирование структурного мышления школьников в процессе обучения программированию в рамках школьного курса информатики // Вопросы современной науки / под редакцией Н. Р. Красовской. Москва: Интернаука, 2016. Т. 14. С. 112–129.

Барышева И. В. Подготовка абитуриентов математических факультетов к ЕГЭ по информатике. Контрольные работы // Сайт электронного обучения ННГУ. URL: <http://www.unn.ru/t-ltaning/> (дата обращения: 16.06.2023).

Каталог заданий по темам. URL: <https://inf-ege.sdamgia.ru> (дата обращения: 16.06.2023).

Образовательный портал для подготовки к экзаменам. Информатика. URL: <https://inf-ege.sdamgia.ru/> (дата обращения: 16.06.2023).

Рабочая учебная программа по информатике (в соответствии с ФГОС). URL: https://karaul-school.astr.eduru.ru/media/2019/10/22/1265929801/inf_7-9.pdf (дата обращения: 16.06.2023)

References

Barysheva I. V., Gergel' V. P., Gorodetskii S. Yu. i dr. *Informatika dlia abiturientov. Zadachi i resheniia* [Informatics for applicants. Problems and solutions]. Nizhnii Novgorod: Izdatel'stvo Nizhegorodskogo gosuniversiteta, 2007. 360 p.

Barysheva I. V., Kozlov O. A. *Formirovanie strukturnogo myshleniia shkol'nikov v protsesse obucheniia programmirovaniu v ramkakh shkol'nogo kursa informatiki* [Formation of schoolchildren's structural thinking in the process of teaching programming in the school course of informat-

ics]. *Voprosy sovremennoi nauki* [Issues of modern science]. Moscow: Internauka, 2016, vol. 14, pp. 112–129.

Barysheva I. V. Podgotovka abiturientov matematicheskikh fakul'tetov k EGE po informatike. Kontrol'nye raboty [Preparing applicants of mathematical faculties for the USE in Informatics. Tests]. *Sait elektronnoogo obucheniya NNGU* [NNSU e-learning site]. Available at: <http://www.unn.ru/t-ltaning/> (accessed: 16.06.2023).

Katalog zadaniy po temam [Catalogue of tasks by topics]. Available at: <https://inf-ege.sdangia.ru> (accessed: 16.06.2023).

Obrazovatel'nyi portal dlia podgotovki k ekzamenam. Informatika [Educational portal for exam preparation. Informatics]. Available at: <https://inf-ege.sdangia.ru/> (accessed: 16.06.2023).

Raboचाia uchebnaia programma po informatike (v sootvetstvii s FGOS) [Academic working programme on Informatics (in accordance with FSES)]. Available at: https://karaul-school.astr.eduru.ru/media/2019/10/22/1265929801/inf_7-9.pdf (accessed: 16.06.2023).

Сведения об авторах

Олег Александрович Козлов – доктор педагогических наук, профессор, ведущий научный сотрудник; <https://orcid.org/0000-0002-4148-2726>, ole-kozlov@yandex.ru, ФГБНУ «Институт стратегии развития образования» (д. 16, ул. Жуковского, 101000 Москва, Россия); **Oleg A. Kozlov** – Doctor of Pedagogical Sciences, Professor, Chief Researcher, <https://orcid.org/0000-0002-4148-2726>, ole-kozlov@yandex.ru, The Federal State Budget Scientific Institution "Institute for Strategy of Education Development" (16, ul. Zhukovskogo, 101000 Moscow, Russia).

Ирина Викторовна Барышева – ibar1950@yandex.ru, ФГБНУ «Институт стратегии развития образования» (16, ул. Жуковского, 101000 Москва, Россия); **Irina V. Barysheva** – ibar1950@yandex.ru, The Federal State Budget Scientific Institution "Institute for Strategy of Education Development" (16, ul. Zhukovskogo, 101000 Moscow, Russia).

Заявленный вклад авторов: О. А. Козлов – 60 %, И. В. Барышева – 40 %. Авторы заявляют об отсутствии конфликта интересов.

Contributions of the authors: Oleg A. Kozlov – 60 %, Irina V. Barysheva – 40 %. The authors declare no conflict of interests.

Статья поступила в редакцию 25.10.2023; одобрена после рецензирования 10.01.2024; принята к публикации 24.01.2024.

The article was submitted 25.10.2023; Approved after reviewing 10.01.2024; Accepted for publication 24.01.2024.