

**В.А. Локалов, И.В. Климов,
Ю.О. Константинова, А.С. Миронов**

**МЕТОДИЧЕСКИЕ ОСОБЕННОСТИ
ПРЕПОДАВАНИЯ КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ В СИСТЕМЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДЕТЕЙ**



**Санкт-Петербург
2020**

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ ИТМО

**В.А. Локалов, И.В. Климов,
Ю.О. Константинова, А.С. Миронов**
**МЕТОДИЧЕСКИЕ ОСОБЕННОСТИ
ПРЕПОДАВАНИЯ КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ В СИСТЕМЕ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ ДЕТЕЙ**

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

РЕКОМЕНДОВАНО К ИСПОЛЬЗОВАНИЮ В УНИВЕРСИТЕТЕ ИТМО
по направлению подготовки 44.03.04 Профессиональное обучение
в качестве учебно-методического пособия для реализации основных
профессиональных образовательных программ
высшего образования бакалавриата

 УНИВЕРСИТЕТ ИТМО

Санкт-Петербург
2020

Локалов В.А., Климов И.В., Константинова Ю.О., Миронов А.С.,
**МЕТОДИЧЕСКИЕ ОСОБЕННОСТИ ПРЕПОДАВАНИЯ КОМПЬЮТЕРНЫХ
ТЕХНОЛОГИЙ В СИСТЕМЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
ДЕТЕЙ**– СПб: Университет ИТМО, 2020. – 53 с.

Рецензент(ы):

Смолин Артем Александрович, кандидат философских наук, доцент (квалификационная категория "ординарный доцент") факультета программной инженерии и компьютерной техники, Университета ИТМО.

Учебно-методическое пособие знакомит учащихся с методическими особенностями разработки и реализации образовательных программ дополнительного образования детей в области компьютерных технологий. Наряду с общими принципами организации системы компьютерных курсов рассматриваются подходы к решению наиболее актуальных методических проблем при обучении по таким востребованным направлениям как 3D-графика, компьютерные сети и программирование. Пособие базируется на многолетнем педагогическом опыте авторов, полученном при работе в Детско-юношеском компьютерном центре Университета ИТМО. Пособие предназначено для студентов, проходящих курс «Педагогическое проектирование и технологии» и обучающихся по направлению «Профессиональное обучение» (образовательная программа «Компьютерные технологии в дизайне»).



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2020

© Локалов В.А., Климов И.В., Константинова Ю.О., Миронов А.С., 2020

Введение

Учебно-методическое пособие «Методические особенности преподавания компьютерных технологий в системе дополнительного образования детей» предназначено для студентов, проходящих курс «Педагогическое проектирование и технологии» и обучающихся по направлению «Профессиональное обучение» (образовательная программа «Компьютерные технологии в дизайне») в Университете ИТМО.

Одна из задач дисциплины «Педагогическое проектирование и технологии» — дать учащимся возможность познакомиться с широким спектром методических и педагогических проблем, с которыми педагоги профессионального обучения часто встречаются в процессе своей практической деятельности, в частности, при педагогическом проектировании и проведении занятий. Для грамотного решения данных проблем у студентов должно быть сформировано системное мышление, которое позволит им понять логику организации и взаимосвязь дополнительных образовательных программ в области компьютерных технологий.

Раздел «Принципы проектирования системы компьютерных курсов» настоящего пособия включает обобщение части теоретического материала для раздела «Педагогическое проектирование» рабочей программы дисциплины «Педагогическое проектирование и технологии», на СРО в котором отводится 25,2 часа, из которых 3 часа — на изучение этого раздела. В этом разделе пособия на примере курсов Детско-юношеского компьютерного центра Университета ИТМО рассматриваются основные принципы проектирования системы учебных курсов по компьютерным технологиям, которые обеспечивают возможность свободного выбора траекторий обучения школьниками, а также позволяют учащимся приобрести базовые профессиональные навыки в выбранном направлении компьютерных технологий.

В качестве теоретической основы для построения модели профессионального развития школьников в системе дополнительного образования были использованы:

- Модель структуры интеллектуальных способностей Дж. Гилфорда [6].
- Концепция зоны ближайшего развития Л.С. Выготского [4].
- Теория деятельности Леонтьева [13].
- Модель поэтапного формирования умственных действий П.Я. Гальперина [5].

Проектирование учебного содержания каждого курса по компьютерным технологиям должно вестись на основе поиска соответствующей последовательности усложняющихся учебных задач. Это обеспечивает как гибкую подстройку учебного содержания под конкретный контингент учащихся, так и возможность формализации содержания обучения.

В следующих трёх разделах «Методические особенности и проблемы обучения 3D-графике и анимации», «Методические особенности и проблемы обучения программированию», «Методические особенности и проблемы обучения по программе «Сети и Web-проектирование»» рассматриваются методические особенности и проблемы обучения по таким востребованным в системе дополнительного образования направлениям обучения, как 3d-графика, компьютерные сети и программирования. Указанные материалы служат как примеры описания методических особенностей и проблем обучения и возможных вариантов их решения для раздела «Педагогическое проектирование» рабочей программы дисциплины «Педагогическое проектирование и технологии», на СРО в котором отводится 25,2 часа, из которых 6 часов — на изучение этих разделов пособия: по 2 часа на каждый. Также эти разделы содержат сведения, которые позволят студентам, преподающим эти или похожие направления компьютерных технологий школьникам, лучше настроить систему задач на актуальный уровень развития и мотивации своей учебной группы.

В заключительном разделе пособия приводится список ошибок, которые часто встречаются при проектировании и проведении занятий начинающими преподавателями, а также рекомендации по устранению этих ошибок. Этот раздел пособия соотносится с разделом «Прогнозирование и оценка результатов профессионально-педагогической деятельности» рабочей программы дисциплины «Педагогическое проектирование и технологии», на СРО в котором отводится 30 часов, из которых 2 часа — на изучение этого раздела пособия.

Принципы проектирования системы компьютерных курсов

С конца 90х годов XX века в связи с растущими потребностями учащихся в дополнительном качественном образовании по информатике начинают стремительно развиваться компьютерные курсы для школьников. В это время в школьном образовании шли дискуссии между сторонниками теоретической концепции школьной информатики академика А.П. Ершова и сторонниками так называемого «пользовательского подхода» [10].

А.П. Ершов считал, что алгоритмические и программистские навыки являются фундаментальными компонентами человеческой деятельности, а их формирование является основной целью национальной программы школьной информатики. Представление о том, что «программирование — вторая грамотность» породило ориентацию школьных образовательных программ преимущественно в этом направлении. В противовес этому сторонники «пользовательского подхода» предлагали сосредоточить внимание учащихся на практическом применении компьютерных приложений, таких как офисные программы, графические, мультипликационные и прочие редакторы.

Государственная образовательная система пыталась объединить оба эти принципа (программистский и пользовательский) в рамках одной обязательной программы по информатике. Эта попытка потерпела неудачу, поскольку ограниченный бюджет школьного учебного времени, отведенного на информатику, принципиально не мог вместить подготовку по всем направлениям компьютерных технологий. Кроме того, существовали проблемы с методическим и с техническим обеспечением, а также с подготовкой преподавателей по такой быстро развивающейся отрасли, как информатика.

Указанные выше проблемы не только не нашли своего решения в дальнейшем, но и в настоящее время во многом усугубились. Это послужило причиной возникновения большого числа дополнительных образовательных структур, занимающихся преподаванием компьютерных технологий. Такие структуры чаще всего ориентируются на запросы рынка образовательных услуг, а их методисты и преподаватели, пользуясь возможностью абсолютно свободно формировать дополнительные образовательные программы, обычно не обращают внимания на необходимость системного подхода к проектированию и реализации системы компьютерных курсов.

Отсутствие продуманных, теоретически обоснованных принципов проектирования образовательных программ компьютерных курсов нередко приводит к казусам. Например, первый уровень на достаточно популярных компьютерных курсах, который называется «Базовая подготовка», почему-то включает в себя изучение робототехники и прикладной математики. Эта базовая

подготовка предназначена для школьников, которые могли бы называть себя «академиками», поскольку система курсов, которые они посещают, называется «Академией».

Можно привести еще много подобных примеров. Их существование подтверждает необходимость разработки принципов проектирования неформальных образовательных систем в области информатики, которые бы базировались на применении объективных психологических законов обучения и развития, то есть на психологической педагогике. Многоплановость и разнообразие задач, стоящих перед дополнительным образованием школьников в области информатики, делают поиск таких принципов нетривиальной задачей. Разный возраст и уровень развития детей, их разные интересы, различная мотивация и заинтересованность родителей учащихся — все это привело к необходимости сформулировать сначала некоторые *общие идеи*, реализация которых позволила бы шаг за шагом найти адекватные этим идеям принципы проектирования образовательной структуры. Поиск привел к следующим *основным идеям*:

- свобода выбора и свобода творчества (как мотивационное начало);
- развитие способностей (объективно обеспечивает возможность выбора);
- постепенная специализация в процессе обучения (как форма самореализации);
- профессиональная направленность обучения (обеспечивает формирование базовых профессиональных навыков, ориентированных на общественные запросы).

Образовательная система должна объединить эти идеи таким образом, чтобы процесс творчества, осознанный и эмоционально пережитый учащимся на самых ранних этапах обучения, дал ему в дальнейшем энергию для определения своих интересов в области информатики и дальнейшего профессионального развития. В этом процессе компьютер должен играть роль не более чем инструмента в профессиональной и творческой деятельности. Однако на пути реализации этих идей возникают некоторые подводные камни, которые являются следствием специфики компьютера как инструментального средства, используемого в процессах решения задач.

В традиционной теории культурно-исторического развития в качестве инструментальных средств мышления рассматривались знаковые системы вообще и речь в частности. Часть инструментальных действий со знаковыми системами в процессе развития переходит во внутренний план и автоматизируется. Например, устный счет осваивается сначала на материальных объектах (счет на пальцах, по палочкам и т.п.), а затем постепенно может происходить в уме. В уме автоматизированные операции, типа сложить 2 и 3,

выполняются гораздо быстрее, чем, например, на калькуляторе. Выполнение в уме *простых* арифметических операций — *новая* интеллектуальная *способность*, которую мы приобретаем в процессе развития, и она, безусловно, оказывает влияние на развитие каких-то других интеллектуальных способностей, таких как оценка параметров объектов, находящихся вокруг нас, принятие решений, прогнозирование событий и пр.

Но что будет, если компьютер «поможет» нам тем, что возьмет на себя автоматизацию простых арифметических операций? Очевидно, это повлияет и на развитие прочих мыслительных процессов, использующих эти вычислительные способности нашего ума. Перенос ряда интеллектуальных операций во внешний план приводит к тому, что человек, работающий за компьютером, начинает идти на поводу у внешних автоматизированных операций. Так, например, вместо того, чтобы искать композиционное решение в уме, дизайнер начинает гонять объекты по экрану в поисках наиболее удачного варианта. В конечном итоге человек, чрезмерно полагаясь на интеллектуальные возможности компьютера, ограничивает развитие своих собственных интеллектуальных способностей.

Чтобы избежать таких негативных последствий изучения информатики, необходимо сделать так, чтобы учащиеся, решая стоящие перед ними задачи, могли абстрагироваться как от интерфейса, так и от конкретных компьютерных технологий. Это же требование в терминах теории деятельности будет звучать как решение задачи с помощью действий, имеющих высокий уровень обобщения.

Учебная задача, как и всякая проблемная ситуация, является необходимым условием развития интеллектуальных способностей каждого учащегося при условии, что она соответствует его зоне ближайшего развития, то есть может быть решена только в сотрудничестве с преподавателем. Разброс по уровню подготовки учащихся, характерный для дополнительного образования, приводит к необходимости разработки *системы учебных задач*, сложность которых меняется в значительном диапазоне. Поэтому разработка принципов усложнения учебных задач является наиболее важной составляющей проектирования образовательной системы, обеспечивающей как ее концептуальную целостность, так и адаптивные возможности [19]. В системе, спроектированной по принципу усложняющихся задач, будет реализована *идея развития способностей*, поскольку проблемная ситуация является необходимым условием для возникновения интеллектуальных реакций.

Следующая идея — *идея постепенной специализации* в процессе профессиональной подготовки — хорошо известна в профессиональном образовании. При формировании концепции образовательной системы ДЮКЦ за основу была взята традиционная модель постепенной специализации, состоящая из трех уровней подготовки: 1 – начального, 2 – основного, 3 – углубленного. Каждый из уровней характеризуется как степенью специализации, так и особыми

принципами построения системы задач, позволяющих сформировать обобщенные действия.

Начальный уровень обучения информатике предназначен для тех, кто вообще не знаком, частично знаком или недостаточно полно знаком с возможностями применения различных областей информатики. В среднем такая подготовка примерно соответствует возрасту 9 – 12 лет.

Как и любой человек, попавший в незнакомое или недостаточно хорошо знакомое место, учащийся прежде всего нуждается в широкой *ориентировке* [5]. Чем шире она будет, тем лучше, то есть на начальном уровне должна отсутствовать какая бы то ни было специализация. На этом уровне учащиеся должны получить первое представление почти обо всех типах задач, которые связаны с использованием компьютера, о способах организации продуктивной работы за компьютером, а также о принципах работы и устройстве самого компьютера. Они учатся работать в операционных оболочках и средах, изучают основные функциональные элементы компьютера, знакомятся с прикладными программами различного типа (текстовыми, 2D и 3D графическими редакторами, архиваторами, электронными таблицами, средствами Интернет и коммуникаций, мультимедиа и др.). В процессе обучения начинают формироваться базовые понятия, необходимые для работы с компьютером: «интерфейс», «команда», «программа», «конфигурация», «операционная система» и др. В конечном счете учащиеся приобретают знания и развивают умения, без которых не обойтись ни в одной области информатики.

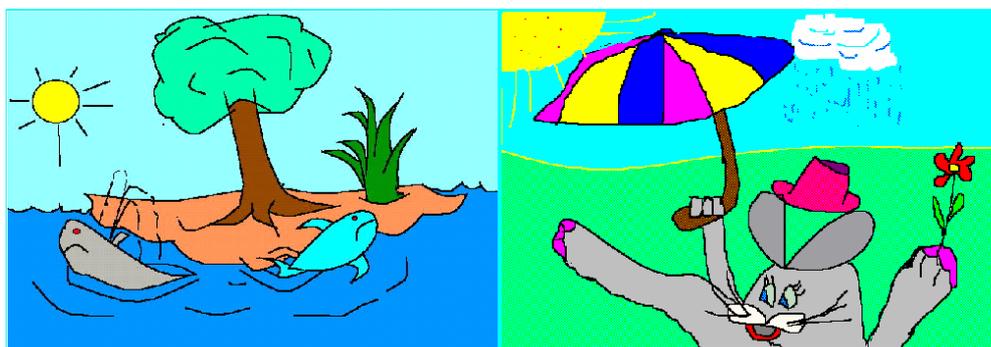
Принцип концентрации внимания на задаче, а не на средствах ее реализации, для начального уровня особенно актуален. Широкая вариативность изучаемых средств должна предполагать такую же вариативность задач, которые используют эти средства. Такая вариативность наиболее благоприятна для развития креативности. Поэтому *реализация идеи развития творческих способностей* может быть рассмотрена в качестве одной из основных задач начального уровня.

Рассмотрим пример, иллюстрирующий возможность сочетания задачи на креативность с отработками навыков рисования в графическом редакторе. Учащемуся дается некоторое исходное изображение и дается задача «придумать и нарисовать, что это может значить/означать» (рис. 1). Основная активность сосредотачивается именно на задаче, а освоение разнообразных графических возможностей инструмента происходит в результате ее решения. Похожий подход можно применить при изучении широкого спектра инструментальных средств, вплоть до работы с операционной системой, при этом развиваются не только инструментальные умения, но и креативность.



а)

б)



в)

г)

Рис. 1. Задание «Что бы это значило?». Придумать и нарисовать как можно больше вариантов исходного изображения: а) исходное изображение; б), в), г) варианты.

При выборе спектра задач очень важно ориентироваться на интересы учащихся, чтобы обеспечить высокий уровень мотивации. Поэтому темы заданий должны быть связаны со сферой актуальных потребностей, а форма их выполнения — быть игровой.

Начальный уровень характеризуется не только отсутствием специализации, но и особыми принципами построения системы усложняющихся задач. Поскольку основной целью данного уровня является развитие общих и специальных креативных способностей, в качестве основной системы классификации удобно использовать модель структуры интеллекта по Гилфорду [5].

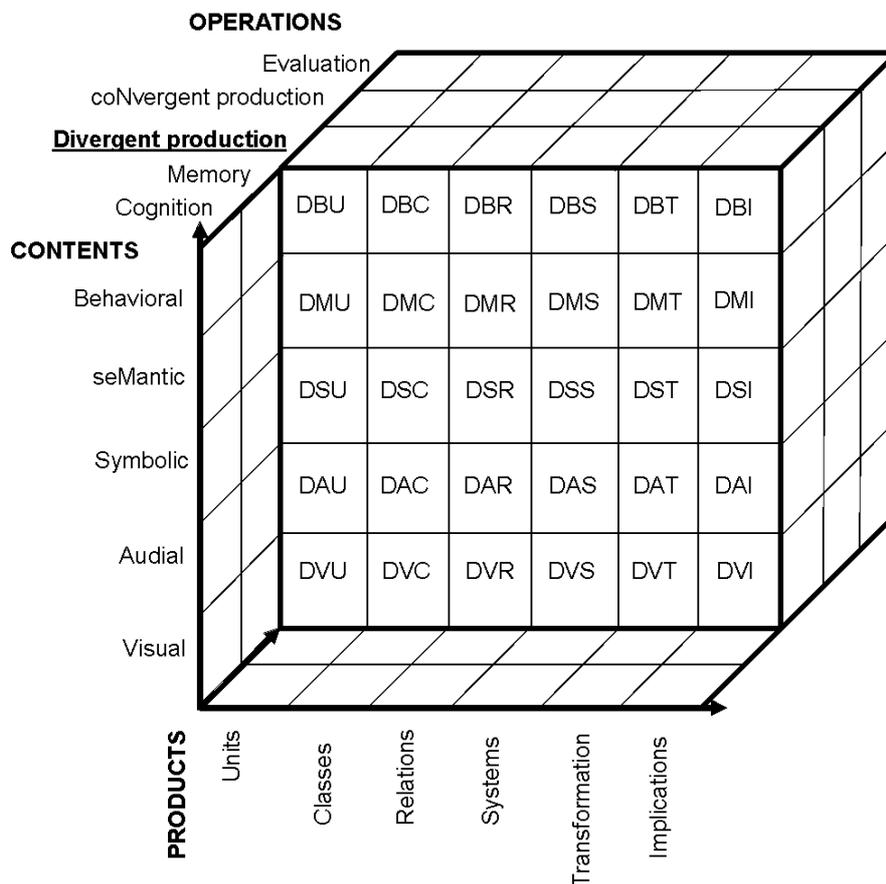


Рис 2. Структура интеллектуальных способностей по Гилфорду и дивергентные способности интеллекта (в сечении).

Она отображает систему элементарных интеллектуальных способностей в виде точек в пространстве трех координат (содержание, процессы, продукты) и изображается обычно в виде параллелепипеда. Для этого параллелепипеда можно указать оси, вдоль которых идет усложнение интеллектуальных процессов. Дивергентные способности являются сечением этого параллелепипеда (рис. 2):

Каждой задаче можно сопоставить несколько интеллектуальных способностей, необходимых для ее решения. Например, для построения разнообразных композиций в задаче «Чтобы это значило» нужны дивергентные способности (D) создания систем (S) на визуальном (V) содержании. В координатах модели Гилфорда эти способности обозначаются как DVS, а чтобы получить различные варианты изображений из одного и того же рисунка — дивергентные визуальные импликации (DVI).

Рассмотрим процесс решения задачи в виде многоуровневой интеллектуальной активности (рис. 3а). Самый верхний уровень отображает устремление к творческой цели (P_x), это стремление приводит к активизации дивергентных способностей (D_{xx}), отвечающих за творческий поиск, который после нахождения решений порождает инструментальную активность (T_x).

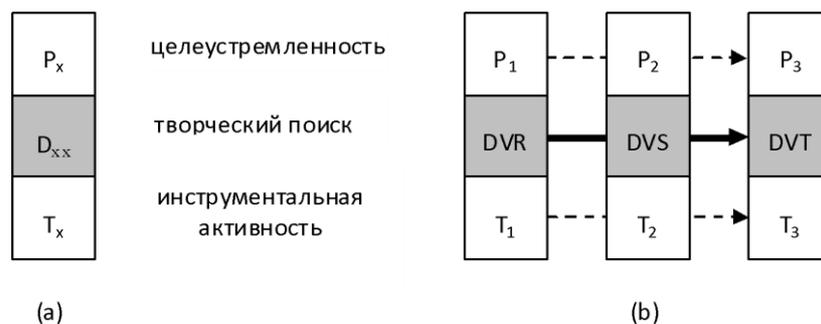


Рис. 3. Структура активности при решении задачи первого уровня (а) и пример построения последовательности задач на основе усложняющегося продукта дивергентных способностей на визуальном содержании (b).

Поиск принципа усложнения задач — это фактически поиск усложнения такого процесса в целом. В качестве примера на рис. 3b показан принцип построения последовательности задач, решение которых связано с получением более сложного дивергентного продукта (визуальных отношений, систем и трансформаций). Рис. 3b также демонстрирует принципиальную возможность строить последовательность задач на основе усложнения целевой P_x (P_1, P_2, P_3) и инструментальной составляющих T_x (T_1, T_2, T_3).

Сложность инструментальной активности можно увеличивать, например, за счет сложности манипуляций с инструментом (например, цепочка из пяти операций будет сложнее цепочки из двух операций). Ее также можно увеличивать за счет сложности самой информации, над которой в уме учащиеся совершают операции и которую, в конечном счете, они преобразуют с помощью инструментальных средств. Графические редакторы работают с визуальной информацией (V), текстовые редакторы и электронные таблицы — с символьной (S). При работе с командами операционной системы и языков программирования ведется обработка семантической информацией (M). На основании данной последовательности можно построить и ряд вводных курсов, в которых будет сделан акцент на развитие способностей DV_x , DS_x и DM_x .

Начальный уровень обучения информатике, развивающий дивергентные способности интеллекта, является важным этапом на пути дальнейшего профессионального развития, поскольку за счет ориентировочной деятельности он формирует целостную картину указанной области.

После прохождения вводных курсов учащиеся, как правило, уже потенциально могут специализироваться в каком-либо одном из направлений информатики, поскольку у них уже должен был сформироваться широкий кругозор. Кто-то, возможно, сразу приходит на второй уровень с уже сложившимися интересами в

этой области и с желанием дополнительно повысить свой уровень по сравнению со школьным образованием. Эти желания и интересы лежат в основе выбора курсов второго уровня.

Перечень курсов, который предлагается учащимся на втором уровне, определяется в основном двумя факторами. Этот перечень отображает, во-первых, основные направления компьютерных технологий и программирования, а во-вторых — представления школьников и их родителей о том, какие направления важны и интересны. На сегодняшний день в этот перечень включены компьютерная 2D и 3D графика и мультипликация, программирование и web-разработка.

Выбор направления — это, прежде всего, акт специализации, связанный с выбором для изучения некоторой предметной области вместе с ее объектами, характеризующимися конкретными структурными, функциональными и технологическими особенностями. Если на начальном уровне учащийся набирал разнообразный опыт, то на уровне основных курсов этот процесс должен смениться процессом упорядочивания опыта, приведения его в определенную систему в отношении некоторой предметной области. Для этого есть интеллектуально-возрастные предпосылки.

Рубежным возрастом перехода от вводных курсов к курсам второго уровня является 12 лет. Этот возраст выбран не случайно. Согласно социокультурной теории Л.С. Выготского, именно с этого возраста начинает формироваться системное мышление в понятиях. При решении задач учащиеся уже в состоянии самостоятельно определять новые понятия, на которые опирается их мыслительный процесс при решении задач.

Понятийное мышление позволяет учащимся перевести процессы решения задач на более высокий уровень осознанности и произвольности. Оно также помогает сосредоточить их внимание на общих принципах решения задач, абстрагироваться от конкретных инструментальных и языковых реализаций. Развитие данного типа мышления учащихся на основном уровне, ориентирующемся на предметную область, скорее всего, должно быть связано с развитием умения проектировать или с проектными компетенциями.

Рассмотрим более подробно структуру активности учащегося при решении учебной задачи, сутью которой будет разработка проекта (рис. 4). Эта структура генетически связана со структурой решения творческой задачи (рис. 3), и поэтому заметно их визуальное сходство. Однако есть и различия.

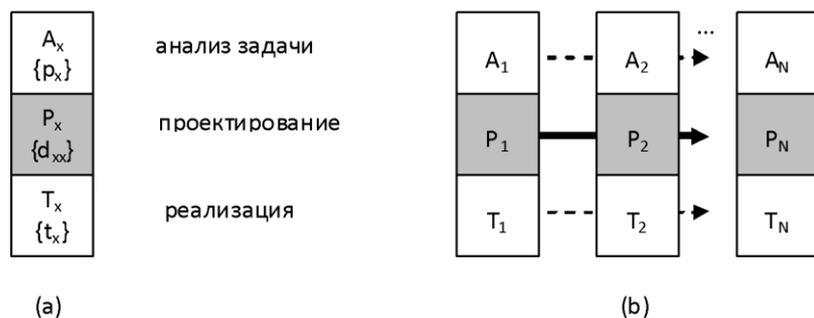


Рис. 4. Структура активности при решении учебной задачи курсов основного уровня (а) и схема построения последовательности усложняющихся задач (б).

Если процесс решения творческой задачи на вводных курсах принципиально может быть направлен на достижение слабо осознанной цели P_x , то проектная задача курса второго уровня должна уже предполагать постепенное ее осознание в процессе анализа требований к структуре и функциональным характеристикам предполагаемого результата (A_x). Иными словами, учащийся должен научиться осознавать не только планируемый результат своей работы, но и думать над тем, зачем этот результат нужен.

Например, работая над мультфильмом по заданной теме в курсе «Дизайн и мультипликация», желательно, чтобы учащийся задумался над такими вопросами, как: «зачем создавать мультфильм?», «какова его основная идея?», «кто его будет смотреть?», «сколько он будет длиться?», «как он будет восприниматься?», и т. п. Преподаватель ведет шаг за шагом учащегося к осознанию требований к будущей разработке и к связям указанных требований с этапами проектирования и реализации. В рамках выявленных ограничений может существовать и некоторое множество творческих целей $\{p_x\}$, касающихся, например, каких-то оригинальных качеств будущего результата.

Проект является формализацией цели разработки и формулируется в терминах предметной области. Дивергентные процессы $\{d_{xx}\}$, которые играли самую важную роль при решении творческих задач, здесь уже являются составной частью, которая обеспечивает поиск оригинальных решений в процессе проектирования (P_x). Например, творческий поиск выразительного образа персонажа.

В зависимости от курса предметами проектирования могут быть: визуальная форма в 2D или 3D графике; временная, динамическая структура в мультипликации; структура взаимодействия программных модулей и алгоритмы в программировании; информационная структура и алгоритмы взаимодействия с пользователем в web-проектировании. Конечно, школьники не делают настоящей проектной документации, но, тем не менее, их замыслы находят свое воплощение в эскизах, схемах, алгоритмах, раскадровках (рис. 5).

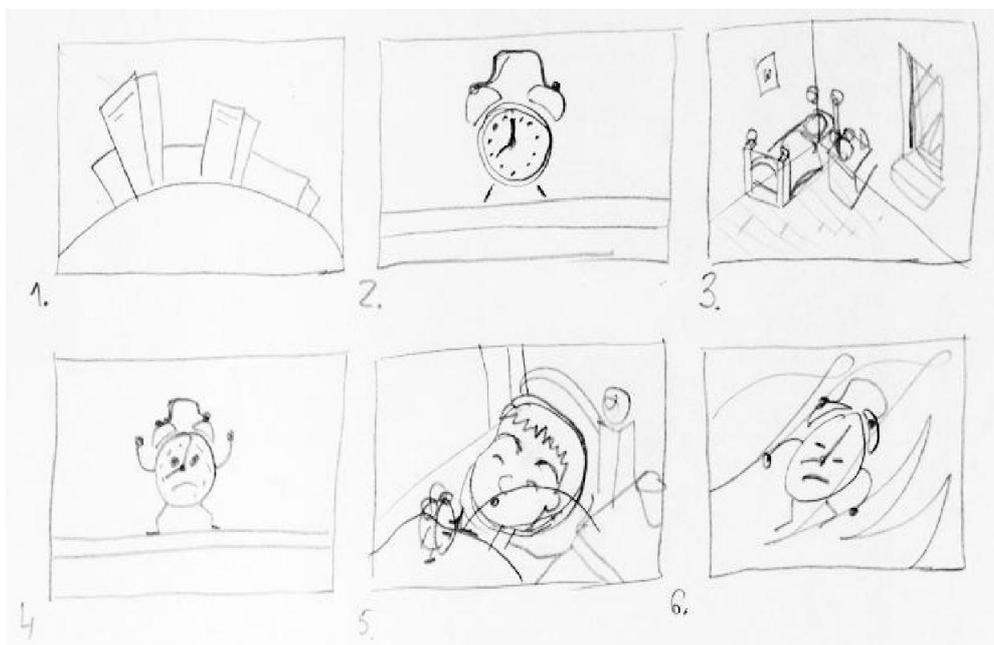


Рис. 5. Пример проекта мультфильма в виде раскадровки.

Усложнение цепочки учебных задач на данном уровне ориентируется, прежде всего, в сторону структуры проектируемого объекта (P_1, P_2, \dots, P_N), которая, в свою очередь, приводит к усложнению последовательности соответствующих технологических операций (T_1, T_2, \dots, T_N).

Технологический процесс описывается в обобщенных терминах предметной области, а не программно-инструментального средства. Тем самым он осознается инструментально-независимым. Можно сказать, что на втором уровне учащиеся как бы изучают обобщенную технологию. Это, в свою очередь, является предпосылкой понимания учащимися необходимости выбора наиболее подходящего инструмента и наиболее оптимальной технологической цепочки $\{t_x\}$, а также более легкого перехода от одного инструментального средства к другому.

Рассмотренные выше два уровня подготовки заложили основу, на которой теперь уже можно пытаться строить углубленную подготовку в области информатики, связанную с профессиональной специализацией. Особенно это актуально для учащихся, которые не удовлетворены уровнем школьной программы по информатике. Развитые дивергентные способности и проектные компетенции являются необходимыми условиями формирования профессиональных действий. Это легко показать.

Любое действие согласно теории Гальперина [5] состоит из ориентировочной и исполнительной части. За подготовку и правильность выполнения действия в

конкретных условиях отвечает ориентировочная часть. Это означает, что ориентировочная часть нужна для того, чтобы «приспособить» действие к разным условиям (временные ограничения, уровень инструментальных возможностей и пр.). Чем лучше приспособляемость действия, тем шире его ориентировочная основа. Поэтому именно на ее формирование должен быть направлен процесс обучения. Очевидно, что профессиональное действие характеризуется широкой ориентировочной основой.

Ориентировочная часть включает в себя четыре основных компонента:

- а) образ цели и связанные с этим образом мотивационные особенности;
- б) выделение значимых для будущего действия компонентов ситуации;
- в) планирование действия;
- г) регуляция действия на всех этапах его выполнения.

Заметим, что выполнение учебных задач первых двух уровней способствует развитию этих компонентов в направлении развития ориентировочной основы. Высокая мотивация действий и широкая возможность разнообразной произвольной регуляции (а, г) была свойственна творческим задачам, а выделение значимых компонентов и планирование (б, в) — проектным компетенциям. Кроме этого, обобщенное представление о технологическом процессе дает широкую возможность быстрой адаптации к новым инструментальным средствам, используемым для решения задач.

Таким образом, первые два уровня обучения неформальной системы в области информатики создают предпосылки для того, чтобы действия учащихся в процессе обучения могли приобрести все больше и больше качеств профессиональных действий. Иными словами, при переходе к третьему уровню за счет свободного выбора и специализации реализуется идея *профессиональной направленности*, которая тесно связана с освоением и реализацией эффективных технологических решений.

На углубленном (третьем) уровне учащийся должен не просто научиться что-то делать, а думать о том, как организовать процесс разработки, как это сделать быстрее и качественнее, какую технологию выбрать, как найти компромиссное решение при противоречивых условиях и т.п.; например, как грамотно вести разработку в ситуации, когда готовые технологические решения, которые используются для разработки сайтов или программ, позволяют сократить время разработки, но ухудшают ее качество.

Сложность учебных задач третьего уровня определяется уровнем требований к ним. Можно выделить по крайней мере три критерия такого рода сложности: 1) степень осознанности действий, 2) эффективность работы и 3) уровень требований к качеству конечного результата. В конце последовательности

усложняющихся задач должны быть задачи, практически не отличающиеся от профессиональных.

Поскольку школьники обычно не работают над коммерческими заказами, мотивация обучения на третьем уровне будет основана на участии в конкурсах, перспективе дальнейшего поступления в высшее учебное заведение или просто на реализации творческих амбиций, формировании портфолио.

Рассмотрим пример *практической реализации* трехуровневой неформальной образовательной системы в области информатики, которая была использована для организации занятий в ДЮКЦ Университета ИТМО. Чтобы эта система обеспечивала свободу выбора направления обучения, она построена на модульном принципе.

Модуль — это, прежде всего, независимая структура, и поэтому она характеризуется своей целью и задачами. В качестве такого модуля целесообразно рассмотреть учебный курс. Для компоновки с другими модулями в недельном расписании, а также с обучением в других организациях, занятия по каждому курсу проводятся один раз в неделю в течение 32-х учебных недель. Продолжительность занятия — 3 академических часа, следовательно, объем модульного курса составляет 96 часов.

Основной критерий при наборе учащихся — их личный интерес и самооценка уровня подготовки. В процессе обучения учащийся имеет возможность перейти на другой курс. Особенности процесса комплектования, ориентирующегося преимущественно на образовательные запросы учащихся, приводят к ситуации некоторого разброса внутри одной учебной группы как по возрасту, так и по уровню подготовки. Это удастся скомпенсировать за счет: 1) особенностей гибкого учебного плана, 2) индивидуальной работы с учащимися преподавателя и ассистентов-стажеров (обычно 1-2 человека), 3) индивидуально ориентированных учебных пособий.

На рисунке 6 приведена актуальная структура курсов ДЮКЦ (2020 год). В связи с быстрым развитием информатики эта структура неоднократно менялась и наверняка будет меняться в будущем по этой же причине. Наиболее существенные изменения происходят обычно в структуре третьего уровня, поскольку там находятся курсы, наиболее тесно связанные с актуальным уровнем технологического развития информатики.

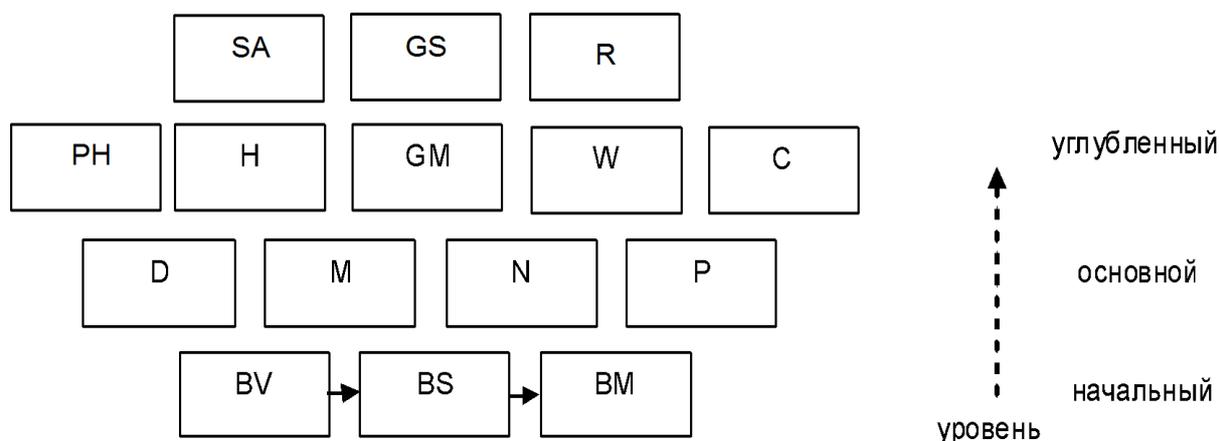


Рис. 6. Структура трехуровневой модульной системы.

Система ДЮКЦ включает следующие курсы на начальном уровне: BV — введение в компьютерную графику; BS — введение в офисные приложения и Интернет; BM — введение в программное и аппаратное обеспечение компьютера. На основном уровне: D — компьютерный дизайн и мультипликации; M — основы трехмерного моделирования; N — сети и Web-проектирование; P — основы программирования (Паскаль, 1-й год). На углубленном уровне: PH — студия фото- и видео-дизайна; H — проектирование и разработка 3D виртуальных сред; GM — разработка игр в 3D-пакетах; W — web-технологии; C — основы программирования (Си, 2-й год); SA — студия 2D-анимации; GS — студия разработки компьютерных игр; R — студия робототехники.

Как сама структура, так и программы вышеперечисленных курсов являются гибкими. Каждый курс включает базовый набор тем, который отражает основное содержание курса, и вариативную часть для адаптации программы под конкретный состав учебной группы. В первую очередь вариативная часть используется для компенсации возможного отставания группы от базового плана. Кроме этого, в зависимости от уровня курса могут появляться и специфические цели.

Так, на вводных курсах необходимо поддерживать разнообразные интересы учащихся к познанию нового, расширить их кругозор. В вариативную часть всех вводных курсов в том или ином объеме входят темы: 1) работа с операционной системой, 2) работа с файловой системой, 3) работа с архиваторами, 4) работа с офисными приложениями, 5) работа с приложениями 2D и 3D графики, 6) работа с Интернетом, 7) работа с командной строкой. В зависимости от конкретной ситуации эти темы содержательно могут меняться и могут быть направлены на повторение старого материала, закрепление навыков или расширения кругозора и

прочего. На начальном уровне, в связи с усложнением развивающих задач, которые решают учащиеся, желателен следующий порядок прохождения курсов: BV-BS-ВМ. На следующих уровнях порядок прохождения курсов не принципиален.

На основных курсах в вариативную часть входит знакомство с рядом инструментальных средств, из которых учащиеся должны научиться выбирать наиболее подходящие для проекта. Например, в курсе М, где основным инструментальным средством является 3DS MAX, альтернативными программами могут быть Blender или SketchUp.

На углубленных курсах вариативная часть может быть посвящена проблемам обеспечения функциональности разработки, ее эффективности и технологичности. Например, на курсе W в вариативную часть может входить изучение таких технологических решений, как Joomla или Drupal.

Оценку результативности предложенной системы в целом сложно дать, потому что каждый учащийся имеет свой собственный путь и свои цели обучения. Однако с помощью некоторых фактов и цифр можно продемонстрировать, что предложенная система удовлетворяет потребности школьников в дополнительном образовании по информатике, а также содействует их творческому и профессиональному самоопределению и развитию.

В системе ДЮКЦ учащимся предоставлена полная свобода не только в выборе курсов, но и в выборе последовательности курсов и продолжительности обучения. Распределение численности учащихся по уровням примерно равномерное (рис.7), и при увеличении или уменьшении количества учащихся в разные годы это распределение сохраняется в процентном выражении с незначительными отклонениями. Это говорит о том, что, по крайней мере, определенный контингент учащихся может удовлетворить свои образовательные потребности.

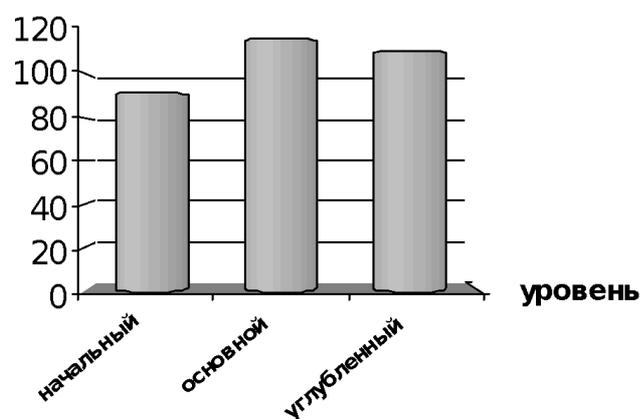


Рис.7. Распределение численности учащихся по уровням обучения (2017/2018 учебный год).

Несмотря на то, что учащиеся могут заниматься на нескольких курсах последовательно или параллельно, чаще всего они выбирают последовательное обучение (один курс — один год). Менее 6% ходят на два курса одновременно. Показательным является, что около 70% процентов учащихся выбирают какой-либо следующий курс (уменьшение образуется в основном из-за выпускников). На рис. 8 представлено распределение учащихся по количеству пройденных курсов. Это распределение говорит о том, что основная часть учащихся удовлетворяет свои образовательные потребности, пройдя 3 курса.

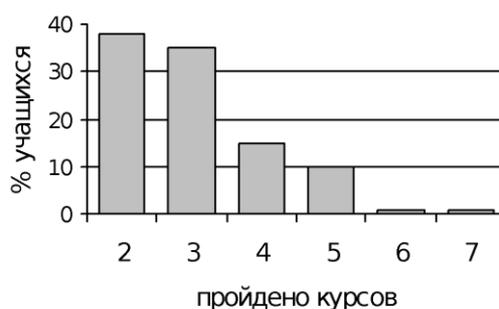


Рис.8. Распределение учащихся по количеству пройденных курсов.

Интерес представляет анализ последовательностей курсов, которые выбирают учащиеся. Они несут в себе информацию о той структуре знаний, которые пытались получить учащиеся, а также о процессе их профессиональной ориентации. Если всю совокупность этих последовательностей условно разделить на четыре направления:

1) графика (любые последовательности из курсов, включающие на втором уровне D или M, а на третьем H, PH, SA, GS, GM);

- 2) программирование (на втором уровне Р, а на третьем С);
- 3) web-проектирование (любые последовательности из курсов, включающие на втором уровне Р и/или N, а на третьем W);
- 4) разработка компьютерных 3D-игр (любые последовательности из курсов, включающие на втором уровне Р и/или M, а на третьем GM, GS);
- 5) не определившееся (включаются сочетания 1), 2), или 3)).

то получится следующая картина, описывающая специализацию учащихся:

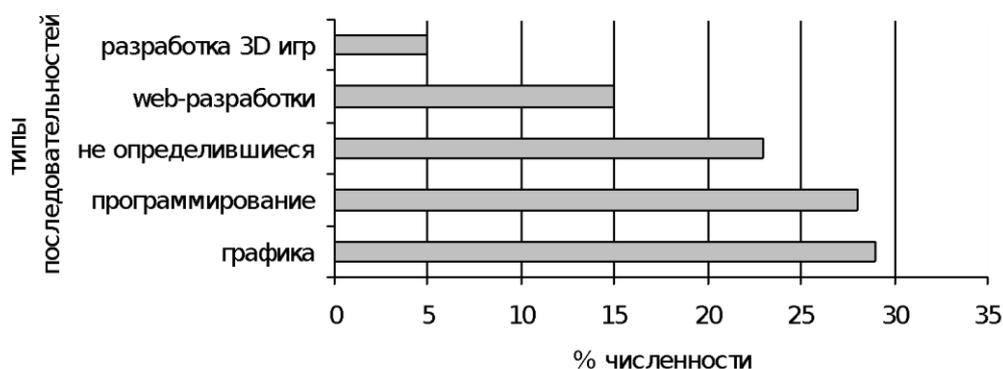


Рис 9. Специализация учащихся.

Рис. 9 показывает, что около 80% учащихся специализируются в каком-либо направлении, двигаясь вверх по уровням обучения. Распределение по направлениям подтверждает реализацию возможности свободного выбора. К учащимся, находящимся в поиске (около 20%), чаще всего относятся те, кто пытается действовать по стратегии широкого охвата материала. Они, как правило, начинают с первого уровня и проходят более трех курсов, расширяя свой кругозор в различных направлениях.

Творческое развитие, как уже было сказано, начинается с развития креативных способностей на первом уровне. Эффективность внедрения в учебный процесс развивающих заданий была доказана ранее [14]. Выход на профессионально-ориентированный уровень углубленных курсов стимулируется организацией и проведением творческих конкурсов по компьютерной графике и программированию, а также направленностью ряда курсов третьего уровня на подготовку к высшей школе.

Проведение выставок и конкурсов детского компьютерного творчества на базе ДЮКЦ началось в 1999 году. Тогда впервые был организован фестиваль для школьников Санкт-Петербурга «Компьютер и творчество», на котором проводилась выставка компьютерной 2D и 3D графики, конкурсы, показывались компьютерные анимации. С тех пор конкурс проводится регулярно. Опыт

участия в подобных выставках и конкурсах является важным элементом образовательного процесса.

Одна из традиций ДЮКЦ, сложившаяся за все время работы, — преемственность поколений. 70% преподавателей, ведущих занятия со школьниками в настоящее время, когда-то сами учились на этих же курсах.

Существование неформальной образовательной системы ДЮКЦ более 25 лет и ее востребованность в течение этого срока подтверждают жизнеспособность теоретических и практических подходов, приведенных выше. Рассмотренная структура организации компьютерных курсов представляет собой один из способов достижения компромисса между свободой и системностью в образовании. Теоретические принципы и практический опыт создания подобных систем может быть полезен при реализации аналогичных проектов

Методические особенности и проблемы обучения 3D-графике и анимации

Как известно, 3D-графика сегодня применяется в различных отраслях и сферах деятельности (мультимедиа, кино, видеоигры, медицина, веб-технологии, реклама и т.д.). Данное направление является востребованным и перспективным, однако грамотных специалистов в этой области не так много. Следовательно, подготовка таких специалистов является важной задачей для образовательного центра.

Процесс формирования знаний и умений в области трехмерной графики у школьников способствует развитию их творческих способностей, визуального мышления и может предопределить дальнейшую профессиональную деятельность.

Развитие визуально-пространственных способностей очень важно при обучении 3D-моделированию, потому что от этого зависят продуктивность и эффективность выполнения задач на пространственное мышление. *Основная проблема* существующих методик и пособий по обучению 3D-моделированию заключается в большем акценте на интерфейсах и возможностях инструментальных средств, чем на развитии визуального мышления.

Еще одна *проблема* обучения 3D-графике состоит в чрезмерной автоматизации множества пространственных преобразований над объектом. Как следствие, человек не включает интеллект в процессе моделирования, а лишь запоминает алгоритмы, которые ему предлагает инструментальное средство, и приучается выполнять манипуляции, анализируя форму модели только во внешнем плане.

Таким образом, в курсе 3D-графики процесс обучения 3D-моделированию и процесс развития визуального мышления неразрывно связаны. В программу входит изучение основ моделирования, общие методы решения задач, грамотные подходы к созданию формы моделируемого объекта. Для расширения визуального опыта слушателей была разработана система из последовательно усложняющихся задач:

1. Задачи на моделирование из геометрических примитивов.
2. Задачи на модифицирующие или дифференцирующие преобразования.
3. Задачи на моделирование сложных фигур и сцен.
4. Задачи на моделирование трехмерного объекта с помощью автоматических преобразований.
5. Задачи на разработку динамичных форм, связанных с физическими процессами.

Первый класс задач является самым простым, так как в него входят задачи, требующие разработки моделей или представления существующей формы из простых фигур (примитивов). Пример представлен на рисунке 10а.

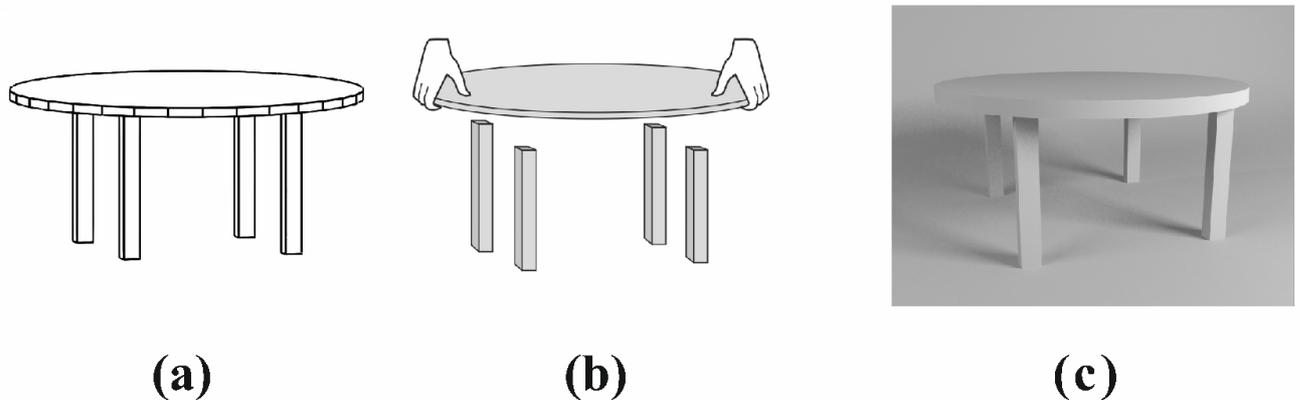


Рис. 10. Моделирование на основе геометрических примитивов (а), используется опыт формообразования из простых материальных объектов (б), визуализация готовой модели (с).

Анализ формы, необходимый для решения подобных задач, опирается на интериоризацию процессов перемещения и соединения объектов — рис 10б. Умение представлять изображаемый объект в виде простых форм или организовывать пространство с их помощью является базовым навыком для большинства видов деятельности, связанных с предметно-пространственными преобразованиями (дизайн, скульптура, архитектура).

В трехмерном компьютерном моделировании этот навык дает возможность максимально эффективно решать широкий класс задач, в которых можно использовать существующие модели как заготовки. В то же время, в технологическом плане на этом этапе отрабатываются важнейшие навыки работы в трехмерном редакторе (перемещение и ориентация объекта в пространстве, работа с видами, позиционирование, настройка параметров объектов).

Второму классу задач характерны задачи на основе модификаций «заготовок», из которых можно получить целевую форму — рис. 11а. На этапе анализа учащийся должен, воспользовавшись мысленным опытом, как бы увидеть способ получения будущей формы из заготовки (рис. 11б) — почти так же, как это делают скульпторы.

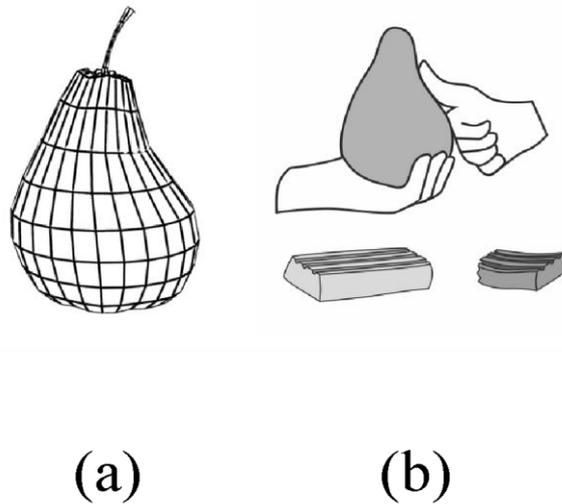


Рис. 11. Моделирование на основе модифицирующих преобразований (а), используется опыт деформации материальных объектов (b).

Основные подходы к выбору или созданию этой самой «заготовки» должны быть уже освоены еще на этапе моделирования из простых фигур. Так же, как из куска пластилина с помощью выдавливания, сплющивания, отсечения кусков и т. п. процедур можно сделать разнообразные фигуры, так и из 3D-заготовки, сделанной из специфического материала, называемого обычно «полигональными сетками», воздействуя на всю заготовку, либо на отдельные ее узлы, грани, ребра, можно создать новую модель.

Здесь учащийся впервые сталкивается со спецификой такого «материала», как полигональные сетки, что приводит к необходимости понимания особенностей работы с этим материалом (плоскостями, ребрами, вершинами) а, следовательно, и особенностей формообразования в отличие от реальных материалов глины или пластилина. Число полигонов как бы определяет степень гибкости и податливости этого материала и, следовательно, связано с возможностью повышения реалистичности формы. В то же время это число желательно минимизировать, поскольку от него зависит время обсчета компьютерной модели. Поэтому задачи, связанные с деформацией и модификаторами, уже дают повод для обсуждения вопросов, связанных с качеством моделирования, в частности, поиска компромисса между реалистичностью и количеством полигонов.

Третий класс задач специализируется на комбинации задач, принадлежащих двум предыдущим классам задач.

Они направлены на разработку более сложных моделей, состоящих из модифицированных объектов и готовых фигур, и(или) моделирование сцен

(рис. 12). Периодически возможен возврат то к первой, то ко второй группе задач, но уже на уровне применения новых технологических решений. Например, создание более сложных композиций с иерархической структурой или применение слоев (возврат к первой группе), или сложные модификации с использованием сплайновых сеток, неравномерных (multi-resolution) сеток и «скульптинга».

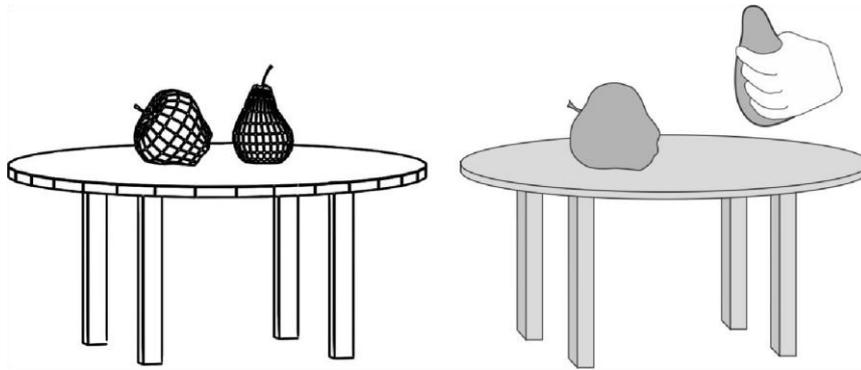


Рис. 12. Моделирование сцен включает комбинации задач.

Четвертый класс задач ориентирован на использовании тех инструментальных средств (3D-редакторов), в функционал которых входят автоматические преобразования формы (например, закручивание, тела вращения, составные объекты и др.). Данные автоматические преобразования позволяют получить новую форму из уже имеющейся (рис. 13а).

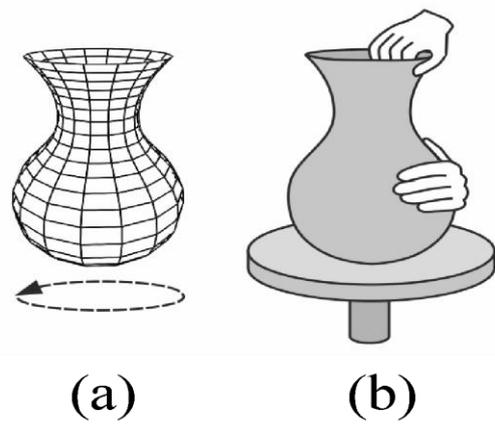


Рис. 13. Модель тела вращения в 3D-редакторе (а), аналогичная технология (б).

Результат образования новой формы просчитывается исходя из заданных параметров взаимодействия. Эта автоматизация наряду с упрощением рутинных процессов одновременно создает и некоторые сложности при моделировании. У учащегося уже нет той заготовки, в которой он мог бы попытаться увидеть будущую модель, как это было в задачах на деформацию. Чтобы понять, можно ли с помощью применения того или иного автоматизированного средства

получить нужную форму, ему нужно как бы проделать мысленно обратное преобразование к тем исходным формам, которые были вначале.

Пятый класс задач является специфическим классом задач на моделирование. Он содержит задачи, требующие создавать текучие, сыпучие, чувствительные к небольшим воздействиям формы. Такие формы имеют жидкости, дым, сыпучие тела, ткани, пламя и т.п. (рис. 14b). Эти формы существенно изменяются при воздействии различных сил: ветра, силы тяжести и др.

Процесс создания такой нестабильной формы полностью автоматизирован в трехмерном редакторе и зависит только от ряда исходных параметров. Здесь учащиеся почти не работают с формой. Они работают только с геометрией пространства, в котором расставляются направленные источники возмущений, а также объекты, на которые эти возмущения воздействуют (рис. 14a). Визуальное правдоподобие полученных моделей реальным явлениям, по всей видимости, является наиболее важным критерием качества модели, не считая, конечно, того, что здесь, как и в моделях на основе полигональных сеток, обратной стороной реалистичности выступает необходимость больших вычислительных затрат.

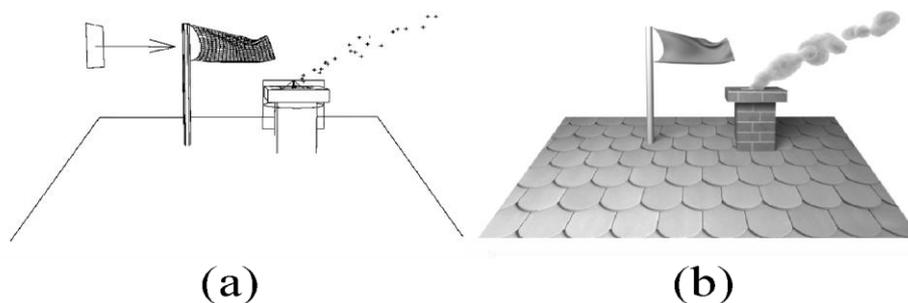


Рис. 14. Модель динамичных форм в 3D-редакторе (а), визуализация моделируемых объектов (b).

Решение задач данного класса дает визуальный опыт влияния параметров исходных объектов (систем частиц, источников возмущений и т.п.) на итоговую визуализацию. Опытный 3D-моделлер может заранее представить, как он должен расставить исходные объекты на сцене и какие задать параметры, чтобы получить желаемый эффект, и что будет, если увеличить или уменьшить тот или иной параметр.

Проектирование трехмерной модели является важнейшим компонентом ее разработки, поскольку оно обеспечивает эффективность способа ее изготовления, а также качество полученного результата. Поэтому *развитию и оценке проектных способностей следует уделять особое внимание при обучении основам трехмерного моделирования* [20]. Существующие в настоящее время методы оценки проектных способностей разработаны преимущественно для

подготовки специалистов в области промышленного дизайна и не могут быть напрямую перенесены на область трехмерного моделирования. Это связано не только с технологическими различиями в производстве промышленных и виртуальных 3d-объектов, но и с различными требованиями, предъявляемыми к конечным продуктам этих производств. Кроме того, 3D-проекты (если они разрабатываются) обычно выполняются в свободной манере без соблюдения каких-либо норм и правил выполнения проектной документации. Такая форма проектирования, к сожалению, не дает возможности оценить, насколько хорошо продумана геометрическая модель формы и насколько оптимален сам процесс моделирования. Но чаще всего процесс проектирования внешне вообще не выделяется разработчиками как отдельный этап. Это характерно для некоторых студентов, которые пропускают этот этап разработки, поскольку не понимают, зачем вообще он нужен, а также для профессионалов, у которых проектные действия полностью перенесены во внутренний план. Указанные обстоятельства приводят к необходимости разработки метода оценки проектных способностей, который бы позволял оценивать наличие и качество проектных действий учащихся, выполняемых в уме. Такой метод был разработан и предложен В.А. Локаловым и И.В. Климовым [18]. Теоретической основой, обеспечивающей конструктивную валидность предложенного метода, является теория поэтапного формирования умственных действий П.Я. Гальперина [5]. Следуя этой теории, проектные действия, можно классифицировать как ориентировочные действия, от которых зависят количественные и качественные параметры внешнего действия в инструментальной среде. На определенном этапе развития эти действия могут интериоризироваться, то есть перейти из внешней материальной формы во внутреннюю умственную форму. При этом проектные действия, выполняемые в уме, наследуют характеристики соответствующих внешних действий. Поэтому если учащиеся научились проектировать трехмерные модели в виде эскизов, которые обеспечивают правильность и оптимальность разработки конечного продукта, то их внутренние проектные действия будут в той же степени правильными и оптимальными. Очевидно, что в зависимости от степени развития проектных способностей будут меняться характеристики операционной структуры в инструментальной среде. Регистрация и анализ этих характеристик были положены в основу практической реализации предлагаемого метода. Эксперимент проводился в Детско-юношеском компьютерном центре Университета ИТМО на курсе «Основы трехмерного моделирования». В эксперименте участвовало 112 школьников, имеющих опыт выполнения проектов в виде эскизов, что позволяло непосредственно оценить их проектные способности. В качестве тестового задания была выбрана форма, проект которой учащиеся в состоянии были сделать в уме. Учащиеся должны были разработать трехмерную модель данной формы непосредственно в среде трехмерного моделирования без предварительного эскизирования, а после эксперимента

оценить как качество конечного продукта, так и операционную структуру выполненных действий. Те же параметры оценивались с помощью анализа полученных моделей и видеозаписи операций, выполненных в среде 3d редактора, а также итогового анкетирования учащихся. Полученные данные позволили получить численные значения множества параметров, которые согласно теории Гальперина характеризуют характеристики сформированных действий, а, следовательно, гипотетически должны быть связаны с уровнем развития проектных способностей. На основании оценок проектных способностей учащихся была построена регрессионная модель и произведена процедура минимизации количества факторов. В результате наиболее значимыми оказались следующие факторы:

- правильность выполнения действий — считается как отношение правильных действий к общему количеству действий с учетом коэффициента правильности, определяемого для каждого действия (в процентах);
- критичность — считается как отношение числа замеченных ошибок учащегося к общему числу допущенных им ошибок;
- оптимальность схемы действий — считается как отношение количества действий при оптимальном моделировании (постоянное число) к общему числу действий, выполненных для создания модели (в процентах).

Описанную модель оценки проектных способностей можно использовать в процессе подготовки специалистов в области 3D-графики.

Для курса «Основы трехмерного моделирования» характерны следующие **методические особенности**:

1. Развитие визуально-пространственных способностей через процесс интериоризации. Любая мыслительная операция представления или преобразования визуального образа изначально производилась с реальным, либо виртуальным визуальным объектом и была внешней. Поэтому, чтобы мыслительная операция переходила во внутренний план, необходимо ставить учащихся в такие условия, при которых они будут вынуждены использовать свой интеллект, прибегая к личному визуальному опыту и формируя на его основе новый опыт.

2. Разработанная система задач для развития визуального мышления, сложность которой определяется в соответствии с зоной ближайшего развития учащихся. Эта зона определяется разницей между тем, что учащийся знал к моменту обучения, то есть уровнем актуального развития, и тем, что он сможет достичь совместно с преподавателем.

3. Широкое применение технологий моделирования для решения большого спектра задач в 3D-графике (создание моделей любой сложности, построение композиции, создание анимации на основе созданных моделей).

4. Возможность комбинирования уже изученных технологий для решения поставленной задачи. По мере овладения той или иной технологией обычно переходят к изучению более сложных форм и задач, используя накопленный опыт, объединяя между собой несколько подходов к моделированию формы.

От грамотности применения перечисленных особенностей на практике зависит успех решения педагогических задач, выполнение слушателями любого класса задач в 3D-графике, гармоничное развитие визуально-пространственных способностей слушателей и личности в целом.

Методические особенности и проблемы обучения программированию

При обучении программированию педагог должен сосредоточить свое внимание на решении следующих основных, тесно связанных друг с другом, задач обучения:

1. Научить учащегося находить общее решение поставленной задачи.
2. Сформировать у учащегося необходимые понятия для решения задач.
3. Научить записывать общее решение задачи, используя формализм языка программирования.

При решении задач в школьной программе обычно внимание учащегося направлено на освоение методов, которые он затем применяет к однотипным задачам. При обучении программированию крайне важно объяснять учащимся, что решение задачи должно быть как можно более общим, что найденное решение должно решать не одну конкретную задачу, а класс схожих задач, и что методы, освоенные ими в школе, это только первая ступенька в поиске более общего решения.

$$ax^2 + bx + c = 0$$

a, b, c — произвольные
вещественные числа

Рис. 15. Квадратное уравнение и коэффициенты.

Например, в школьном курсе математики учащиеся решают десятки квадратных уравнений. Эти уравнения отличаются друг от друга только значениями коэффициентов (рис. 15). Решив несколько уравнений, учащийся запоминает формулу для нахождения корней уравнения и в дальнейшем ей пользуется. Но этой формулы оказывается недостаточно, для того чтобы учащийся мог написать программу нахождения корней квадратного уравнения. Почему? Дело в том, что прежде чем применять общую формулу, учащийся должен осознать, что значения коэффициентов могут быть любыми, в том числе и равными нулю. И если коэффициент при квадрате неизвестного равен нулю, то применить общеизвестную формулу нельзя (деление на ноль). Однако такое уравнение вполне может иметь одно или бесконечно много решений. Другими словами, необходимо провести анализ значений коэффициентов и только после этого искать решение тем или иным способом. К сожалению, в курсе школьной математики этот анализ не проводится.

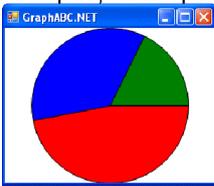
36 2	15 3	36 15	15 6	6 3
18 2	5 5	30 2	12 2	6 2
9 3	1 1	6	3	0
3 3				
1 1				

Рис. 16. Нахождение наибольшего общего делителя.

С другой стороны, метод решения задачи, который учащийся освоил в курсе школьной математики, может быть сложным и громоздким для реализации в виде программного кода. Например, нахождение наибольшего общего делителя двух натуральных чисел. Метод, который учащиеся используют в школе, заключается в выписывании последовательностей простых делителей сначала для одного, затем для других чисел. А затем вычеркиваются те делители, которые есть в одном числе и отсутствуют в другом. Произведение оставшихся после вычеркивания делителей и есть наибольший общий делитель. Конечно, можно написать программу, реализующую этот алгоритм. Но для этого требуется от учащегося умение работать с массивами (необходимо сохранять простые делители чисел, а затем вычеркивать), что затратно по памяти, и перебирать делители числа, что затратно по времени. При этом учащиеся даже не знают, что существует простой и очень эффективный алгоритм, позволяющий найти наибольший общий делитель двух натуральных чисел — алгоритм Евклида.

Даже если учащийся освоил какой-либо метод в курсе школьной программы, то это не означает, что он сможет его применить при решении задачи.

Напишите программу, решающую следующую задачу. С клавиатуры вводятся количество пятерок, количество четверок и количество троек, полученных учащимся в четверти. На экран необходимо вывести проценты пятерок, четверок и троек от общего количества оценок, полученных учащимся в четверти. Очевидно, что сумма процентов пятерок, четверок и троек должна быть равна 100 процентам.



Далее выведите в центре графического окна круговую диаграмму (см. рисунок). В этой диаграмме зеленый сектор будет соответствовать части троек от общего количества оценок, синий сектор — части четверок, а красный — пятерок.

Рис. 17. Пример задачи.

Для решения этой задачи (рис. 17) требуется составление пропорций — сначала для нахождения процентов тех или иных оценок от общего их числа, а затем для нахождения углов. И хотя учащиеся знают, как составлять пропорции, они испытывают серьезные затруднения при решении этой задачи.

Важным моментом, на который педагог должен обратить свое внимание, является условие учебной задачи. Прежде всего необходимо убедиться, что учащиеся верно и полностью понимают условие поставленной задачи, иначе они будут не в состоянии ее решить. Кроме того, важно, чтобы учащиеся понимали, зачем им надо решать сформулированную задачу — это дополнительно мотивирует их.

Например, учащимся 8 класса было предложено написать программу, решающую следующую задачу:

Дана квадратная матрица порядка M .
Обнулить элементы матрицы, лежащие ниже
главной диагонали.

Рис. 18. Пример задачи.

Условие этой задачи (рис. 18) учащемуся вряд ли понятно, так как ни что такое матрица, ни что такое ее порядок и диагональ, он не знает. Конечно, можно ему дать необходимые определения, но понимания, зачем матрицы нужны, у него от этого не появится. А значит, и не будет понимания, зачем ему эту задачу решать.

Учащимся можно предложить изначально другую задачу (рис. 19):

В верхнем правом углу стандартного графического окна, линиями рисуется квадратное игровое поле размером 10X10 ячеек. Компьютер, используя датчик случайных чисел, загадывает два числа в диапазоне от 1 до 10. Первое число — номер горизонтали игрового поля, второе — номер вертикали игрового поля. Таким образом, компьютер загадывает ячейку, которая располагается на пересечении горизонтали и вертикали. Пользователь с 10 попыток должен отыскать загаданную компьютером ячейку. Для этого он вводит с клавиатуры два целых числа номер по горизонтали и номер по вертикали ячейки. Если номера горизонталей и номера вертикалей не совпадают, то в ячейке, указанной пользователем, рисуется перечеркнутый квадрат. Если номера горизонталей или номера вертикалей совпадают, то в ячейке, указанной пользователем, рисуется синий квадрат. Если номера горизонталей и номера вертикалей совпадают, то в ячейке, указанной пользователем, рисуется красный квадрат и на этом программа завершается.

Рис. 19. Пример задачи.

Хотя в этой формулировке использование двумерных массивов не нужно, но требуется взаимодействие с двумерной структурой игрового поля (для того

чтобы отобразить очередной ход, необходимо перейти от номера ячейки (по горизонтали/вертикали) к координатам графического окна (x/y)).

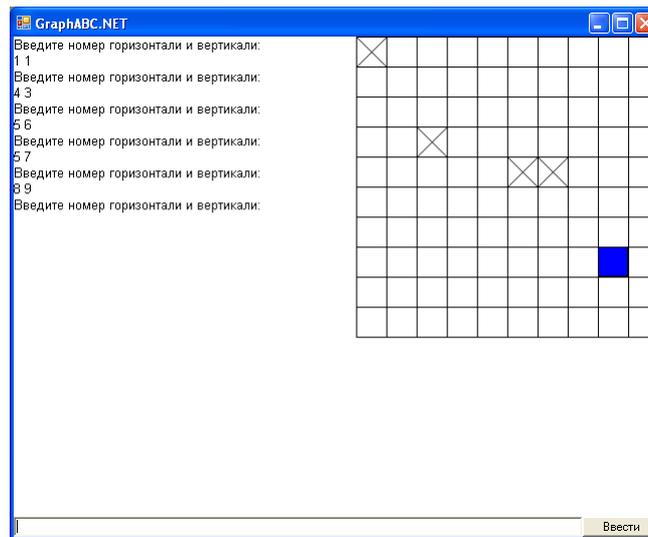


Рис. 20. Пример задачи.

Опыт решения этой задачи (рис. 20) будет полезен при программировании полноценного морского боя, где понадобится хранить положение кораблей в двумерном массиве и где без него уже не обойтись. Сделать игру более сложной и интересной может быть дополнительным стимулом для учащегося при написании программы.

Проблема, с которой учащиеся сталкиваются при решении задачи, это их неспособность объяснить, какие действия и в какой последовательности надо выполнять, чтобы задачу решить. Преподавателю с самого начала следует крайне внимательно относиться к тому, насколько полно и последовательно учащийся излагает решение задачи. Неполное и/или непоследовательное описание решения не только не позволит учащемуся записывать решение задачи на языке программирования, но и создаст трудности при освоении метода пошаговой детализации (метода, позволяющего эффективно бороться со сложностью задач).

В процессе решения задачи важно, чтобы учащиеся находили несколько различных решений (не для всех задач это возможно). Основными критериями оценки качества того или иного решения могут являться количество операций и объем памяти. Чем эти значения меньше, тем качество решения выше. Преподавателю важно научить учащихся оценивать качество решения, что позволит им осознанно выбирать наилучшее. Например, (рис. 21):

С клавиатуры вводится натуральное число n . Получить и вывести число, получаемое записью в обратном порядке цифр числа n . Например, $n = 1234$, выводится — 4321, $n = 1000$, выводится — 1.

Рис. 21. Пример задачи.

Чаще всего учащиеся предлагают записывать цифры числа в строку или символьный массив, а затем эту строку (массив) выводить. После того, как преподавателем указывается, что по условию задачи необходимо получить число, учащиеся предлагают преобразовать строку (массив) в число и затем его (число) вывести на печать. Эта задача может быть решена и без дополнительных строк или массива, с использованием позиционной системы счисления. Учащиеся знают, что такое позиционная система счисления, знают названия разрядов в десятичной системе, но воспользоваться своими знаниями не могут. Как выясняется в процессе обсуждения, для решения задачи необходимо находить остатки от деления числа или целого частного на 10 и, складывая их, умножать на 10, тем самым формируя новое число, в котором цифры записаны в обратном порядке. Очевидно, что при таком решении не надо хранить цифры в массиве или строке. А значит, второй алгоритм более предпочтителен, так как использует меньше оперативной памяти.

При обучении программированию преподаватель должен способствовать формированию у учащихся ряд понятий, необходимых для решения задач. При этом преподаватель должен начинать вводить новое понятие только тогда, когда без него решить поставленную задачу невозможно. Если решить задачу учащийся может без вводимого понятия, то он его не воспримет, так как не будет понимать его (понятия) необходимость.

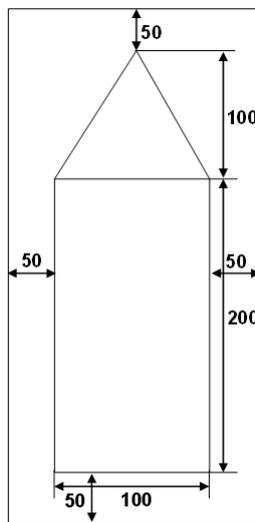


Рис. 22. Пример задачи.

Например, в графическом окне необходимо нарисовать домик (рис. 22). Размеры элементов, из которых должен состоять домик, и его положение в графическом окне показаны на рисунке. После того, как программа учащимися написана и домик заданных размеров отображен в графическом окне, учащимся предлагается увеличить домик и окно в полтора раза. Для этого придется переписывать программу, увеличивая все размеры в полтора раза. Далее учащимся предлагается уменьшить первоначальные домик и окно в два раза. Конечно, можно еще раз переписать программу. Но учащимся предлагается решить задачу в общем виде, выбрав некоторый размер в качестве основного, а все остальные размеры выразить через него. В результате получается новый рисунок, в котором основной размер — высота крыши (h), а все другие размеры выражены через него.

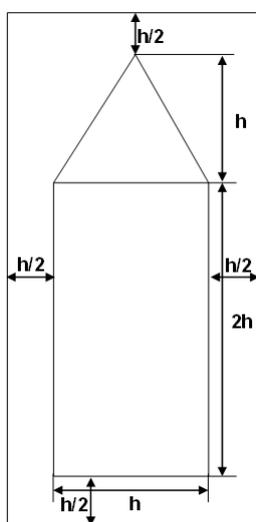


Рис. 23. Пример задачи.

Если теперь в программе задать значение для величины h (рис. 23), то переписывать программу больше не потребуется. Далее преподавателем вводится понятие «переменной». Очевидно, что без использования переменных, учащиеся обречены на многократное переписывание программы.

При этом педагог должен отдавать себе отчет, что понятия, вводимые им, формируются у учащихся далеко не сразу (проходя в своем развитии несколько стадий). Говорить о том, что-то или иное понятие сформировано, можно только тогда, когда учащийся им правильно пользуется при решении поставленных задач.

Следует порекомендовать начинать вводить понятия программирования на более абстрактном уровне, а лишь затем демонстрировать учащимся, каким образом это понятие находит свое воплощение в конкретном языке программирования. В то же время понимание учащимися необходимости использования того или иного понятия должно идти от учебной задачи.

С самого начала преподавателю необходимо объяснить учащимся, что для записи решения задач используется язык, который имеет свой синтаксис и семантику. И что мало знать, как синтаксически правильно строить высказывания в этом языке, надо, чтобы эти высказывания имели смысл. Если учащийся не может объяснить, как решать задачу на естественном языке, он не сможет записать ее решение на языке программирования.

Преподаватель должен уделять внимание вопросу исправления синтаксических ошибок учащимися в их программах, каждый раз фиксировать внимание учащегося на ней, пояснять причину этой синтаксической ошибки и способ ее устранения. Кроме того, особое внимание преподаватель должен уделить методу поиска и исправления логических ошибок в программах учащихся.

Преподавателю рекомендуется предлагать учащимся время от времени, особенно по ходу освоения конструкций языка программирования, записывать решение задачи (текст программы) не в интегрированной среде разработки, а на листе бумаги. Это позволяет преподавателю увидеть многие ошибки и понять причины их возникновения. А учащийся в этом случае вынужден внимательнее относиться к тому, что он пишет, и тем самым лучше овладевает языком программирования.

Дополнительно преподавателю рекомендуется прочитать книги по программированию [1, 2, 3, 8, 12].

Методические особенности и проблемы обучения по программе «Сети и Web-проектирование»

Сетевые технологии — это обширная и многосторонняя область технической информатики. Для формирования у школьников ясного представления о принципах устройства и функционирования современных сетей передачи данных нужно использовать чёткую терминологию. При этом следует понимать, что многие понятия, используемые в сетевых технологиях, тесно взаимосвязаны — это нужно учитывать, определяя порядок преподавания учебного материала. Для лучшего понимания изложенных ниже терминов рекомендуется ознакомиться со специализированной литературой по основам сетевых технологий, основам работы в сети Интернет [16].

Многолетний опыт проведения внешкольных курсов для детей «Введение в программное и аппаратное обеспечение персонального компьютера» и «Основы администрирования операционных систем» в ДЮКЦ позволил определить базовый набор терминов, необходимый для знакомства школьников с темой «Сети передачи данных», и удобный для усвоения порядок введения этих терминов по ходу урока. Предлагаемая методика отвечает принципу обучения в «зоне ближайшего развития», то есть на том уровне, который может быть достигнут ребёнком под руководством учителя. Предполагается, что учащиеся уже знакомы с такими понятиями, как «компьютер», «информация», «данные».

Опыт преподавания курса показывает, что лучше не вводить более 2-3 определений новых для школьников терминов за один урок. Под новыми терминами имеются в виду слова и словосочетания, с которыми большинство учащихся рассматриваемого возраста почти не встречались, в отличие от тех слов и словосочетаний, которые «на слуху» у современных школьников. Например, проводимые опросы показали, что почти каждый школьник знаком с терминами «компьютерная сеть», «wi-fi», «интернет», «локальная сеть», «IP-адрес», «сайт», «сервер» и способен либо самостоятельно, либо с помощью подсказок преподавателя дать определения этим понятиям. Другими словами, определения этих терминов располагаются в зоне ближайшего развития учащихся, и, при необходимости, все эти определения можно рассмотреть на одном занятии. Новыми же для школьников терминами, не имеющими широкого распространения среди простых обывателей, но при этом важными в сфере сетевых технологий, являются: «хост», «протокол», «сетевой порт», «клиент», «маска подсети», «шлюз», «DNS» и др. Чтобы не перегружать учеников большим количеством определений новых неизвестных слов, лучше вводить эти термины на отдельных занятиях, например, посвятить отдельный урок теме «Сетевые протоколы», отдельный для темы «Клиент-серверное взаимодействие» и т.д.

Обязательно следует постоянно перемежать теорию с практикой и активно использовать визуальные образы и примеры из жизни при объяснении понятий. Например, рассматривая локальные и глобальные сети, необходимо показать разницу между ними визуально, попросить детей привести примеры локальных и глобальных сетей.

Порядок введения терминов должен учитывать, что определения одних понятий могут опираться на другие понятия. В первую очередь объясняется значение термина «компьютерная сеть». Это словосочетание широко распространено в современном обществе и для большинства школьников является общеупотребительным, поэтому ребятам для начала предлагается попробовать самостоятельно объяснить, что это такое. Обсуждение ответов учащихся часто сводится к тому, что «компьютерная сеть — это несколько связанных друг с другом компьютеров...» — и в этот момент полезно обратить внимание на суть и назначение этой связи. Например, спросить: «Если связать несколько компьютеров верёвками, то это будут связанные друг с другом компьютеры, но можно ли назвать это компьютерной сетью?» В результате учащиеся либо самостоятельно, либо с помощью преподавателя приходят к заключению, что компьютерная сеть — это несколько компьютеров, объединённых друг с другом с целью передачи данных между ними.

Далее полезно будет обратить внимание учеников на то, что термин «компьютерная сеть» часто употребляют по отношению к более общему понятию — сеть передачи данных. Ведь обмениваться данными друг с другом могут не только компьютеры (в их общем понимании), но и другие устройства: мобильные телефоны, принтеры и т.д. Таким образом, понятие «компьютерная сеть» расширяется до понятия «сеть передачи данных» - это несколько устройств, объединённых друг с другом для обмена информацией.

Разобрав различия между локальными и глобальными сетями, можно перейти к рассмотрению сред передачи данных, либо к вопросам адресации в сетях. Если предполагается сделать обзор сетевого оборудования, то делать это следует только после рассмотрения сред передачи данных (ведь оборудование различается по используемым средам) и принципов адресации (в противном случае учащиеся просто не смогут понять разницу между сетевым коммутатором и маршрутизатором). Много лет назад мы также знакомили учащихся с таким устройством, как сетевой концентратор (хаб), объясняя его отличия от сетевого коммутатора (свитча), но в последние годы отказались от этого в связи с тем, что сетевые концентраторы морально устарели и практически нигде не используются.

Знакомство с принципами адресации в сети следует начать с рассмотрения понятия «IP-адрес». Рекомендуется сразу же перейти к практике, рассмотрев на компьютерах команды операционной системы для определения собственного IP-

адреса (`ipconfig` в ОС Windows, `ifconfig` в GNU/Linux) и команду проверки доступности других сетевых устройств по их адресам (`ping`). Обратите внимание, что `ping` локальных компьютеров не будет работать, если в настройках брандмауэра этих компьютеров не разрешён запрос входящего эха (в настройках фильтрации протокола ICMP).

Далее своевременным будет ввести понятие «доменное имя» и кратко рассказать о службе DNS, не вдаваясь в подробности её функционирования, а лишь донести до учеников её основную суть: преобразование удобных для человека символьных адресов наподобие `google.com` в удобные для компьютера, но неудобные для человека числовые адреса, наподобие `213.158.11.46`. Подкрепить это на практике, показав с помощью команды `ping`, что любому доменному имени соответствует некоторый IP-адрес (пусть учащиеся таким образом попробуют узнать числовые адреса своих любимых сайтов). Если занятие проходит на компьютерах в локальной сети, где каждый компьютер имеет своё локальное доменное имя, то также пропинговать и эти имена, продемонстрировав совпадение с IP-адресами, увиденными ранее с помощью команды `ipconfig`.

Полезным будет на этом этапе ввести понятие «хост» как устройство, имеющее свой адрес в сети. При этом в качестве хоста может рассматриваться как компьютер, так и отдельные его компоненты, через которые компьютер подключен к сети. Например, в компьютере может быть несколько сетевых устройств: WiFi-адаптер, LAN-адаптер, USB-модем и т. д. — в этом случае у каждого из этих устройств будет свой адрес, так что это будет несколько хостов.

Рассматривать структуру IP-адреса (его двоичное представление, сравнение IPv4 и IPv6) имеет смысл только в том случае, если далее предполагается изучать маршрутизацию в компьютерных сетях. То же самое касается и рассмотрения масок подсети — вообще говоря, для рассмотрения маски подсети сперва нужно объяснить, что такое «подсеть». Несмотря на наличие вопросов по структуре IP-адреса и маски подсети в едином государственном экзамене по информатике и ИКТ, необходимость этих знаний рядовому пользователю представляется сомнительной. Тем не менее, эти знания необходимы для ручной настройки компьютерных сетей. Если учащиеся недостаточно уверенно чувствуют себя при работе с двоичной системой счисления, то можно рассмотреть маски подсети только трёх видов: `255.0.0.0`, `255.255.0.0` и `255.255.255.0` (исторически они известны как класс А, класс В и класс С). Прочие маски подсетей используются реже, и обычно нет необходимости с ними сталкиваться при настройке локальных сетей. В этом случае можно сказать, что два IP-адреса будут относиться к одной подсети в том случае, если совпадают части IP-адреса, стоящие напротив чисел 255 в маске подсети. Эти части IP-адреса будут называться адресом подсети, а расположенные правее — адресом хоста. Для корректной работы нескольких компьютеров в одной локальной сети их маски

подсети должны совпадать, адреса подсетей также должны совпадать, а адреса хостов — различаться.

Для возможности осуществления ручных настроек в параметрах сети на практике необходимо использовать виртуализированный компьютер, так как на реальном компьютере из соображений безопасности сетевые настройки должны быть доступны для изменения только администратором компьютерного класса. Существуют различные программы для создания виртуальных машин: VMWare, Parallels Desktop, VirtualPC, Hyper-V, Bochs, KVM, Oracle VM VirtualBox. Мы рекомендуем использовать на учебных занятиях программу Oracle VirtualBox, т.к. она является бесплатной, кроссплатформенной, имеет русскоязычный интерфейс и при необходимости легко может быть установлена учениками на своих домашних компьютерах. В программе VirtualBox можно создать любое количество виртуальных машин и запускать их по очереди, либо одновременно (если хватает ресурсов физического компьютера). Для работы на виртуальной машине необходимо установить на неё операционную систему (это может делаться так же, как и на реальном компьютере, а можно и проще: если дистрибутив операционной системы загружен из интернета, то не обязательно записывать его на компакт-диск или загрузочную флешку — можно установить систему прямо из файла загрузочного образа).

В программе VirtualBox поддерживаются четыре разных способа подключения виртуализированного компьютера к сети: «NAT», «Сетевой мост», «Внутренняя сеть» и «Виртуальный адаптер хоста». По умолчанию используется подключение в режиме «NAT», так как оно успешно работает, независимо от способа подключения к сети физической машины, и обеспечивает виртуальной машине доступ в интернет при наличии этого доступа у физической машины. При подключении через «NAT» виртуальная машина оказывается в отдельной виртуализированной локальной сети, подключенной к реальной сети через созданный на физической машине виртуальный маршрутизатор. Несмотря на наличие доступа с виртуальной машины к реальной локальной сети и сети Интернет, доступ из реальной локальной сети к виртуальной машине через NAT невозможен (если не произвести специальные настройки).

По этой причине для начального знакомства с сетями на виртуальной машине рекомендуется использовать подключение через «Сетевой мост», так как оно является более простым с точки зрения сетевой организации и адресации. В этом случае виртуальная машина подключается напрямую к реальной локальной сети, получая IP-адрес наравне с реальными компьютерами. Это позволяет полноценно работать с сетью как при использовании реального компьютера. Но проблема при использовании «Сетевого моста» заключается в том, что для работы виртуальной машины требуется свободный IP-адрес в локальной сети. При использовании большого количества реальных и виртуальных компьютеров в одной сети может

возникнуть нехватка IP-адресов — вероятность возникновения этой проблемы необходимо обсудить заранее с администратором локальной сети и принять необходимые меры. Также нужно быть осторожным при ручном изменении настроек в параметрах сети на виртуальных машинах: если указанный вручную IP-адрес совпадёт с адресом одного из реальных компьютеров в сети, то это может привести к серьёзным неполадкам в работе сети. Поэтому нужно заранее запросить у администратора информацию о свободных IP-адресах, которые могут быть использованы в учебных целях, и внимательно следить, чтобы ученики правильно производили настройки.

Если нет возможности использовать подключение типа «Сетевой мост» (из-за отсутствия доступных IP-адресов в реальной сети или при отсутствии сетевого адаптера), то можно рассмотреть сетевое взаимодействие виртуальных компьютеров с помощью подключений типа «Внутренняя сеть» или «Виртуальный адаптер хоста» и создав несколько виртуальных машин на одном физическом компьютере. Оба эти способа подключения создают виртуализированную сеть, но, в отличие от режима «NAT», позволяют виртуальным машинам, запущенным на одной и той же физической машине, общаться друг с другом. Отличие «Виртуального адаптера хоста» от «Внутренней сети» заключается в том, что физическая машина в первом случае также подключается к виртуализированной сети и может взаимодействовать с виртуальными машинами (но не через свой физический сетевой адаптер, а через виртуальный, имеющий отдельный сетевой адрес). Ну а «Внутренняя сеть» является полностью изолированной не только от реальной сети, но и от физической машины, на которой она создана.

Используя разные способы подключения и разные ручные настройки сетевых параметров на виртуальной машине, можно имитировать различные сетевые неполадки: отсутствие доступа к шлюзу, к DNS-серверу, отсутствие подключения шлюза к Интернету и др. Необходимо рассказать и показать на практике, как с помощью команд `ipconfig`, `ping` и `tracert` диагностировать подобные неисправности, чтобы обнаруживать причину неполадок. Не забывайте, что пинговать можно только те компьютеры, в брандмауэрах которых разрешён запрос входящего эха в настройках протокола ICMP.

Для знакомства с различными сетевыми протоколами и формирования представления о стеке протоколов рекомендуется использовать на практике анализатор трафика (сниффер) наподобие WireShark. При этом рекомендуется сначала использовать подключение виртуальной машины к сети в режиме «Внутренняя сеть», чтобы изолировать трафик от внешних воздействий и рассмотреть интерфейс программы и представление заголовков перехватываемых сетевых протоколов. Затем переключить виртуальные машины в режим «Сетевой

мост» (или в режим NAT, если подключение в режиме «Сетевой мост» технически невозможно) и рассмотреть перехват внешних протоколов.

Клиент-серверное взаимодействие может быть рассмотрено с помощью всё того же анализатора трафика WireShark как на примере простых протоколов типа DNS, так и на более высокоуровневых, в частности, HTTP. Если в программе курса имеется достаточно часов на изучение этой темы, то для лучшего представления о сути сервера рекомендуется на практике установить веб-сервер на виртуальной машине (наиболее логичным представляется установка Apache HTTPD) и также отслеживать запросы к нему и ответы с помощью сниффера. При подключении через «Сетевой мост» доступ к веб-серверу будет работать с физических машин других учеников (при этом нужно не забыть добавить протокол HTTP в список исключений брандмауэра). Также наличие работающего веб-сервера на виртуальной машине является хорошей возможностью продемонстрировать на практике принцип работы технологии PAT (проброс портов). В этом случае виртуальная машина подключается к физической через NAT, и в настройках NAT настраивается проброс 80 порта TCP на 80 порт виртуальной машины. После этого веб-сервер виртуальной машины будет доступен с других физических компьютеров при обращении с них на физическую машину с настроенным NAT. Также можно изменить порт хоста (оставив неизменным порт гостя) и показать, что теперь веб-сервер доступен через нестандартный порт. Для большей наглядности и осмысленности — это можно сделать, создав веб-сервера на двух разных виртуальных машинах, установленных на одной физической машине, и обращаться к этим серверам по одному IP-адресу, но через разные порты.

Для знакомства с протоколом FTP рекомендуется сначала показать учащимся работу с FTP-клиентами. Для этого необходимо иметь доступ к какому-либо FTP-серверу. Поскольку одно из важных преимуществ FTP перед HTTP — это возможность аутентификации пользователей, то лучше помимо анонимного доступа продемонстрировать школьникам доступ на FTP с логином и паролем. Для этого лучше всего заранее установить и настроить собственный FTP-сервер (например, на виртуальной машине) и создать на нём логины и папки для каждого пользователя. После практического знакомства с FTP-протоколом на клиенте можно дать каждому ученику установить FTP-сервер на виртуальной машине и рассмотреть с ними основные серверные настройки и процесс создания пользователей. Мы использовали для этих целей FileZilla Server. Подключение виртуальной машины в режиме «Сетевой мост» или «Виртуальный адаптер хоста» позволяет осуществлять прямое соединение клиента на физической машине с FTP-сервером на виртуальной машине. Затем, переключив виртуальную машину в режим «NAT», можно показать необходимость использования настройки Passive Mode в параметрах FTP-сервера. При настройке Passive Mode на сервере указывается диапазон портов для подключения клиента.

Рекомендуется в учебных целях указать небольшое количество портов (два-три) и добавить их все в исключения брандмауэра на виртуальной машине. После этого указанные порты пробрасываются с физической на виртуальную машину в настройках NAT. Анализатор трафика Wireshark позволит увидеть разницу в запросах и ответах при использовании Passive Mode и при обычном режиме.

Наконец, для любого современного пользователя будет полезным умение настраивать домашние маршрутизаторы и точки доступа WiFi. При рассмотрении особенностей изучения настроек маршрутизатора следует обратить внимание на то, что в разных моделях маршрутизаторов интерфейс настроек может сильно отличаться, поэтому основной упор при обучении школьников нужно делать на объяснение сути настроек, и стараться не привязываться к конкретному интерфейсу. Рекомендуется рассмотреть несколько разных интерфейсов настройки маршрутизаторов. Рассматривая настройку WiFi, можно попробовать показать зависимость силы принимаемого сигнала от выбранного канала.

В курсе «Основы администрирования операционных систем» нами также рассматривается установка Active Directory на Windows Server и создание контроллера домена. При этом Windows Server работает на виртуальной машине, и мы успешно подключаем к развёрнутому на нём домену операционные системы, запущенные на других виртуальных машинах, в том числе на других физических компьютерах. Таким образом, использование Virtual Box позволяет изучить на практике все особенности сетевого администрирования, в том числе работу с доменами Windows.

В заключение еще раз отметим, что особенность дополнительной образовательной программы «Сети и Web-проектирование» заключается, с одной стороны, в необходимости познакомить учащихся с аппаратной и программной частью компьютерных сетей: назначением и управлением сетевым оборудованием, установкой и настройкой специализированного программного обеспечения для работы с сетевыми ресурсами; с другой стороны — с проектированием и разработкой web-сайтов; также есть набор понятий, без владения которыми невозможно разобраться, как работает компьютерная сеть: протокол, сервер, клиент, сообщение, ip-адрес, DNS, URL и т.п.

При обучении аппаратной части компьютерных сетей и объяснении основных принципов работы компьютерных сетей важно оценить актуальный уровень развития учащихся, который, как правило, не включает владение такими понятиями, как сигнал, источник, приемник, преобразование сигналов, архитектура и т.п., и построить изложение учебного материала таким образом, чтобы сначала познакомить учащихся с этими понятиями. Также важно выстроить последовательность практических заданий, в рамках которых учащиеся будут иметь возможность «пощупать» и поработать с каждым изучаемым сетевым оборудованием.

При обучении проектированию и разработке Web-сайтов важно показать учащимся необходимость каждого из этапов работы: от сбора информации и определения цели сайта до размещения сайта в сети Интернет и тестирования.

При подготовке и проведения занятий по программе «Сети и Web-проектирование» педагогу необходимо учесть следующие особенности:

- Не все учащиеся понимают важность планирования своей работы на компьютере и необходимость этапа проектирования.
- Знакомство в рамках программы с языком разметки гипертекста — это, как правило, для учащихся первый опыт логической и визуальной разметки документов, что несет некоторые трудности при переходе к мышлению на новом уровне абстракции.
- В последнее время всё чаще учащиеся не умеют работать с электронной почтой и с файловой системой, а также медленно печатают текст на компьютере.

Часто встречающиеся ошибки при проведении занятий и рекомендации по их устранению

В заключении укажем и кратко прокомментируем ряд наиболее типичных ошибок, которые наиболее характерны для начинающих преподавателей. Наряду с недостатком опыта, причиной данных ошибок может быть поверхностная теоретическая подготовка в области педагогической психологии.

Учебная задача не соответствует программе курса

Программа курса содержит основные положения, которые должны связывать задачи этого курса с комплексом задач всей системы курсов, а также с общей концепцией развития дополнительного образования, принятой на государственном уровне. В рассмотренной трехуровневой модели компьютерных курсов, например, одной из задач вводных курсов является формирование представлений учащихся о характере деятельности, свойственном тому или иному направлению компьютерных технологий (второй уровень), а также развитие дивергентных операций инструментального плана.

На курсах же третьего уровня вместо того, чтобы вбивать учащимся в голову какие-то общепринятые профессиональные требования к качеству, следует заложить основы критического мышления, которые дали бы им понять, откуда взялись эти представления, а также в дальнейшем позволили бы выйти им на путь профессионального саморазвития.

Неправильно сформулировано или неправильно выдано учебное задание

Как уже было сказано, учебная задача (учебное задание) является основным содержательным элементом планирования занятия. Она определяет *цель учебного действия*. Решение задачи — выполнение учебного действия. Проблемы с дисциплиной на компьютерных занятиях бывают часто связаны не с отсутствием внутренней мотивации, а с неграмотной формулировкой или неграмотной выдачей учебного задания.

В.А Локалов предложил рассматривать следующую совокупность критериев качества учебных заданий (задач):

1. **Актуальность** (освоение или тренировка именно того, что сейчас чрезвычайно необходимо).
2. **Простота** (задача не должна требовать дополнительной умственной или физической работы, не относящейся к содержанию обучения по теме).
3. **Полнота** (задача должна тренировать все умения, которые соответствуют содержанию обучения по теме).

4. **Интерес** (внутренняя и имеющая положительную эмоциональную окраску мотивация).
5. **Осмысленность** (в контексте будущей деятельности, учащийся должен понимать, что решение подобных задач ему в жизни пригодится).
6. **Выполнимость** (за урок).
7. **Соответствие зоне ближайшего развития.**

Поясним эти критерии на примере. Предположим, на занятии с темой «Введение в html» преподаватель следующим образом сформулировал задачу:

«Давайте создадим html-страницу, на которой будет: 1) какое-нибудь слово — заголовок первого уровня; 2) какое-нибудь число — заголовок второго уровня и ниже какая-нибудь любимая картинка, которую вы скачаете из Интернета с подписью».

Очевидно, что это задание не отвечает по крайней мере первым пяти критериям. С уверенностью можно сказать, что выполнение задания, выданного таким образом, *не будет актуальным* для учащихся, если им не объяснят, например, что сейчас необходимо выполнить задание потому, что для выполнения следующего задания им потребуется размечать заголовки и вставлять картинки. Задание *требует дополнительной интеллектуальной работы*, поскольку слова «какое-нибудь» и «любимую» потребуют дополнительных размышлений. Следовательно, задание не удовлетворяет критерию «простота».

Следует опасаться заданий типа «что больше любите», «что умеете» «какой-нибудь», «придумайте», если это не специальные задания на развитие креативности.

Критерий «полнота» не соблюден, потому, что «введение» в инструментальное средство или знакомство с языком предполагает как минимум *показ возможностей* этого средства (языка). Задание, в котором непонятно, зачем нужно делать страницу с каким-то словом, цифрой и картинкой, *вряд ли будет интересным*, поскольку совершенно непонятно, зачем это делать, за исключением случая, когда школьник по каким-то совершенно не зависящим от преподавателя причинам вдруг решил, что html ему необходим. Но даже и такой мотивированный ученик не сможет по приведенному в примере заданию разобраться, для решения каких задач в дальнейшем он сможет использовать полученные знания.

Следующие два свойства учебных задач можно использовать для оценки качества формулировки учебных и профессиональных задач в методической документации. Их мы назвали «*методическая определенность*» и «*методическая реализуемость*». Первое свойство показывает, что профессиональная задача

достаточно *однозначно сформулирована*, а второе — то, что можно *научить решать* эту задачу, то есть разработать соответствующую методику.

Отметим, что чем более абстрактно сформулирована задача, тем более сомнительна ее полноценная методическая реализуемость. В самом деле, если мы говорим о том, что «учим решать дизайнерские задачи», то мы претендуем на то, чтобы научить решать *любую* задачу в *любой* области дизайна, что, в общем случае, конечно, невыполнимо. Поэтому для более реалистичного планирования необходимо перейти на более низкий уровень абстракции формулировки.

Выбор уровня абстракции для задачи по критериям «методическая определенность» и «методическая реализуемость» проиллюстрируем на следующем примере. Предположим, что мы хотим, чтобы учащийся смог решить задачу «*Изобразить геометрическую фигуру*». Решению такой задачи может соответствовать и нарисованный на компьютере красный треугольник, и черный квадрат Малевича. Человек, умеющий «изобразить фигуру», умеет изобразить *любую* фигуру в *любой* технике, в отличие от человека, умеющего решать задачу «нарисовать квадрат на компьютере в редакторе Paint». В то же самое время научить изображать *любую* фигуру в *любой* технике невозможно, поскольку в этом случае множество фигур и множество техник могут содержать сколь угодно большое число заранее неизвестных неопределенных элементов. Поэтому задача «изобразить фигуру» является *методически неопределенной*.

Итак, задача является методически неопределенной, если нельзя однозначно указать множества:

- желаемых результатов (включая *критерии качества* выполненного задания);
- способов решения задачи;
- условий выполнения задачи (в т. ч. разнообразные ограничения на разработку).

Поэтому задача «Нарисовать зеленый круг [результат] на компьютере [условие] в графическом редакторе Paint [способ] за 5 минут [условие]» будет методически определенной задачей.

Очевидно, что не всякая методически определенная задача может быть методически реализуемой. Например, задача «нарисовать анатомически правильную фигуру человека [результат] в графическом редакторе Paint, используя только стандартные графические примитивы [способ] за 5 минут [условие]» вряд ли может быть решена даже при обучении хорошо подготовленных рисовальщиков.

Отметим, что *объяснение решения* задачи должно вестись на *более высоком уровне абстракции*, чем формулировка задачи. Этим обеспечивается развитие

умения решать целый класс задач. Путь к развитию подобного умения лежит через методику, в которой решается обычная конкретная задача, но комментарии и объяснение этой задачи даются с помощью обобщенных терминов.

Например, учащийся должен освоить «Рисование в графическом редакторе *геометрического примитива заданного цвета и размера в заданных координатах*» (это класс задач), а соответствующая ей методически определенная задача — «Нарисовать в Paint *красный прямоугольник размером 50x100 с координатами верхнего левого угла (10,20)*». Объяснение такой задачи необходимо "привязывать" к более высокому уровню абстракции.

Найдем панель, где располагаются геометрические примитивы и, в частности, необходимый нам геометрический примитив [более абстрактное понятие]-прямоугольник [более конкретное понятие, которое объясняется в задаче]... и т.д.

Прием, который позволяет преподавателю оценить, насколько хорошо сформулирована учебная задача:

Представьте, себя на месте ученика. Дайте себе ту задачу, которую Вы даете ученикам. Попробуйте оценить задачу по критериям

*1) определенность (должна быть однозначно понятна цель, критерии качества);
2) реализуемость (должен быть понятно, как решать задачу):*

Главное: необходимо перед занятием самостоятельно решить задачу, которая будет предложена учащимся, и оценить полученный результат.

Учебные задачи при изложении *теоретического материала* в общем виде могут формулироваться как усвоение последовательности логически связанных смысловых единиц, раскрывающих тему, и примеров, конкретизирующих, подтверждающих или доказывающих (на основе эксперимента или логических рассуждений) основные положения и выводы.

Отсутствие рефлексии по поводу качества ведения занятия

В процессе подготовки к очередному занятию, а также в процессе проведения занятия преподаватель должен задуматься о качестве своих (планируемых или выполняемых) педагогических действий. На основании обобщения теории обучения и опыта практической подготовки преподавателей Локаловым В.А. были предложены следующие компоненты самоконтроля качества подготовки и ведения занятия:

1. Контакт.
2. Управление.
3. Акцент.
4. Контроль.

5. Коррекция.

Контакт — ощущение и понимание текущего состояния (психического, физического, интеллектуального) как группы, так и отдельного учащегося. Контакт обеспечивает необходимые условия взаимодействия преподавателя и учащихся. Бывает, что неопытный преподаватель сам разрывает контакт с учащимися, не обращая внимания на поведение учащихся, реакции понимания или не понимания. В таком случае он вообще рискует быть не услышанным.

Хорошее **управление** группой — это ясные и понятные всем указания, обеспечивающие управление учебной деятельностью всех учащихся. Наиболее эффективное управление возникает при условиях, когда каждый учащийся понимает цель и смысл выполняемого действия, как в контексте задания, так и всего занятия (курса) в целом.

Акцент необходим для привлечения внимания учащихся к главному содержанию (ключевым смыслам) учебного занятия. Акцент следует делать на определении и пояснении смысла новых терминов и понятий, на трудных местах, на типовых ошибках и т.п. Акцент обязательно должен быть связан с решением текущих задач обучения (с содержанием обучения). Способы акцентирования учебного материала зависят от индивидуального стиля преподавателя. Это может быть повышение или понижение высоты голоса, замедление темпа речи, выдержанная пауза перед главным учебным содержанием. Акцент можно сделать с помощью визуально контрастной записи на доске и т.д.

Контроль результата педагогических действий необходим преподавателю как для оперативного внесения изменений непосредственно в ходе урока, так и для **коррекции** методики в целом. Например, если на какой-то фазе занятия преподаватель заметил, что его не понимают, то ему следует сразу же повторить объяснение, а в дальнейшем понять причину непонимания и, возможно, скорректировать саму методику объяснения и (или) способ взаимодействия с группой (контакт, управление, акцент).

Содержание

Введение.....	3
Принципы проектирования системы компьютерных курсов.....	5
Методические особенности и проблемы обучения 3D-графике и анимации.....	22
Методические особенности и проблемы обучения программированию	30
Методические особенности и проблемы обучения по программе «Сети и Web-проектирование»	37
Часто встречающиеся ошибки при проведении занятий и рекомендации по их устранению	45
Литература	51

Литература

1. Ахо, А. В., Хопкрофт, Д., Ульман, Д. Д. Структуры данных и алгоритмы. Пер. с англ. - М.: Издательский дом «Вильямс», 2001. – 384 с.
2. Ван Тассел Д. Стиль, разработка, эффективность, отладка и испытание программ. – М.: Мир, 1981.
3. Вирт Н. Алгоритмы и структуры данных. Новая версия для Оберона. — М.: ДМК Пресс, 2010.
4. Выготский, Л. С. Мышление и речь. Психика, сознание, бессознательное. (Собрание трудов). - М.: Издательство Лабиринт, 2001. – 368 с.
5. Гальперин П. Я. Лекции по психологии: учебное пособие. — М.: Книжный Дом "Университет", 2007. — 400 с.
6. Гилфорд Дж. Три стороны интеллекта. //Психология мышления/Под ред. А.М.Матюшкина. – М.: Прогресс, 1987.- 532с.
7. Грабарь, М. И. Измерение и оценка результатов обучения. - М.: ИОСО РАО, 2000. – 93 с.
8. Дал О., Дейкстра Э., Хоор К. Структурное программирование. – М.: Мир, 1975.
9. Дмитриев В.И., Басова Л.В. Сущность и особенности реализации принципа доступности обучения школьников// Современные проблемы науки и образования, №3 (41), 2012. URL: <http://www.science-education.ru/pdf/2012/3/5.pdf>
10. Ершов А.П., Звенигородский Г.А., Первин Ю.А. "Школьная информатика (концепции, состояние, перспективы)". (Ershov, AP, Zwenigorodski, GA, Pervin, AYU & Yunerman, NA 1983, 'Computers in School. Formulating a National Program', Alberta Printout, vol. 4, no. 2, pp. 34-37.)
11. Иванов В.К. Мировые информационные ресурсы: Конспект лекций. - 2006.
12. Кнут Д. Искусство программирования. Том 1. Основные алгоритмы. В 4-х томах. Пер. с англ. — 3-е изд. — М.: Вильямс, 2017.
13. Леонтьев А.Н.. Деятельность. Сознание. Личность (поновее)
14. Локалов В.А. Развитие творческих способностей школьников на внешкольных занятиях по информатике. Автореферат канд. дисс. - СПб., 1999. - 20 с.
15. Меерович М.И. Формулы теории невероятности. Технология творческого мышления. – Одесса: Полис, 1993. – 232с.
16. Мерзлякова С.В., Пирская А.С., Смирнова Е.В. Основы работы в сети Интернет: Учебно-методическое пособие. – СПб., 2008. – 120 с.
17. Пейперт, С. Переворот в сознании: дети, компьютеры и плодотворные идеи. - Пер. с англ.: М.: Педагогика, 1989. – 224 с.

18. Lokalov V., Klimov I. Design skills assessment in teaching 3D modeling. SHS Web of Conferences. 2019. Vol. 66., https://www.shs-conferences.org/articles/shsconf/pdf/2019/07/shsconf_erpa2019_01013.pdf
19. Lokalov V.A. Children's computer club as an example of non-formal educational system in the field of informatics //Journal of International Scientific Publication: Educational Alternatives, ISSN 1313-2571, Vol. 12, 2014, pp 27-37, <http://www.scientific-publications.net/en/issue/1000008/>
20. Makhlay D.O., Lokalov V.A., Klimov I.V. The development of visual thinking in learning computer 3d modeling//Journal of International Scientific Publication: Educational Alternatives, ISSN 1313-2571, Vol. 12, 2014, pp 72-80, <http://www.scientific-publications.net/en/issue/1000008/>

Локалов Владимир Анатольевич
Миронов Андрей Сергеевич
Климов Игорь Викторович
Константинова Юлия Олеговна

**МЕТОДИЧЕСКИЕ ОСОБЕННОСТИ
ПРЕПОДАВАНИЯ КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ
В СИСТЕМЕ ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
ДЕТЕЙ**

Учебно-методическое пособие

В авторской редакции

Редакционно-издательский отдел Университета ИТМО

Зав. РИО

Н.Ф. Гусарова

Подписано к печати

Заказ №

Тираж

Отпечатано на ризографе

Редакционно-издательский отдел
Университета ИТМО
197101, Санкт-Петербург, Кронверский пр., 49