

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ ФИЗИКО-ТЕХНИЧЕСКИЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

НР

НАУКА В РЕГИОНЫ

***Методические рекомендации
к курсу дополнительного общего образования
«ПРОГРАММИРОВАНИЕ КАК МОДЕЛИРОВАНИЕ»
для третьего года обучения***



МФТИ
Долгопрудный, 2021



Иннопрактика

Автор:
Дединский Илья Рудольфович
Старший преподаватель
кафедры информатики МФТИ



СОДЕРЖАНИЕ

Пояснительная записка	3
Актуальность и новизна программы	4
Общая характеристика курса	6
Цели и задачи обучения	7
Принципы технологии преподавания	10
Структура и объём занятий	17
Общая характеристика тематики проектных работ	19
Начальное обучение программированию	26
Характеристика общей части проектных работ	28
Содержание курса	30
Материально-техническое обеспечение образовательного процесса	39
Литература	40

Пояснительная записка

Курс рассчитан на проведение занятий для детей с физико-математическим уклоном в соответствии с концепциями об универсальности и профилизации обучения. Основная направленность занятий заключается не в рассмотрении информатики как простого продолжения математических дисциплин и/или отработке чисто компьютерных программистских или пользовательских навыков, а в обучении универсальным и эффективным методам работы с информацией в различных областях знания и технологии физико-математической направленности. Основной тематикой работ является моделирование различных физических и математических объектов и процессов.

Специфика информатики как синтетической науки, включающей элементы математики, философии, стилистики, психологии и инженерии, состоит именно в том, что преобладающий способ реализации её собственных знаний — это работа со знаниями более «конкретных» дисциплин (предметных областей).

Сочетание обучения методам, а не фактам, работы на примерах из смежных школьных дисциплин, универсальности большинства изучаемых приёмов превращает, таким образом, информатику в дисциплину, максимально интегрирующую предметы школьного курса. Такая концепция приводит к повышению эффективности процесса обучения в целом.

Необходимо подчеркнуть, что в ряде курсов наиболее существенным приобретением ученика становятся не столько знания, умения и навыки, сколько метаумения и метанавыки,

определяющие правильное формирование подхода к проблеме, способа её рассмотрения, поиска решения и реализации, а, в конечном счёте, и организацию знания, получаемого ребёнком. Это является следствием системного подхода, применяемого как непосредственно в процессе обучения, так и при планировании занятий. В конечном счёте, речь идёт о выработке у ученика элементов системной методологии мышления. При полном отсутствии этих компонент получается стандартное выучивание и воспроизведение, при полном наличии — в идеале становится ненужным традиционный учебный процесс (ребёнок способен получать и организовывать материал самостоятельно). В ходе обучения достигается компромисс между этими формами, он определяется индивидуально для каждого ребёнка, в зависимости от его мотиваций и склонностей.

Актуальность и новизна программы

В настоящее время проектная составляющая в школьном обучении программированию сильно уменьшилась за последние 10-15 лет, превратившись в моделирование в **Excel** и знанию возможностей **PowerPoint**, и, к сожалению, уступив в массовости олимпиадам, подразумевающим небольшой объём каждой задачи и крайне малое время жизни кода.

Работа над проектными задачами, подразумевающими не только логику и расчёты, но и визуализацию, потребовала от учащихся гораздо больших навыков и даже совсем другого стиля мышления, чем сейчас даётся школьникам как в общеобразовательном курсе программирования, так и в курсах олимпиадной направленности, читающихся в огромном количестве на разных курсах дополнительного образования.

Здесь можно присоединиться ко многим вузовским преподавателям, физикам, биологам, лингвистам и другим профессиональным представителям предметных областей, да и самим программистам, утверждающим, что сейчас даже среди школьников физико-математической направленности стали редки те, кто может взять задачу за пределами олимпиад по программированию и грамотно решить её, не переписывая по сто раз и не теряя понимание собственного кода программы. Такая тенденция удручает.

Актуальность подхода заключалась в длительном характере работы над проектом (в течение смены) и сравнительно большой для школьников объём кода проекта (более 500 строк). Это объяснялось тем, что в работах присутствовала как расчётная, так и интерфейсная часть, которая почти всегда составляла большую часть кода проекта, что вообще характерно для визуализации. Такое положение дел неизбежно приводит к необходимости строгой архитектуры проекта, что зачастую игнорируется в школьных курсах программирования. «Традиционный» подход, где эти части смешиваются, принципиально неприменим для задач, решение которых требует больше 3-4 учебных часов и значительный объём кода, а это именно тот тип задач, который характерен для вузов и для профессиональной деятельности, и к которому надо готовить школьника, чтобы в дальнейшем не пришлось его переучивать.

Новизна подхода заключается в большом упоре на технологическую грамотность проекта и его проработанность с точки зрения современных технологий программирования. Зачастую проектные работы, выполненные школьниками, смотрятся неплохо с точки зрения интерфейса, но их внутренняя архитектура, надёжность, технологическая грамотность на всех уровнях оставляют желать много и много лучшего, что обычно объясняется тем,

что в этих проектах «важнее всего физика, а не программирование». На практике же это обычно означает, что школьник (а далее и студент) не может написать и отладить программу большего объёма даже на аналогичную тему, путается в коде и в отладке, и в конце концов бросает проект, а плохое качество кода не даёт его последователям в полной мере использовать разработанные методики моделирования. Чтобы научиться работать с проектами не школьного уровня, ему необходимо серьёзно переучиваться и менять стиль мышления при программировании, что удаётся не каждому. Остальные вместо самостоятельной работы над задачей «нанимают программистов» и вынуждены тратить время и силы на постановку им технических заданий и верификацию данных, и финансы на собственно разработку. Таким образом, такой исследователь, скажем, учёный-физик, становится несамостоятельным, что ограничивает эффективность его работы. Используемый подход, основанный на принципе «разрабатывать небольшие учебные проекты так, как это делается в больших», призван устранить эту часто встречающуюся проблему.

Общая характеристика курса

Курс рассчитан на учащихся физико-математических школ, изучавших основы программирования в том или ином объёме.

Программой предполагается проведение проектных работ, рассчитанных на четверть, полугодие или год, и направленных не только на отработку отдельных технологических приёмов, но и созданию целостной работы, суммирующей в себе знания ученика как по программированию, так и по ряду предметных областей, связанных с темой работы. Со-

держание теоретической и практической компонент курса находится в соотношении приблизительно 1:2 – 1:4. Для облегчения реализации работа разбивается на части. Формы контроля — устный опрос, беседа, поэтапная сдача проекта.

Курс излагается с опорой на развитие самостоятельного мышления учащихся, элементы самообразования. Для выполнения учащимися выбираются задания, допускающие решения в эстетически привлекательной форме, имеющие игровой компонент.

Цели и задачи обучения

Общей целью курса в контексте основного общего образования с учётом специфики учебного предмета является формирование универсального стиля научного и практического мышления, основанного на системном подходе, направленного на самостоятельное творческое познание и исследование окружающего мира. Это конкретизируется в виде изучения общих закономерностей функционирования, создания и применения автоматизированных информационных систем — программ, разрабатываемых учащимися.

Общеучебные задачи курса

- Организация процессов мышления и познания.
- Обучение наиболее общим приёмам работы с информацией, пригодных для использования в любой предметной области.
- Интеграция программ школьного курса, взаимное усиление информатики и других школьных предметов по принципу: технология работы с информацией — из информатики, конкретные примеры и задачи — из смежных предметов.

- Развитие самостоятельности в работе и творческой активности.
- Усвоение и развитие технологии рационального решения задач:
- Определение адекватных способов решения учебной задачи на основе заданных алгоритмов
- Комбинирование известных алгоритмов деятельности в ситуациях, не предполагающих стандартное применение одного из них.
- Разработка собственных алгоритмов решения задач.
- Использование для решения познавательных и коммуникативных задач различных источников информации.
- Владение умениями совместной деятельности (согласование и координация деятельности с другими её участниками, объективное оценивание своего вклада в решение общих задач коллектива; учёт особенностей различного ролевого поведения).

Внутрипредметные общие задачи курса

- Освоение знаний, составляющих основу научных представлений об информации, информационных процессах, системах, технологиях и моделях.
- Овладение умениями работать с различными видами информации с помощью компьютера и других средств информационных и коммуникационных технологий, организовывать собственную информационную деятельность и планировать её результат.
- Выработка навыков применения средств ИКТ в повседневной жизни, при выполнении индивидуальных и коллективных проектов, в учебной деятельности.
- Развитие познавательных интересов, интеллектуальных и творческих способностей средствами ИКТ.

- Воспитание ответственного отношения к информации с учетом правовых и этических аспектов её распространения, избирательного отношения к полученной информации.
- Выработка навыков применения средств ИКТ в повседневной жизни, при выполнении индивидуальных и коллективных проектов, в учебной деятельности, дальнейшем освоении профессий, востребованных на рынке труда.

Ценностные ориентиры содержания учебного предмета

- Важнейшей частью курса является формирование системы учебных ценностей (предпочтений) ученика. В конечном счёте, это формирование и есть основная инвариантная методологическая задача курса, так как всё остальное — технология и будет неотвратно изменяться с течением времени.
- Во главу угла ставится задача, её решение и, главное, путь от задачи к решению, а не программирование (кодирование) как таковое.
- Для записи алгоритма на языке программирования выбирается минимальное подмножество средств языка, чтобы не акцентировать внимания на кодировании и для более лёгкого перехода на другие языки программирования.
- Самостоятельность решения является ключевым условием, которое необходимо доказать при сдаче работы.
- Понимание учащимся тех средств, с помощью которых он решил задачу, ставится выше уровня самих средств решения.
- Аккуратность и надёжность решения ставятся выше «программистских трюков», иногда позволяющих в отдельных случаях добиться несколько лучших результатов.
- Главным методологическим принципом является системный подход.

Принципы технологии преподавания

1. Принцип самостоятельного исследования объектов и явлений с точки зрения системного подхода. Методы подачи материала:

- Обучение навыкам самостоятельной работы с информацией:
- Техника и технология информационного поиска («где что?»).
- Определение структуры системы понятий («что к чему относится?»).
- Поиск базовой системы понятий («что главное?»).
- Связь с другими предметными областями («где еще?»).
- Исследование «черных ящиков» (приборов, программ, учебных тем).
- Использование знаний об аналогах и более общих объектах.
- Выявление общих принципов устройства, работы.
- Выявление характерных свойств объектов и иерархия этих свойств.

2. Принцип творчества, для:

- Закрепления полученных теоретических навыков.
- Эмоционального подкрепления в ходе творческой работы.
- Организации и планирования проектов, обучения планированию.
- Взаимодействия учеников в коллективных проектах.

3. *Принцип эмоционального подкрепления, для:*

- Увеличения эффективности усвоения.
- Развития уверенности в своих силах.
- Развития эстетики.

4. *Принцип технологичности:*

- Постоянное стремление повышать КПД своей работы.
- Умение работать над конкретной задачей в контексте её возможного развития.
- Развитие технология решения задачи вместе с её собственно решением.
- Обучение технологии на сравнении «плохих» и «хороших» примеров (от первых — ко вторым).
- Формирование технологически корректной системы ценностей ученика на уровне «люблю работать высокоэффективно» и «не люблю рутинные и ненадёжные действия, ищу, как сделать это лучше».

5. *Принцип смежных дисциплин:*

- Разбор материала из смежных предметов, работа на примерах из смежных областей.
- Взаимное усиление информатики и смежных дисциплин.
- Интеграция смежных дисциплин.

6. Принцип открытой педагогической технологии:

- Открытая и ясная система ценностей, установок.
- Подчёркнутая расстановка акцентов в подаче материала.
- Понимание «критических мест» в ходе изучения и способов их преодоления.
- Понимание связи материала и методики его подачи.
- Активное использование взаимодействия учеников и взаимного обучения.

Результаты освоения курса

Результатом прохождения курса учащимися должны стать понимание основных принципов программирования, его применения для решения различных межпредметных задач и владение основными необходимыми для этого алгоритмическими конструкциями.

Личностные результаты

- Владение навыками анализа и критичной оценки получаемой информации с позиций её свойств, практической и личной значимости, развитие чувства личной ответственности за результат.
- Оценка окружающей информационной среды и формулирование предложений по её улучшению.
- Организация индивидуальной информационной среды, в том числе с помощью применения средств программирования и автоматизации.
- Использование различных источников информации различных типов для повышения своего образовательного уровня и подготовке к продолжению обучения.

Метапредметные результаты

- Владение основными общеучебными умениями информационного характера: анализа ситуации, планирования деятельности, обобщения и сравнения данных и другими.
- Получение опыта использования методов и средств информатики: моделирования; формализации структурирования информации; компьютерного эксперимента при исследовании различных объектов, явлений и процессов.
- Умение создавать и поддерживать индивидуальную информационную среду, обеспечивать защиту значимой информации и личную информационную безопасность.
- Владение навыками работы с основными, широко распространёнными средствами информационных и коммуникационных технологий.
- Умение осуществлять совместную информационную деятельность, в частности при выполнении проекта.

Предметные результаты

В сфере познавательной деятельности

- Освоение основных понятий и методов программирования и компьютерного моделирования.
- Выделение основных информационных процессов в реальных системах, нахождение сходства и различия протекания информационных процессов в различных предметных областях.
- Выбор язык представления информации в соответствии с поставленной целью, определение формы представления информации, отвечающей данной задаче (таблицы, схемы, графы, диаграммы и др.).

- Преобразование информации из одной формы представления в другую без потери смысла и полноты информации.
- Оценивание информации с позиций её свойств (достоверность, объективность, полнота, актуальность и т.п.).
- Развитие представлений об информационных моделях как основном инструменте познания, общения, практической деятельности, знания основных областей применения метода моделирования.
- Разработка и запись алгоритмов, т.е. построение модели решения задачи с применением основных алгоритмических конструкций для описания алгоритмов, проверка правильности алгоритма, нахождение и исправление типовых ошибок.
- Освоение основных конструкций языка программирования.
- Знание основных свойств алгоритма, типов алгоритмических конструкций: следование, ветвление, цикл; понятия вспомогательного алгоритма и других.
- Знание программного принципа работы компьютера.
- Умение выполнять базовые операции над объектами: цепочками символов, числами, списками, деревьями; проверять свойства этих объектов; выполнять и строить простые алгоритмы.
- Умение проводить компьютерные эксперименты с использованием моделей объектов и процессов.
- Оценивание числовых параметров информационных процессов (объёма памяти, необходимого для хранения информации, скорости вычислений и т.п.).
- Решение различных задач из разных сфер человеческой деятельности с помощью средств информационных технологий.

- Приобретение опыта использования информационных ресурсов общества и электронных средств связи в учебной и практической деятельности.
- Осуществление мер по обеспечению защиты значимой информации и индивидуальной информационной безопасности, в частности, при работе в сети Интернет.

В сфере ценностно-ориентационной деятельности

- Понимание роли информационных процессов как фундаментальной реальности окружающего мира и определяющего компонента современной информационной цивилизации.
- Умение выделять критерии оценки информации, получаемой из разных источников.
- Оценка информации, в том числе получаемой из средств массовой информации, свидетельств очевидцев, интервью: использование ссылок и цитирование источников информации, анализ и сопоставление различных источников.
- Выявление проблем жизнедеятельности человека в условиях информационной цивилизации и оценка предлагаемых путей их разрешения, умение пользоваться ими для планирования собственной деятельности.
- Следование нормам жизни и труда в условиях информационной цивилизации.

В сфере коммуникативной деятельности

- Понимание особенностей представления информации различными средствами коммуникации (на основе естественных, формализованных и формальных языков).
- Осознание основных психологических особенностей восприятия информации человеком.

- Овладение навыками использования основных средств телекоммуникаций, формирования запроса на поиск информации в Интернет с помощью программ навигации (браузеров) и поисковых программ, осуществления передачи информации по электронной почте и др.
- Соблюдение культуры поведения в сети Интернет.

В сфере трудовой деятельности

- Понимание принципов действия различных средств информатизации и автоматизации, их возможностей и ограничений.
- Овладение навыками использования технических средств информационных технологий для решения различных задач.
- Знакомство с основными программными средствами компьютера (круг решаемых задач, система команд и пр.).
- Умение тестирования используемого оборудования и программных средств.
- Выбор соответствующего средства информационных технологий для решения поставленной задачи.

В сфере эстетической деятельности

- Приобретение навыков компьютерного дизайна.
- Овладение умениями создания эстетически-значимых объектов с помощью возможностей средств информационных технологий (графических, цветовых, звуковых, анимационных).
- Демонстрация на примерах эстетически-значимых компьютерных моделей из различных образовательных областей.

В сфере физической деятельности

- Понимание особенностей работы со средствами информатизации, их влиянии на здоровье человека.
- Владение профилактическими мерами при работе с средствами информатизации.
- Соблюдение требований безопасности и гигиены в работе с компьютером и другими средствами информационных технологий.

Структура и объём занятий

Работа велась с учащимися, закончившими 7 и 8 класс физико-математической школы. Учащиеся были разделены на четыре группы: три группы седьмых классов и одна группа восьмого, также индивидуальная работа велась с учащимися десятых классов.

Работа учащихся была разделена по направлениям, соответствующим тематике проектных работ:

- математическое моделирование процессов в идеальном газе,
- математическое моделирование динамики поступательного и вращательного движения, систем связанных объектов,
- численное моделирование процессов теплопереноса и динамики водной среды,
- визуализация трёхмерных геометрических объектов и моделирование распространения световых лучей в неоднородной оптической среде,

- проведение экспериментального сравнения эффективности алгоритмов сортировки,
- моделирование системы взаимодействующих частиц на примере сетевой игры,
- моделирование представления знаний и реализация экспертных систем,
- обработка данных физического эксперимента (фиттинг функций),
- начальное обучение программированию.

Теоретическая часть занятий включает в себя общетехнологические и общеметодические сведения, так или иначе касающиеся любого направления работ. Также теоретическая часть включает семинары общего характера, посвящённые рассмотрению разделов, специфичных для каждого данного направления. Эти семинары проводятся для всех учащихся, но адаптируются к общему уровню группы и специфике её интересов. Дальнейшее обсуждение углубление этой специфической тематики проводится в рамках индивидуально-групповой работы с учащимися в рамках реализации конкретных задач, но так, чтобы сама реализация проектов учащимися оставалась самостоятельной.

Объём обязательных занятий составляет 64 академических часа для группы в 8-10-15 чел. Факультативные занятия для желающих проводятся в объёме 4-8 академических часов еженедельно. Наполняемость групп факультативных занятий составляет 4-15 человек.

Общая характеристика тематики проектных работ

Математическое моделирование процессов в идеальном газе

Основной задачей является моделирование различных физических (термодинамических) процессов, происходящих в идеальном газе, и вычисление его интегральных характеристик. Учащиеся реализуют математическую модель поведения идеального газа, состоящего из большого количества частиц, помещённую в ёмкость со впускными и выпускными клапанами для газа, поршнем и датчиками давления и температуры. Частицы могут иметь разную массу и взаимодействовать путём соударения, что приводит к задачам определения соударений (*collision detection*) и реакции на соударение (*collision response*), что ведёт к известным в физике задачам центрального и нецентрального удара шаров. Изображение ёмкости с частицами выводится на экран, показания датчиков отображаются на графиках и гистограммах, динамически изменяющихся во время работы программы. Модель реализуется на языке C++ и визуализируется средствами двумерной компьютерной графики.

Конкретными задачами этой тематики может быть проведение модельных экспериментов по зависимости давления, температуры и объёма в газах (частичные газовые законы и уравнение Клапейрона-Менделеева), установление равновесия в газовой среде после возникновения возмущения (изменение температуры, давления и объёма), установление равновесия по скоростям в результате соударений молекул, теплоперенос в газах, явления конденсации, распространение волн в газовой среде.

Математическое моделирование динамики поступательного и вращательного движения, систем связанных объектов

Основной задачей является моделирование механического движения системы материальных точек, связанной жёсткими связями. Так как необходимый для этого теоретический материал не проходит в школьном курсе физики, встаёт необходимость его адаптивного изложения группе учащихся. В ходе работы реализуется представление векторов в программе и разрабатывается библиотека операций для них, включая повороты координатных систем. Модель реализуется на языке C++ и визуализируется средствами двумерной компьютерной графики.

Конкретными задачами этой тематики может быть реализация динамики материальной точки, рассмотрение природы вращательного движения и программная реализация необходимого математического аппарата, реализация динамики связанных пар материальных точек, реализация композиций из большего числа точек с жёсткими связями.

Численное моделирование процессов теплопереноса и динамики водной среды

Темами этой группы проектов выступают классические вычислительные задачи в математической физике: численное интегрирование уравнений теплопереноса Фурье и уравнений динамики жидкости Навье-Стокса (в упрощённой форме). Для этих проектов необходимая теоретическая база отсутствует в школьном курсе как по физике, так и по математике – интегральному и дифференциальному исчислениям, численным методам сеточного моделирования. Это компенсируется занятиями по основам этих разделов в адаптивном изложении,

без выводов и строгих доказательств, но с объяснением механизмов процессов и физической природы рассматриваемых явлений. В силу этих причин эти темы не подходят для массового изложения восьми- и девятиклассникам, однако в индивидуальном порядке, на преподавательском (относительно вузовского) уровне, могут быть достигнуты успехи, что позволит учащимся самостоятельно написать и отладить моделирующие программы. Модели реализуются на языке C++ и визуализируются средствами двумерной и трёхмерной компьютерной графики в среде Windows с помощью библиотек TXXLib и OpenGL.

Конкретными задачами этой группы могут быть моделирование кинетики нагрева металлических двумерных предметов произвольной формы и моделирование динамики водной поверхности при различных возмущениях (падении капель, неравновесных начальных условиях и др.).

Необходимо отметить сложность визуализации этого рода моделей, особенно в случае трёхмерного отображения. Здесь будут значительны не только трудности в программировании и необходимости изучения ряда методов линейной алгебры, но и в подборе параметров визуализации, чтобы результирующая демонстрация выглядела убедительной и эстетичной.

Визуализация трёхмерных геометрических объектов и моделирование распространения световых лучей в неоднородной оптической среде

Тематика этой группы работ – трёхмерные визуализации математических и физических объектов и явлений. Здесь необходимо ознакомиться с основными понятиями векторной и линейной алгебры, аффинными преобразованиями, матрицами и операциями над ними, а также с алгоритмами закраски поверхностей и сокрытия невидимых поверхностей. В ходе работы

реализуется собственная графическая трёхмерная библиотека с архитектурой, схожей с библиотекой **OpenGL**, что в последующем облегчит переход учащихся на **OpenGL** и понимание механизмов функционирования современных графических библиотек.

Модели реализуются на языке C++ и визуализируются средствами двумерной и трёхмерной компьютерной графики

Одной из конкретных задач может быть самостоятельная трёхмерная визуализация геометрических объектов, построенных из треугольников (куба, тетраэдра, различных поверхностей), с закраской поверхностей и моделированием пересечений, реализованная на базе двумерной графической библиотеки, основанной на **Windows GDI**. Другой конкретной задачей может быть реалистическая визуализация распространения светового луча в неоднородной оптической среде, реализованная с помощью трёхмерных библиотек **OpenGL** или **Microsoft DirectX**.

Проведение экспериментального сравнения эффективности алгоритмов сортировки

Задачи этой группы наиболее близко стоят к классическим внутренним задачам программирования, что позволяет начать их реализацию с задач, близких по тематике и способам подачи к школьным. Однако, в отличие от распространённого сейчас школьно-олимпиадного подхода, эти задачи не «выбрасываются» после реализации, а используются для вычислительного эксперимента по сравнению эффективности сортировок массивов. Программа сортирует массивы различной длины различными алгоритмами, подсчитывает количество

операций сравнения элементов массивов и их обменов, и строит графики зависимости этих величин от длины массивов. Такой подход позволяет легко визуализировать понятие эффективности алгоритмов, детально сравнить алгоритмы в случае сортировки объектов разной природы (с быстрым или медленным сравнением, с быстрым и медленным обменом) и построить практическое руководство по выбору подходящего алгоритма сортировки для различных объектов и различных состояний исходного массива. Для слабых учащихся, которым труден системный подход к программированию со следованием чёткой архитектуре, есть возможность в качестве промежуточного этапа выводить данные по эффективности в файл и далее строить графики в Excel, что позволяет выровнять сложность этого проекта в случае гетерогенной или слабой группы.

Конкретными задачами этой группы могут быть сравнение эффективности различных обменных сортировок (пузырьковой, шейкерной, выбором, простыми и двоичными вставками, быстрой, слиянием, подсчитывающей и др.) по количеству сравнений и обменов и её визуализацией в виде графиков.

Модель реализуется на языке C++ и визуализируется средствами двумерной компьютерной графики.

Моделирование системы взаимодействующих частиц на примере сетевой игры

Разработка приложений клиент-серверной архитектуры давно стала стандартом в промышленном программировании, но такой подход требует особо дисциплинированного

архитектурного мышления и умения думать на сторонах клиента и сервера, что совершенно не развивается в школьном курсе программирования. В то же время, реализация сетевых приложений является для школьников желанным делом и это хороший способ повысить их профессиональную грамотность. Проект разрабатывается в команде, хорошо имитирует взаимоотношения в профессиональной среде разработчиков, ставит школьников в довольно «взрослые» условия, но при этом остаётся очень педагогически адаптивным и удобным для проведения. Наибольшую сложность в изучении представляет концепция сокетов, но разработка отдельного модуля (библиотеки) для её реализации позволяет более слабым участникам команды воспользоваться ей, как готовым продуктом, но при этом полностью осознавая базовые понятия сетевых технологий (протокол, адрес, пакет и т.п.).

Конкретными задачами этой тематики может быть реализация программной библиотеки для передачи данных по сети на базе **Windows Sockets**, реализация однопользовательской и парной игры «Агар», моделирующей поведение микроорганизмов, разработка стандарта протокола обмена данными по сети и реализация этого протокола, разработка сервера игры и клиентских приложений.

Модель реализуются на языке C++ и визуализируется средствами двумерной компьютерной графики и **Windows Sockets** в среде **Windows**.

Моделирование представления знаний и реализация экспертных систем

Эти задачи когда-то считались задачами «искусственного интеллекта» (например, известная система **Mycin**), хотя их реализация в самом простейшем виде несложна для тех уча-

щихся, которые знакомы с основными структурами данных и алгоритмами над ними. Сейчас такие программы широко используются в автоматической технической и медицинской диагностике, составлении определителей. За основу берётся хранение базы данных в виде двоичного дерева, в листьях которого хранятся объекты, а в узлах – вопросы, классифицирующие объекты. Пользователь задумывает объект, находящийся или не находящийся в базе, и далее отвечает на вопросы, расположенные вдоль пути от корня к листу, содержащему объект. Если объект в базе присутствует, он таким образом будет найден, если нет – он добавляется в базу с указанием уточняющего вопроса, отличающего его от соседнего объекта. Путём сравнения путей от корня к объектам находят сходства и различия объектов, а сами пути являются определениями.

Конкретной задачей может быть моделирование экспертной системы в виде двоичного спра-вочника, реализующей диалог с пользователем, добавление объектов в базу данных и их классификация, автоматическое построение определений и сравнение объектов.

Модель реализуется на языке C++ и визуализируется средствами двумерной компьютерной графики.

Обработка данных физического эксперимента (фиттинг функций)

Обработка данных физического эксперимента часто сводится к линеаризации полученных экспериментальных данных с целью облегчения проведения прямой по линеаризованным данным и дальнейшем нахождении коэффициентов этой прямой, из которой выводятся результаты измерений. Данный метод по сути прост, но требует отдельного вывода преобра-

зований линеаризации для каждой конкретной зависимости, и, следовательно, неэлементарен для автоматизации в общем случае. Кроме того, линеаризация не является аффинным преобразованием, поэтому, отображая экспериментальную точку в соответствующую точку на линеаризованном графике, она искажает величины погрешностей измерений при каждой точке. Минимизация (фиттинг) функций лишена этого недостатка и её гораздо проще реализовать независимо от формы исследуемой зависимости, но может быть выполнена только машинно, так как связана с большим объёмом вычислений. Это удачно ложится на разработку компьютерной программы.

Конкретными задачами может быть фиттинг функций, описывающих различные физические закономерности, с целью минимизации погрешностей в экспериментальных данных. Одной из задач может быть измерение начальной скорости полёта футбольного мяча после удара футболистом и ускорения свободного падения вблизи поверхности Земли.

Модель реализуются на языке C++ и визуализируется средствами двумерной компьютерной графики. Данные траектории футбольного мяча можно получить обработкой видеозаписи удара при высокой скорости съёмки.

Начальное обучение программированию

Высокая гетерогенность учащихся часто обуславливает необходимость проведения начального обучения программированию среди ряда школьников. Среди них бывают те, кто ранее совсем не изучал программирование, иногда большую часть этой группы составляют

школьники, 1-2 года занимавшихся так называемым «олимпиадным программированием», состоящим в решении большого количества очень маленьких с проектной точки зрения задач, имеющих чисто логический характер и почти никак не связанных с предметными областями. После 15-60 минут решения такой задачи ребёнок переключается на другую задачу, что даёт очень малое время жизни кода и он быстро теряет свою актуальность, так как по условиям олимпиад нельзя использовать свои наработки от предыдущих задач (так в олимпиадах ведётся борьба со списыванием). Педагогически это приводит к крайне небрежному отношению к коду и особенно к архитектуре программы, что почти неизбежно ведёт к неспособности ребёнка решать задачи больше минимальных и особенно внешних с точки зрения собственно программирования, то есть они фактически неспособны написать никакую полезную для себя и других программу, кроме самых простых. Автор неоднократно замечал это при работе со школьниками физико-математических школ и с первокурсниками МФТИ и МГУ. Это проявляется и по мнению самих детей, когда они сравнивают свои умения и результаты с результатами детей, занимавшихся с использованием проектного подхода, который при этом не исключает, а развивает логический, придавая ему актуальность.

Поэтому для таких детей имеет смысл организованы занятия по начальному обучению программированию. Часть из них попытается использовать свои олимпиадные навыки для того, чтобы обучаться не совсем с нуля, но тут может сказаться воспитанная у них привычка работать над задачей недолго и неумение делить задачу на части, что породит необходимость выполнять (учителю) очень мелкое деление на подзадачи и, как следствие, невысокий

темп обучения. Проектный подход таким детям нравится, определённых результатов они достигают, но им, безусловно, хочется большего. Та часть детей, которые занимаются программированием недавно, часто хочет изучать программирование вообще с нуля, мотивировав это тем, что в предшествующем обучении им было далеко не всё понятно и часто их решения были не вполне самостоятельными. Во время занятий они успешно и с большим удовольствием проходят часть курса по начальному обучению программированию с применением проектного подхода.

Занятия начальному обучению проходят позитивно и результативно, особенно у тех, кто ходил не только на уроки, но и на факультативы, Однако очень жаль, что заинтересованные и часто перспективные дети часто становятся заложниками однобоких олимпиадных подходов, которые, безусловно, удобны для ряда преподавателей с прагматической точки зрения.

Характеристика общей части проектных работ

Как говорилось выше, основной особенностью проводимых проектных работ является длительный характер работы и сравнительно большой для школьников объём кода проекта (более 500 строк). Это объясняется тем, что в работах присутствует как расчётная, так и интерфейсная часть, которая почти всегда составляет большую часть кода проекта, что вообще характерно для задач визуализации. Такое положение дел неизбежно приводит к необходимости строгой архитектуры проекта, что зачастую игнорируется в школьных курсах программирования. В качестве основного архитектурного паттерна была выбрана концепция

MVC (Model-View-Controller), применяемая при решении многих задач с визуализацией и интерактивным взаимодействием с пользователем. Это позволило учащимся упорядочить составные части своего проекта и отделить расчёты от визуализации. «Традиционный» подход, где эти части смешиваются, принципиально неприменим для задач, решение которых требует больше 3-4 учебных часов и значительный объём кода.

Эти же особенности, на более низком уровне архитектуры, приводили к повышенным требованиям к структуре программы, ее декомпозиции на классы, функции и библиотеки, к форматированию кода, к именам идентификаторов, к простоте и ясности изложения алгоритмов на языке программирования. В условиях большого объёма кода эти требования были для учащихся совершенно естественными, так как иначе они просто начинали путаться в своих проектах.

Синтетический характер задач в сочетании с, опять же, значительным объёмом кода, приводит к повышенным требованиям к надёжности программ и их самодиагностике, что часто игнорируется в проектах школьников. Здесь же сделанные вовремя проверки и развёрнутые диагностические сообщения воспринимаются учащимися как благо, облегчающее отладку программ и в целом работу над ними.

Необходимость длительной работы обуславливает хранение кода в репозиториях и применение системы контроля версий (**git**). Это обеспечивает сохранность кода в условиях работы компьютерного класса, а также делает удобным отслеживание изменений в проектах и откат к предыдущим версиям, если это было необходимо.

Для работы используются языки программирования C++ и язык программирования шейдеров HLSL, среды программирования **Microsoft Visual Studio** и **CodeBlocks** (компилятор g++ 4.8.1 в среде MinGW).

По ряду тематик детям имеет смысл подготовить тезисы работ и краткие доклады с демонстрацией работы программ, по которым могут быть сняты короткие видеоролики.

Содержание курса

Общая часть проектных работ

№	Тема занятия	Содержание
1	Техника безопасности	ТБ при работе в компьютерном классе. Правила пользования классом и компьютерной сетью Центра. Размещение пользовательских данных? установка и настройка необходимых программ.
2	Репозитории кода и системы контроля версий	Жизненный цикл проекта. Понятие версии. Системы контроля версий, их преимущества. Работа с системой контроля версий, понятие коммита. Понятие репозитория кода, необходимость репозитория. Репозитории git на github.com и работа с ними.
3	Архитектура программного обеспечения (1)	Проблема сложности ПО и понятие архитектуры. Микроархитектура (функциональная декомпозиция). Проблемы с глобальными данными. Архитектура как описание декомпозиции и взаимодействия. Функциональный подход к архитектуре.

<i>№</i>	<i>Тема занятия</i>	<i>Содержание</i>
4	Архитектура программного обеспечения (2)	Основные виды архитектурных паттернов. Слоёная (layered) архитектура и программные библиотеки. Модули, декомпозиция на модули. Модули в задачах визуализации. архитектурный паттерн MVC.
5	Надёжность программного обеспечения	Проблема надёжности. Виды и примеры ошибок. Истории поиска ошибок. Контроль входных данных, параметров и действий пользователя. Внутренний контроль в программе. Контроль входных параметров функций. Понятие транзакции, виды транзакционных гарантий по Абрамсу. Двойной контроль объектов при изменении.

Математическое моделирование процессов в идеальном газе

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Динамика материальной точки и идеального газа	Динамика материальной точки, программа моделирования динамики. Взаимодействие молекул. Задача взаимодействия сферических тел в случае центрального и нецентрального ударов.
2	Операции над векторами и их реализация	Представление вектора в виде класса. Операции над векторами. Поворот системы координат. Проекция вектора на прямую.
3	Обзор задач тематики	Задачи об идеальном газе. Моделирование газовых законов, теплопереноса, равновесных процессов.

Математическое моделирование динамики поступательного и вращательного движения, систем связанных объектов

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Динамика вращательного движения	Природа вращательного движения. Задача об «идеальной гантели». Угловые кинематические величины. Тангенциальное (касательное) пространство. Разложение движения на линейную и тангенциальную составляющие.
2	Динамика связанных объектов	Задача динамики связанных тел. Системы уравнений связей. Решения системы уравнений связей.

Численное моделирование процессов теплопереноса и динамики водной среды

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Введение в понятие дифференциального уравнения и его численное решение	Понятие о производной. Понятие о дифференциальном уравнении. Однородные и неоднородные уравнения. Задача теплопереноса. Одномерные задачи, приводящие к дифференциальным уравнениям. Простейшие уравнения математической физики.
2	Введение в численные методы решения дифференциальных уравнений	Дискретизация и обезразмеривание задачи. Разностные отношения для первой и второй производной. Простейшие разностные схемы. Сеточное моделирование, понятие временного слоя. Понятие граничного слоя, краевые эффекты в сеточных представлениях.
3	Векторные поля и операции над ними	Примеры векторных и скалярных полей. Оператор набла. Операторы градиента, дивергенции и Лапласа, их реализация и выражение через оператор набла. Разностное представление оператора Лапласа.

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
4	Задача теплопереноса и уравнение теплопроводности	Нестационарное уравнение теплопроводности. Явные разностные схемы. Задача Коши для уравнения теплопроводности. Типы граничных условий. Понятие о сходимости разностных схем и критерии Куранта. Моделирование распространения тепла по неоднородной среде.
5	Задача динамики водной среды и уравнение Навье–Стокса	Уравнение Навье–Стокса для несжимаемой жидкости. Волновое уравнение. Уравнение мелкой воды. Динамическая и кинематическая вязкость. Явная разностная схема для уравнения второго порядка. Начальные условия на двух временных слоях. Граничные условия и отражение волн. Моделирование поверхности воды в ограниченной области с препятствиями.

Визуализация трёхмерных геометрических объектов и моделирование распространения световых лучей в неоднородной оптической среде

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Задача проектирования трёхмерных тел	Проектирование трёхмерных тел на двумерную плоскость, виды проекций. Реализация ортогональной и перспективной проекций.
2	Аффинные преобразования в трёхмерном и четырёхмерном пространстве	Виды аффинных преобразований (смещение, масштабирование, поворот) и их реализация. Эффективность композиции аффинных преобразований. Матрицы, их свойства, матричный вид аффинных преобразований.
3	Архитектура конвейера рендеринга. Закраска поверхностей.	Задача о рендеринге примитива (треугольника). Линейная и билинейная интерполяции. Интерполяция цветов.

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
4	Алгоритмы отсечения невидимых поверхностей	Задача отсечения. Метод отсечения задних граней по векторному произведению. Общие методы отсечения поверхностей. Алгоритмы художника, другие векторно-ориентированные алгоритмы. Алгоритм Z-буфера, его реализация.
5	Разработка на платформе Microsoft DirectX 11	Архитектура DX11, понятие конвейера и шейдера. Растеризация примитива в DX11. Создание контекста рендеринга DX11 и работа с ним. Буфера DX11. Шейдеры HLSL, их разработка. Отладка шейдеров. Реализация системы частиц. Реализация полупрозрачности. Задача трассировки светового луча. Реализация трассировки луча с его преломлением.

Проведение экспериментального сравнения эффективности алгоритмов сортировок

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Задача сравнения эффективности алгоритмов	Проблема роста количества операций. Вывод количества операций сравнения и обмена для пузырьковой сортировки и сортировки выбором. Понятие об асимптотической эффективности.
2	Экспериментальный анализ эффективности сортировки пузырьком	Методы подсчета количества операций. Реализация вывода количества операций на экран и в файл. Построение графиков количества операций в зависимости от длины массива. Сравнение с теоретическими данными.
3	Экспериментальный анализ эффективности сортировки выбором	Реализация подсчета количества операций в сортировке выбором. Построение графиков количества операций в зависимости от длины массива. Сравнение с теоретическими данными.

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
4	Реализация сортировки вставками и анализ её эффективности.	Алгоритм сортировки простыми вставками. Реализация функций поиска места вставки и функции операции сдвига массива. Анализ эффективности сортировки простыми вставками. Двоичный поиск. Рефакторинг простых вставок в двоичные. Измерение эффективности и сравнительный анализ.

Моделирование системы взаимодействующих частиц на примере сетевой игры

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Реализация однопользовательской и парной версий программы	Задача взаимодействия объектов «бактерий», реализация операций «поедания» и «деления».
2	Реализация парной версии с использованием сетевых файлов	Реализация взаимодействия программ по сети с помощью сетевых файлов, особенности работы с сетевыми файлами. Блокировка файлов. Проблемы синхронизации данных в приложениях.
3	Основные понятия компьютерных сетей	Технологии передачи данных в компьютерных сетях. Понятие сетевых пакетов. Архитектура ISO-OSI. Понятие адресов, их виды. Понятие сетевых протоколов. Понятие порта. Протокол TCP/IP, его свойства.
4	Реализация парной версии с использованием сетевых соединений	Понятие сетевого соединения. Библиотека txNetwork и работа с ней. Установление сетевого соединения и обмен данными по нему. Разработка протокола обмена данных.
5	Реализация сетевой библиотеки	Технология Windows Sockets. Инициализация приложения, получение информации об адресе, открытия и закрытия соединения, посылки и приема данных.

Моделирование представления знаний и реализация экспертных систем

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Простейшие структуры данных	Понятие структуры данных. Структура данных «Стек». Операции над стеком. Реализация надежной структуры данных, функции поддержки отладки и верификации.
2	Структура данных «Список»	Структуры данных «Односвязный список» и «двусвязный список». Реализация операций включения и исключения элементов списка. Ошибки в списках, операции верификации списков, функции поддержки отладки. Программа и язык dot, библиотека Dotter. Построение графических дампов списков.
3	Структура данных «Дерево»	Нелинейные структуры данных. Структура данных «Дерево». Рекурсивный характер деревьев. Применение деревьев. Операции по включению и исключению узлов деревьев. Обходы дерева и их виды. Запись дерева в файл и чтение из файла.
4	Введение в экспертные системы	Понятие об экспертных системах. Экспертные системы в виде двоичных справочников. Работа двоичного справочника, поиск существующего объекта и вставка нового. Построение определений объектов. Сравнение объектов.

Обработка данных физического эксперимента (фиттинг функций)

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Введение в свойства экспериментальных данных и методы их обработки	Получение данных физического эксперимента. Процесс измерения. Систематическая и случайная ошибки. Рабочий процесс обработки данных. Линеаризация зависимостей, причины ее применения, достоинства и недостатки.

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
2	Введение в фиттинг функций	Фиттинг функций как альтернативный метод обработки данных. Целевая функция (функция ошибки). Целевая функция как сумма квадратов отклонений, ее свойства. Методы перебора при фиттинге: метод Монте-Карло, идеи метода градиентного спуска.
3	Реализация программы фиттинга функций и его визуализации	Особенности реализации программы фиттинга. Отладка программ, особенности получения исходных данных. Реализация программы фиттинга.
4	Практическое применение метода фиттинга, обработка данных натурального эксперимента	Проведение натурального эксперимента по движению тела под действием силы тяжести (видеосъемка полета футбольного мяча), получение зависимости координат мяча от времени по видеозаписи. Обработка результатов эксперимента по движению тела под действием силы тяжести. Расчеты без учета силы аэродинамического сопротивления. Оценка результатов фиттинга и их достоверности. Учет аэродинамического сопротивления, изменение математической модели, перерасчет данных. оценка полученных результатов с точки зрения достоверности.

Начальное обучение программированию

<i>№</i>	<i>Тема</i>	<i>Содержание</i>
1	Введение в алгоритмическое решение задач	Задача и ее решение. Стадии решения задачи. Алгоритм как решение задачи. Способ записи алгоритмов с помощью псевдокода. Исполнитель алгоритма. Множество допустимых команд исполнителя, детализация алгоритма.

№	Тема	Содержание
2	Исполнители и алгоритмы	Практический пример решения задачи. Исполнитель «ТХ». Запись алгоритма в нескольких видах (псевдокод, блок-схема). Ошибки в алгоритме и способе его записи. Перевод алгоритма в программу (кодирование). Компьютер как исполнитель. Понятие о языке программирования.
3	Исполнитель «ТХ» и его свойства.	Основные возможности графической библиотеки ТХ Library («команды «ТХ»). Система координат экрана. Решение задач на составление алгоритмов и их запись в различных формах.
4	Практикум по программированию	Практика: проект «мультфильм» (линейные алгоритмы и их программирование). Работа в среде CodeBlocks. Элементарные операции в редакторе. Набор и отладка первой программы. Ошибки кодирования. Воспроизведение алгоритмических ошибок и ошибок кодирования.
5	Вспомогательные алгоритмы	Смысловое разделение частей алгоритма. Концепция процедурного программирования. Функции без параметров. Оформление функций. Вызов функции. Пошаговая отладка в программе с функциями. Именованное пространство. Понятие стиля программирования.
6	Практикум по программированию	Практика: рефакторинг проекта «мультфильм» с применением функций без параметров.
7	Вспомогательные алгоритмы с параметрами	Функции с параметрами, возможности, которые они предоставляют. Грамотное использование параметров функции (взаимная независимость). Числовой тип данных. Отладка программы с переменными и параметрами. Понятие о библиотеке функций.

№	Тема	Содержание
8	Практикум по программированию	Практика: рефакторинг проекта «мультфильм» с применением функций с параметрами.
9	Циклические алгоритмы	Повторяющиеся действия в алгоритмах. Разбор циклических алгоритмов. Цикл while. Работа с переменными в цикле. Отладка программы с циклами. Ошибки при работе с циклами.
10	Практикум по программированию	Практика: доработка проекта «мультфильм» с использованием циклических алгоритмов. Взаимодействие героев. Доработка мультфильма до компьютерной игры.

Материально-техническое обеспечение образовательного процесса

Для преподавания используется компьютерный класс, оснащённый современными персональными компьютерами **IBM PC** под **OS Windows** с возможностью высокоскоростного доступа в Интернет из расчёта 1 компьютерное рабочее место на 1 ученика, а также аналогичным компьютерным рабочим местом учителя с мультимедийным проектором и принтером.

В качестве программного инструментария для работы с языком C++ используются среды **Code::Blocks** (компилятор g++ 4.8.1 в среде MinGW) и **Microsoft Visual Studio 2013**. В качестве библиотеки программной поддержки для отображения двумерной графики используется бесплатная библиотека **TX Library**, разработанная автором курса. Для отображения трёхмерной графики использовались библиотеки **OpenGL** и **DirectX 11**.

Литература

1. Feldman, Michael B. Data structures with Ada. – PRENTICE-HALL, INC., Englewood Cliffs, New Jersey 07632.
2. Muchnick S. Advanced Compiler Design and Implementation. 2007.
3. Абрамов С. А., Гнездилова Г. Г., Капустина Е. Н., Селюн М. И. Задачи по программированию. М., 1998.
4. Александреску А. Современное проектирование на С++. М., 2003.
5. Аммерал Л. Машинная графика на персональных компьютерах. М., 2002.
6. Ахо А., Ульман Дж. Теория синтаксического анализа, перевода и компиляции. М., 1972.
7. Ахо А., Лам А., Сети Р., Ульман Дж., Компиляторы – принципы, технологии, инструментарий". М., 2013.
8. Баженова И. Ю. Visual С++. Уроки программирования. М., 2011.
9. Бар Р. Язык Ада в проектировании систем. М., 2008.
10. Барфилд Э., Уолтерс Б. Программирование «клиент-сервер» в локальных сетях. М., 2007.
11. Бейбер Р.А. Программное обеспечение без ошибок. М., 2006.
12. Бентли Дж., Жемчужины программирования. М., 1998.
13. Березин Б. И., Березин С. Б. Начальный курс С и С++. М., 2006.
14. Боресков А. В., Шикин Е. В., Шикина Г. Е. Компьютерная графика: первое знакомство. М., 2006.
15. Бочков С. О., Субботин Д. М. Язык программирования Си для персонального компьютера. М., 1990.
16. Братко И. Программирование на языке Пролог для искусственного интеллекта. М., 1990.
17. Брой М. Информатика. Вычислительные структуры и машинно-ориентированное программирование. М., 2008.
18. Брой М. Информатика. Структуры систем и системное программирование. М., 2006.
19. Брудно А. Л., Каплан Л. И. Московские олимпиады по программированию. М., 1990.
20. Брукс Р. Мифический человеко-месяц. М., 1998.
21. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++. М., 2012.
22. Вайнер Р., Пинсон Л. С++ изнутри. М., 2013.
23. Вандевурд Д., Джосаттис Н. "Шаблоны С++. Справочник разработчика. М., 2011.
24. Влассидес Дж. Применение шаблонов проектирования. Дополнительные штрихи. М., 2012.
25. Гамма Д., Хелм Р., Джонсон Р., Влассидес Дж, Приемы объектно-ориентированного проектиро-

- вания. Паттерны проектирования. М., 1999.
26. Графическая библиотека OpenGL, доступная по адресу <http://opengl.org>.
 27. Графическая библиотека TX Library с открытым исходным кодом, доступная по адресу <http://txlib.ru>.
 28. Грис Д. Конструирование компиляторов для цифровых вычислительных машин. М., 1982.
 29. Грэхем Дж., Кнут Д., Поташник М. Конкретная математика. М., 2010.
 30. Дагене В. А. и др. 100 задач по программированию. М., 1993.
 31. Дайитбегов В. А., Черноусов Е.А. «Основы алгоритмизации и алгоритмические языки». М., 1992.
 32. Доорс Дж. и др. ПРОЛОГ — язык программирования будущего. М., 2010.
 33. Зенкин А. А. Когнитивная компьютерная графика. М., 1991.
 34. Керниган Б., Ритчи Д. Язык программирования Си. М., 2012.
 35. Керниган Б., Пайк Дж. Практика программирования. М., 2001.
 36. Кнут Д. Искусство программирования для ЭВМ. Тома 1-4. М., 2001-2014.
 37. Колмогоров А. Н., Драгалин. А. Г. Введение в математическую логику. М., 1982.
 38. Коплиен Дж. Мультипарадигменное проектирование на C++. СПб, 2007.
 39. Коплиен Дж. Программирование на C++. СПб, 2005.
 40. Кормен Р., Лейсерсон Дж., Ривест Р. Алгоритмы, построение и анализ. М., 2008.
 41. Косарев В. И. 12 лекций по вычислительной математике (вводный курс) М., 2000.
 42. Костин А. Е., Шаньгин В. Ф. Организация и обработка структур данных в вычислительных системах. М., 1987.
 43. Котов Ю. В. Как рисует машина. М., 1988.
 44. Котов Ю. В., Павлова А. А. Основы машинной графики. М., 1993.
 45. Крокет Ф. МФС. Мастерская разработчика. М., 2008.
 46. Круглински Д. Основы Visual C++. М., 2007.
 47. Кук Г., Бейз Г. Компьютерная математика. М., 1990.
 48. Ласло М. Вычислительная геометрия и компьютерная графика на C++. М., 2007.
 49. Лафоре Дж. Объектно-ориентированное программирование в C++. М., 2014.
 50. Липаев В. В. Надежность программных средств. М., 2008.
 51. Лорьер Ж.-Л. Системы искусственного интеллекта. М., 1991.

52. Лукас П. С++ под рукой. М., 2003.
53. Лэнгсам Й., Огенстайн М., Тененбаум А. Структуры данных для персональных ЭВМ. М., 2009.
54. МакКоннел С. Совершенный код. СПб, 2010.
55. Малпас Дж. Реляционный язык Пролог и его применение. М., 1982.
56. Мартин М., Уэйт У., Прата С. Язык Си для начинающих. М., 1988.
57. Мейерс С. Эффективное использование С++. СПб, 2012.
58. Мика Б. и др. Практическое руководство по программированию: М., 2006.
59. Моль А. Теория информации и эстетическое восприятие. М., 1966.
60. Мюррей У., Паппас К. Создание переносимых приложений для Windows. СПб, 2007.
61. Нешков Г. и др. Множества. Отношения. Числа. Величины. М., 1978.
62. Нильсон Н. Искусственный интеллект. Методы поиска решений. М., 1973.
63. Новиков Ф.А. Дискретная математика для программистов. Учебник. СПб, 2001.
64. Парницкий Н. Основы статистической информатики. М., 1981.
65. Паронджанов В.Д. Как улучшить работу ума. (Новые средства для образного представления знаний, развития интеллекта и взаимопонимания) – М., 1998.
66. Прата С. Язык программирования С++. М., 2010.
67. Прата С. Язык программирования С. М., 2010.
68. Проектирование пользовательского интерфейса на персональных компьютерах. М, 2012.
69. Ракитин В. И., Первушин В. Е. Практическое руководство по методам вычислений с применением программ для персональных компьютеров. М., 1998.
70. Рассохин Д. От С к С++. М., 2013.
71. Рихтер Дж. Windows для профессионалов. М., 1995.
72. Рябенский В. С. Введение в вычислительную математику. М., 2000. Саттер Г. Решение сложных задач на С++. СПб, 2010.
73. Саттер Г. Новые сложные задачи на С++. СПб, 2014.
74. Саттер Г., Александреску А. "Стандарты программирования на С++"
75. Седов Л. И. Механика сплошной среды. М., 1972.
76. Страуструп Б. Дизайн и эволюция языка С++. СПб, 2005.

77. Страуструп Б. Язык программирования Си++. М., 2014.
78. Тихомирова Л. Ф., Басов А. В. Развитие логического мышления детей. М., 1995.
79. Тихонов А. Н., Самарский А. А. Уравнения математической физики. М., 1977.
80. Уилтон Р. Видеосистемы персональных компьютеров IBM PC и PS/2. Руководство по программированию. М., 1994.
81. Фаулер М. Рефакторинг. СПб, 2009.
82. Федоренко Р. П. Введение в вычислительную физику. М., 1994.
83. Филлипс Д., Гарсиа-Диас А. Методы анализа сетей. М., 1984.
84. Фролов А. В., Фролов Г. В. Графический интерфейс GDI в Microsoft Windows. М., 1994.
85. Фролов А. В., Фролов Г. В. Программирование видеоадаптеров. М., 1992.
86. Хантер Дж. Основные концепции компиляторов. М., 2006.
87. Хантер Дж. Трансляция с языков высокого уровня. М., 2012.
88. Хантер Дж., Томас У. Программист-прагматик. СПб, 2007.
89. Холзнер С. Visual C++ 6. Учебный курс. СПб, 2010.
90. Хэзфилд А., Кирби Дж. Искусство программирования на С. СПб, 2013.
91. Шень А.Х. Алгоритмика (учебник для математических классов). М., 1995.
92. Шикин Е.В., Боресков А.В., Зайцев А.А. Начала компьютерной графики. М., 2003.
93. Шикин Е.В., Плис А.И. Кривые и поверхности на экране компьютера. Руководство по сплайнам для пользователей. М., 1996.
94. Шилдт Г. Теория и практика С++. СПб., 2006.
95. Эккель Б. Философия С++. СПб, 2008.
96. Элти Дж., Кумбс М. Экспертные системы. Концепции и примеры. М., 1987.

