
Введение

Практикум реализован в виде крупных проектов к отдельным главам учебников И. А. Калинина, Н. Н. Самылкиной по информатике углубленного уровня для старшей школы (10–11 классы). Это позволяет обеспечить модульную структуру всего изучаемого материала и реализовать возможность работы по индивидуальному учебному плану. Проектная работа предусмотрена ФГОС и должна быть интегрирована в учебный процесс. Это особая форма организации деятельности обучающихся в рамках одного или нескольких учебных предметов. Она позволяет сформировать навыки самостоятельного применения приобретенных знаний и способов действий при решении различных задач прикладного характера. Предусматривается обязательная защита результатов выполнения индивидуального проекта. Поэтому достаточно крупные вычислительные задачи и практические задания, результаты которых необходимо объяснить, мы объединили в отдельные проекты. К выполнению проекта следует приступать после изучения всей главы учебника, поскольку он опирается на весь содержательный материал главы. Результаты выполнения заданий, объединенных в проект, демонстрируются и объясняются обучающимися. Это может быть тематический семинар, конференция или индивидуальная защита только перед учителем. Формат определяется исходя из потребностей участников образовательного процесса.

Первый проект (к главе 3 учебника 10 класса) посвящен имитационному моделированию в среде AnyLogic. Здесь рассматриваются актуальные проблемные задачи по всем основным подходам в имитационном моделировании: агентное моделирование, дискретно-событийное моделирование и модели системной динамики. Содержание дополняет задачи, разобранные в учебнике.

Общая цель выполнения данного проекта — освоение современного инструмента имитационного моделирования и демонстрация возможностей метода. Конкретные цели и задачи ставятся совместно учителем и обучающимися в зависимости от того, какие модели планируется исследовать. Можно исследовать их все либо выбрать модели из предложенных в практикуме задач, а также реализовать разобранные задачи из учебника в дополнение к задачам из практикума.

Для практикума одна из ведущих мировых компаний-разработчиков средств имитационного моделирования, компания XJ Technologies (AnyLogic), предоставляет специально адаптированную к условиям школьного курса версию среды AnyLogic, которая позволяет создавать, демонстрировать и исследовать широкий спектр моделей из самых разных областей практической деятельности. Использование этой среды позволяет не только теоретически обсудить важность и возможности методов моделирования, но и продемонстрировать их важность и возможности для решения практических задач на актуальную тематику. Программу имитационного моделирования AnyLogic вместе с лицензионным соглашением на ее использование можно скачать по адресу: <http://methodist.lbz.ru/authors/informatika/8/>.

Второй проект «Алгоритмы и программы», относящийся к главе 4, отличается от остальных тем, что этот материал можно изучать параллельно с изучением тем этой главы: «Структуры данных» и «Типовые алгоритмы». При изучении программирования предполагается, что школьники уже владеют первичными навыками составления алгоритмов и программ, предусмотренными стандартом основного общего образования. Для учащихся классов углубленного уровня уже не актуален методический прием, опирающийся на графическое изображение алгоритмической конструкции (блок-схема) для перехода к анализу реального алгоритма. По теме предусматривается развитие уже известного материала через изучение теоретических основ создания и оценки алгоритмов; рассматривается проблема алгоритмической неразрешимости и дается ряд эффективных решений для задач, которые используются впоследствии при изучении информационных технологий. Содержание работы в первую очередь ориентировано на формирование представления об основных подходах к хранению и обработке данных при разработке программ, их вводу (как с клавиатуры, так и из файла) и выводу; демонстрируется работа с основными структурами данных и базовыми алгоритмами сортировки и поиска. В дальнейшем навыки программирования станут основой решения других проектных задач в области информационных технологий, как и было заявлено авторами. В этом проекте подробно рассмотрены задания из учебника; непосредственно из практикума они могут быть реализованы в виде программного кода. Для контроля сформированности необходимых навыков программирования приводятся задания для самостоятельной отработки.

Третий проект «Обработка статистических данных» (к главе 5) можно выполнить независимо от изучения довольно сложного теоретического материала учебника. Необходимая теория интегрирована в процесс обсчета статистических данных, позволяющих определить качество используемого тестового материала в ходе

проведения, например, государственной итоговой аттестации выпускников школ. Обучающиеся получают доказательства того, что в основу тестовых процедур заложена научно обоснованная методика, используемая во всех странах. Самый ценный результат выполнения данного проекта для обучающегося — это способность к самостоятельной информационно-познавательной деятельности, включая умение ориентироваться в различных источниках информации, критически оценивать и интерпретировать информацию, получаемую из различных источников. Именно этот результат можно считать общей целью данного проекта. Используемый инструмент для расчетов — табличный процессор.

Четвертый проект «Технология обработки текста» (к главе 6) может быть двухсоставным. Первая его часть может включать традиционную работу по подготовке текста по выбранной тематике со сложным форматированием. Объем и сложность подготовки оригинал-макета издания зависят от его типа (буклет, реферат, брошюра, плакат и пр.). Предполагается, что это межпредметная проектная работа. Задание из учебника уточняется учителем. Возможно, что такая работа окажется слишком простой для обучающихся, поэтому ее можно пропустить или предложить избирательно.

Вторая составляющая этого проекта более интересна и обязательна для выполнения. Общей целью работы является использование регулярных выражений для анализа текста. До тех пор, пока текст невелик, его можно проанализировать без применения техники, прочитав и обдумав. Но как только текст становится большим (особенно если он разбит на множество фрагментов), его анализ становится как минимум затруднительным — на чтение и обдумывание нужно немало времени.

Если же к большому объему текста добавляются сжатые сроки обработки, то чистый «ручной» анализ становится невозможным (или слишком дорогим) и появляется необходимость автоматизировать его обработку.

Анализ текста на естественном языке — многогранная задача. Различные методы и средства анализа разработаны для многих узких областей — это задачи поиска, перевода, выявления закономерностей и т. д. Мы продемонстрируем несколько специализированных методов автоматической обработки текста, которые в разных формах применяются очень широко.

Для выполнения заданий используется редактор Expresso, основанный на использовании механизма регулярных выражений среды .Net. Один из существенных плюсов такого рода средств — возможность отслеживать исполнение выражения, результаты его применения к тексту, создавать библиотеки готовых шаблонов.

Следующие три проекта относятся к 11 классу.

Проект «Графика и визуализация» к главе 1 состоит из двух самостоятельных частей. Использовать их можно как совместно, так и по отдельности.

В первой части рассматривается графика на плоскости. Приведены задания по темам, рассмотренным в учебнике: использование цветowych моделей, графические фильтры, реализация графических примитивов и геометрические преобразования фигур на плоскости. Для реализации используется программная среда PascalABC.NET, использованная ранее, и стандартные функции Microsoft.NET. Задания позволяют изучить базовые алгоритмы обработки графической информации и выявить основные проблемы, возникающие при этом.

Вторая часть проекта посвящена трехмерному моделированию. Для освоения основ работы с трехмерными моделями реальных объектов используется свободно распространяемая программа Google Sketchup версии 8 (возможно использование и более современных версий — без изменения содержания). Эта программа предназначена для так называемого «эскизного» моделирования: в ней нельзя создать действительно сложную модель или фотореалистичное изображение (тем более анимированное), зато удобно использовать для подготовки модели здания, предмета мебели, посуды и т. п. Именно эти модели рассматриваются в работе.

В проекте «Обработка звукового файла» к главе 2 используется третья версия продукта. Общей целью данного проекта — знакомство с основными понятиями музыкальной звукорежиссуры и основными приемами обработки звука на компьютере. Работа будет интересна многим старшеклассникам в основном из-за творческой составляющей и актуальности этой деятельности.

Проект «Защита данных в сетях» к главе 5 посвящен защите данных в сетях и основам криптографии. Задачи, рассматриваемые в рамках этой части практикума, предназначены, во-первых, для изучения фактического порядка функционирования сетевых компонентов и, во-вторых, для ознакомления с проблематикой защиты данных в сетях. Именно это можно считать общей целью выполнения этого проекта.

Приведенные задания требуют подготовки и использования виртуальной машины, на которой и отрабатываются практические действия по защите персональной машины. Рассматриваются средства обновления системы, установки и настройки межсетевого экрана, средства работы с сертификатами и шифрованием. Все практические работы связаны с непосредственной демонстрацией того, что бывает при отсутствии средств защиты.

Желаем успехов!

10 класс

- Проект «Моделирование» к главе 3 учебника
- Проект «Алгоритмы и программы» к главе 4 учебника
- Проект «Обработка статистических данных» к главе 5 учебника
- Проект «Технология обработки текста» к главе 6 учебника

Имитационное моделирование

Как вам уже известно из учебника, имитационное моделирование — мощнейший метод исследования самых разных проблем, в первую очередь — проблем чисто практических. Чтобы освоить его основные виды и познакомиться с его возможностями, мы построим несколько моделей с использованием среды AnyLogic. Российская компания The AnyLogic Company, производитель этой среды, любезно предоставила школьникам специальную версию, которой более чем достаточно для выполнения практических работ. Перед началом работы кратко опишем интерфейс среды.

Основное рабочее окно среды выглядит примерно так (рис. 1).

С левой стороны вверху — дерево, в котором показаны открытые сейчас модели (на иллюстрации — моделей две, открыта Epidemic), а в модели — объекты, из которых она состоит. В открытой сейчас модели мы видим основное пространство — Main, параметры и стартовую страницу модели Simulation, на которой можно будет размещать настройки перед прогонами.

Среда позволяет моделировать самые разные процессы через взаимодействие *объектов* — программных моделей предметов, явлений и связей. Каждая такая модель-объект включает в себя данные-описания (свойства) и программы-действия.

Наша цель при создании модели реального процесса — собрать комплекс объектов, который будет вести себя так же, как моделируемая нами действительность, конечно, в той мере, в какой наша модель вообще может это делать (напомним, что модель — *упрощенное* подобие действительности). Собрать комплекс мы можем, разместив объекты в рабочей зоне и связав их явно, указав, например, траекторию перехода объектов, записав действие-программу, вычислить значение параметра и т. д. Практически все наши описания подготовки моделей — это описание того, из каких объектов мы собираем модель, как их соединяем и какие значения свойств и действий придаем. Перед запуском модели среда переведет наш комплекс объектов в исполняемый код и (если это удастся) запустит.

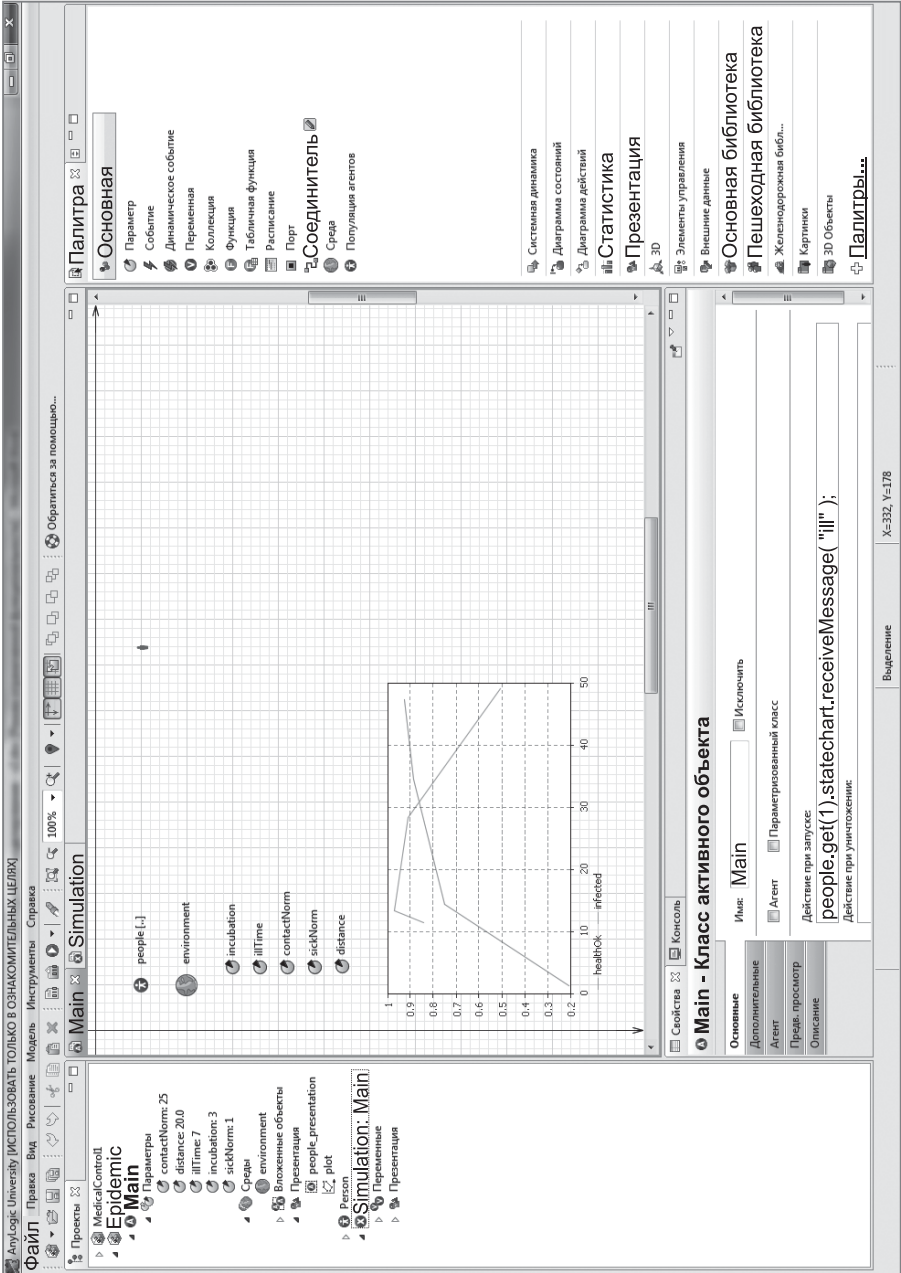


Рис. 1. Основное рабочее окно среды AnyLogic

Основу языка программирования, на котором мы фактически описываем объекты в среде, составляет язык программирования Java. Его синтаксис, как и описание объектов, описан в разделе помощи к среде, которой мы очень рекомендуем пользоваться во время работы. Из языков, на которые мы опираемся при изучении информатики, ближе всех к нему язык С. Несмотря на это, среда построена так, что писать как таковые программы нам фактически не придется, поскольку там пошагово описано и создание моделей-примеров.

Возвращаясь к изображению основного рабочего окна системы, отметим, что объекты размещаются на общем поле — в графическом редакторе. Объекты можно добавлять, перетаскивая их из палитр наборов объектов в правой части (сейчас открыта палитра **Основная**). Свойства и методы объектов можно описывать, задавая им значения в нижней центральной части рабочего окна.

Начнем с наиболее иллюстративного вида имитационных моделей — агентных моделей.

Задача 1. *Изучение движения учащихся через турникеты с помощью агентной модели*

В школе № 0 планируется повысить уровень обеспечения безопасности. В первую очередь специалисты рекомендовали ограничить проход в школу посторонних. Для этого на входе установят два турникета и будут проверять всех входящих — требовать предъявления специальной карты.

У директора школы есть подозрение, что применение такого подхода приведет к существенным затруднениям перед началом занятий.

Как подтвердить или проверить эти предположения?

Решение

Создадим модель, изображающую первый этаж школы с турникетами и агентами-пешеходами, единственная задача которых — пройти через турникет.

При этом мы учтем, что проход через турникет (а точнее, проверка) занимает некоторое время, и время от времени будет попадаться человек без документов (например, без карточки), которого нужно возвращать.

Время прохода может меняться (человек проверяет документы медленнее, чем автомат — карточки), вероятность появления «неправильного» документа — тоже.

Чтобы модель выглядела реалистично, найдем на сайте авторской мастерской (<http://methodist.lbz.ru/authors/informatika/8/>) в разделе «Электронное приложение» типовой проект школы и загрузим план первого этажа.

Далее запустим среду и командой **Файл/Создать** создадим пустую модель (рис. 2):

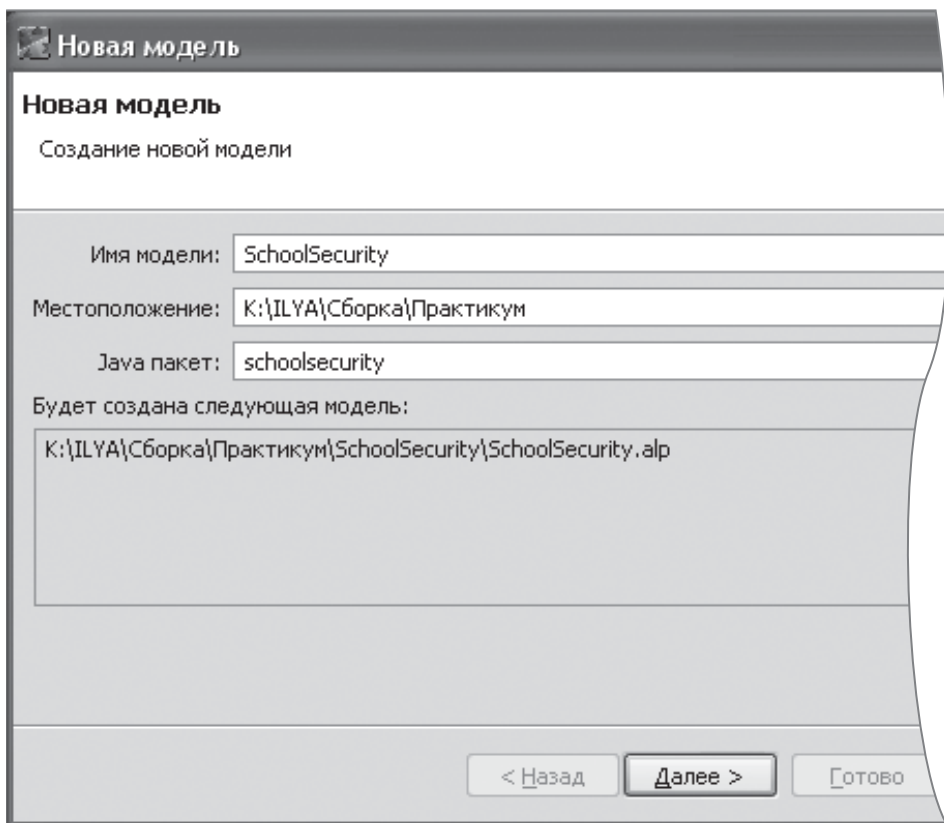


Рис. 2. Создание пустой модели

На втором шаге укажем, что модель создаем «с нуля», не используя заготовки.

План первого этажа очистим от лишних деталей и разместим на рабочем листе: из набора компонентов **Презентация** перетащим на рабочее поле объект **Изображение** и добавим в его параметрах картинку плана этажа (рис. 3).

Вставленную картинку «растянем».

Теперь прямо поверх плана отметим стены (поскольку на картинке система не может автоматически распознать их). Для этого построим ломаные вдоль стен. Обратите, пожалуйста, внимание:

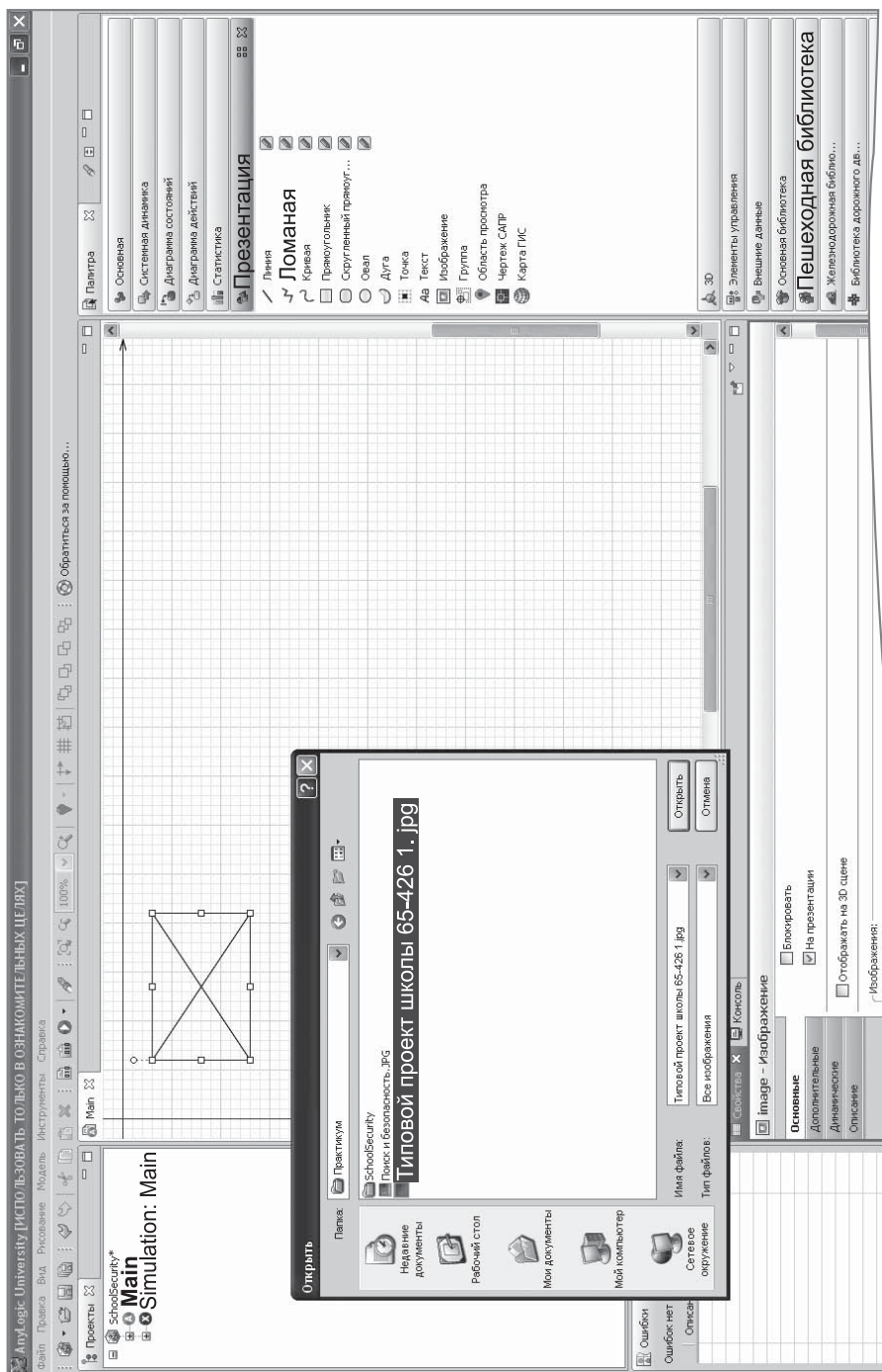


Рис. 3. Создание плана первого этажа

[. . .]