

КУРС «ВВЕДЕНИЕ В ПРОФЕССИОНАЛЬНУЮ ДЕЯТЕЛЬНОСТЬ», или как вдохнуть жизнь в то, к чему прикоснулся чиновник*

Сениченков Ю. Б.

Санкт-Петербургский политехнический университет Петра Великого, Санкт-Петербург, Россия

Аннотация

В статье представлен вариант учебного курса «Введение в профессиональную деятельность»¹ для студентов первокурсников, будущих специалистов в области информационных технологий. Студентам предлагается с помощью отечественной среды визуального моделирования Rand Model Designer разработать компьютерную игру, пройдя все этапы создания программного обеспечения. Обсуждаются возможности использования системы поддержки учебных курсов SAKAI.

Ключевые слова: обучение математическому и компьютерному моделированию, объектно-ориентированное моделирование, технологии компьютерного моделирования, компьютерное моделирование для начинающих.

Цитирование: Сениченков Ю. Б. Курс «Введение в профессиональную деятельность», или как вдохнуть жизнь в то, к чему прикоснулся чиновник // Компьютерные инструменты в образовании. 2018. № 6. С. 16–38. doi:10.32603/2071-2340-2018-6-16-38

РАЗДРАЖЕННОЕ ВСТУПЛЕНИЕ²

Представляю себе, как радовался инноватор-чиновник, предложивший включить в учебные планы обязательные для всех дисциплины «Введение в профессиональную деятельность», «История и методология науки», «Основы инженерной деятельности», «Производственная практика», и после многих лет борьбы с ретроgrадами от образования, предупреждавшими, что от этой «инновации» ничего хорошего не будет, добившийся своего.

*Поддержка Европейской комиссией выпуска этой публикации не означает одобрения содержания, отражающего взгляды только авторов, и Комиссия не может нести ответственность за любое использование информации, содержащейся в ней.

¹Курс был разработан для проекта InMotion: «Новые стратегии обучения инженеров с использованием сред визуального моделирования и открытых учебных платформ» <http://www.inmotion-project.net/index.php/gu/>.

²Это реакция на стресс, пережитый в процессе подготовки к аккредитации. Те, кто избежал подобных стрессов, могут сразу перейти к Введению.

Возьмем, к примеру, «Производственную практику». Еще в советские времена, с плановой экономикой и возможностью заменить инициативу приказом, организация производственной практики вызывала трудности. На кафедре, где я учился, было четыре студенческих группы, около шестидесяти человек. Все мы учились по специальности «ЭВМ». Куда нас было отправить на практику и чем занять? Выход нашелся. На рижском заводе ВЭФ собирали конвейерным способом знаменитые приемники ВЭФ. При чем здесь ЭВМ? А при том, что кафедра была на Радио-Физическом Факультете (РФФ). Следовательно, студенты радио-физического факультета знакомились с «передовыми» производственными технологиями. Был ли заинтересован ВЭФ в нас, и чему нас собирались научить? Был заинтересован и еще как, так как практику проводили летом, в желанный для всех период отпусков, и можно было дать отпуск шестидесяти работникам ВЭФ сразу! Чему я научился? Моя операция на конвейере сводилась к завинчиванию гайки и покрытию ее лаком: окунул кисточку в лак и мазнул им прикрученную гайку. И делал я это через каждые две минуты в течение всего рабочего дня, когда ко мне подползал очередной полусобраный приемник. После окончания практики я твердо знал, что ноги моей не будет ни на каком передовом производстве, но я навсегда полюбил старую Ригу, где мы проводили все свободное время. Однако чиновники были довольны! В моем отчете была отражена история завода, приведена схема приемника, выделена жирно гайка и обоснована эффективность конвейерного производства. В нынешние времена, когда практически все студенты работают, организовать практику стало намного легче — студент приносит справку с предприятия, где он работает, и пишет отчет о своей гайке. Душа чиновника должна радоваться!

От воспоминаний перейдем к курсу «Введение в профессиональную деятельность» для студентов первокурсников (осенний семестр).

Чему должен быть посвящен этот курс? Чему мы хотим научить студента? Ответ был бы простым — надо рассказать об особенностях профессии, предстоящей учебе и будущем трудоустройстве, если бы не матрица компетенций учебного плана (табл. 1).

Таблица 1. Фрагмент матрицы компетенций

УК-1	Способен осуществлять поиск, критический анализ информации, применять системный подход для решения поставленных задач
УК-2	Способен определять круг задач в рамках поставленной цели и выбирать оптимальные способы их решения, исходя из действующих правовых норм и имеющихся ресурсов и ограничений
УК-3	Способен осуществлять социальное взаимодействие и реализовать свою роль в команде
УК-6	Способен управлять своим временем, выстраивать и реализовывать траекторию саморазвития на основе принципов образования в течение всей жизни
ОПК-2	Способен применять компьютерные / суперкомпьютерные методы, современное программное обеспечение, в том числе отечественного происхождения, для решения задач профессиональной деятельности
...	...

Решение задачи, чему и как учить, требуется еще и перевести на язык чиновника, используя универсальные, профессиональные и прочие компетенции. Справедливости ради, надо сказать, что чиновники «компетенции не пишут». Это мы сами, стараясь им угодить, пишем на никому не понятном языке непонятно о чем.

Мало того, нужно придумать, как проверять полученные знания — например, с помощью тестов (десять вопросов на каждую компетенцию с четырьмя возможными ответами!), если выбрать тест средством контроля.

Попробуйте составить рабочую программу этого курса.

Дано:

Предположим, что вы выбираете понятную и простую схему организации этого курса: выделяете каждому приглашенному ведущему преподавателю или руководителю предприятия по два часа и просите рассказать о своем предмете или производственной, научной или любой другой профессиональной деятельности. Так большинство кафедр и поступает. В результате, скорее всего, это будет каша, сумбур, рассказы о себе любимом, но, возможно, из этих фрагментов сложится образ будущей профессиональной деятельности. И слава Богу!

Требуется:

Составьте рабочий план дисциплины по сегодняшним требованиям для этого сценария так, чтобы и проверяющий планы чиновник не придирался бы, и вам не было бы стыдно.

1. ВВЕДЕНИЕ

Попробуем разработать содержательную часть рабочей программы дисциплины «Введение в профессиональную деятельность» для студентов первого курса, изучающих информационные технологии.

Студент первого курса в осеннем семестре обладает только школьными знаниями, еще одержим желанием учиться и потому чрезвычайно загружен, старателен, но не всегда понимает, чего от него хотят преподаватели, и практически всегда еще не умеет планировать свою работу, что чаще всего приводит к стрессу. Курсы по математике, физике, алгоритмизации — базисные, необходимые и трудоемкие для него дисциплины, если он хочет стать настоящим профессионалом. На их фоне дисциплина «Введение в профессиональную деятельность» может быть увлекательной игрой (нужных знаний и умений еще нет), обязательно заканчивающейся успехом (стрессов и так хватает). Игра — это активная деятельность, в отличие от пассивных лекций о будущей профессии. Еще лучше, если это будет коллективная деятельность — наиболее распространённая форма будущей профессиональной деятельности. Коллективная деятельность и коллективная ответственность за конечный продукт обычно приводят к тому, что каждый в студенческом коллективе из практически незнакомых людей в конце концов выбирает себе (или ему отводят) роль по его силам, и происходит это достаточно мягко, без обид. И пусть в таком коллективе, успешно справившемся с заданием и получившем «бригадный» зачет, будут слабые игроки. Надо просто закрыть на это глаза, и тем самым помочь слабым преодолеть трудный психологический рубеж — первый учебный год в университете.

Попытаемся смоделировать поэтапную, коллективную работу над законченным программным продуктом в течение одного семестра: от формулировки задачи заказчиком (преподавателем) до приемки им же созданного программного обеспечения и оплаты (бригадный «зачет» — «незачет») проделанной работы.

Разрабатываемый программный продукт может быть компьютерной игрой, созданной в конкретной, сильной (известной или интуитивно понятной) практически еще школьнику программной среде. Компьютерную игру хорошо бы снабдить 2D- или 3D-образами, и если среда позволяет это сделать, то создание игры можно разделить на этапы, приведенные в таблице 2.

Таблица 2. Этапы создания игры

1	Неформальное описание будущей игры	Краткое описание возможных игр (преподаватель), организация коллективов разработчиков, распределение игр между коллективами (студенты)
2	Формализация игры. Математическая модель	Представление будущих игр коллективами. Разработка технического задания. Построение математической модели
3	Алгоритмическая модель	Алгоритмическое описание будущей игры
4	Программная реализация игры	Программная реализация игры на заданном языке с визуализацией поведения в виде графиков. Проверка работоспособности на тестах
5	Визуализация поведения (2D- и 3D- анимация)	Выбор приборов (2D-анимация) для задания конкретных значений параметров игры, выбор 3D-образов для визуализации поведения
6	Тестирование игры	Создание тестов — примеров успешного и неуспешного выбора параметров для достижения цели игры
7	Создание презентации и выступление на «приемочной комиссии» (преподаватель, студенты)	Создание презентации в среде PowerPoint и выступление перед студенческой аудиторией
8	Письменный отчет о проделанной работе	Написание отчета (лучше в форме статьи в соответствии с требованием журнала КИО)

2. ВЫБОР ИГРЫ

Компьютерная игра может быть моделью таких игр, как футбол, баскетбол, бильярд, и тогда — это динамические системы с хорошо известными школьнику уравнениями движения. Построение математической модели не должно вызывать затруднений.

Пример.

А. Простая игра. Разработайте игру, в которой требуется попасть мячом в корзину, находясь от нее на заданном расстоянии и выбирая начальную скорость мяча и угол бросаемого вами мяча. Игроку дается возможность сделать три попытки.

Б. Игра средней сложности. К условиям простой игры добавьте дополнительное препятствие – игрока другой команды, стоящего перед щитом на заданном расстоянии.

В. Сложная игра. Рассмотрите возможность попадания мяча в корзину от щита.

В нашем случае математическая модель — это уравнения движения тела, брошенного под углом к горизонту. Любой школьник пишет их, не задумываясь, и без ошибок!

Используя только эти уравнения, можно придумать достаточно большое число индивидуальных заданий различной сложности (табл. 3).

В приведенных играх мы находим примеры:

- дискретных динамических систем (поднимаемся по лестнице),
- непрерывных динамических систем (съезжаем с горы),
- событийно-управляемых динамических систем (парашютист по команде летчика совершает прыжок).

Заметим, что парашютист и самолет — это еще и многокомпонентная динамическая система!

Сама игра — это последовательность попыток с выбором (это школьная задача по физике, и она может быть решена заранее) или угадыванием значений параметров (игра

Таблица 3. Задания

№	Название	Сложность	Неформальное описание
1	Баскетбол	Простая	Выбрать начальную скорость и угол так, чтобы брошенный мяч попал в корзину. Координаты корзины, ее размеры, высоту игрока выбрать самостоятельно
		Средней сложности	Добавьте в качестве препятствия игрока другой команды, находящегося на заданном расстоянии от бросающего
		Сложная	Рассмотрите две дополнительные возможности попадания мяча в корзину: с отскоком от щита и от пола
2	Хоккей	Простая	Игрок находится на поле в точке с координатами (x, y) и пытается попасть шайбой в ворота. Начальная скорость шайбы и положение ворот заданы. Выберите правильный угол бросания*
		Средней сложности	Добавьте в качестве препятствия стенку из игроков другой команды, находящихся на заданном расстоянии от бросающего. Для того чтобы попасть в ворота, стенку нужно «перебросить»
		Сложная	Рассмотрите дополнительную возможность попадания шайбы в ворота от борта
3	Горка и бассейн	Простая	Мальчик поднимается по ступенькам горки «дискретно» — за каждый временной такт он мгновенно поднимается на следующую ступеньку. Поднявшись на вершину горки, он начинает съезжать вниз с постоянной скоростью V . Наклонная плоскость горки расположена под углом 45 градусов к горизонту и заканчивается трамплином с заданным углом наклона. Выберите скорость V и угол наклона трамплина так, чтобы мальчик приземлился на заданном расстоянии от трамплина*
		Средней сложности	Мальчик начинает съезжать с горки только по команде «Поехал!»
		Сложная	Сразу же за трамплином расположите бассейн: если начальные скорость и угол трамплина выбраны неудачно, то мальчик попадет в бассейн и ему придется плыть до края бассейна
4	Две горки	Простая	Мальчик поднимается по ступенькам левой горки «дискретно» — за каждый временной такт он мгновенно поднимается на следующую ступеньку. Поднявшись на вершину горки, он начинает спускаться по второй, правой, горке также дискретно, пока не окажется на земле*
		Средней сложности	Левая и правая горка расположены на расстоянии L друг от друга. Для того чтобы попасть с левой горки на правую, нужно совершить прыжок по законам тела, брошенного под углом к горизонту. Выберите скорость и угол прыжка так, чтобы попасть на верхнюю площадку второй горки и дискретно спуститься вниз
		Сложная	Прыжок с горки на горку совершать только по команде «Прыгай!»
5	Самолет и парашютист	Простая	Самолет с парашютистом летит на заданной высоте с постоянной скоростью. В некоторый момент времени T парашютист выпрыгивает из самолета и начинает спускаться с постоянной вертикальной скоростью V . Выберите скорость V так, чтобы парашютист приземлился не дальше, чем на расстоянии L_{MAX} от начальной точки полета*

№	Название	Сложность	Неформальное описание
5	Самолет и парашутист	Средней сложности	Прыжок парашютиста начинается по команде «Прыгай!»
		Сложная	Выбрать момент прыжка и скорость V так, чтобы парашютист приземлился в заданных границах $[L_{MIN}, L_{MAX}]$
6	Самолет и ракета	Простая	Самолет взлетает под углом Angle с постоянной скоростью V . Достигнув заданной высоты, продолжает горизонтальный полет. В аэропорту в этот момент узнают, что экипаж забыл завтраки для пассажиров, и решают доставить их с помощью ракеты. С какой скоростью и под каким углом должна вылететь ракета, чтобы доставить завтраки на самолет*
		Средней сложности	Вылет ракеты осуществлять по команде «Пуск!»
		Сложная	Доставив завтраки, ракета спускается на землю, а самолет продолжает свой путь
7	Шары на прямой	Простая	Три шара с массами M_1, m, M_2 расположены на одной прямой в указанном порядке и могут двигаться только по ней. Начальные скорости шаров M_1, M_2 равны нулю. Скорость шара с самой маленькой массой m равна V . Шар с массой m в начальный момент начинает двигаться случайным образом влево или направо и при каждом ударе с соседями передает им часть своей энергии, меняя свою скорость по знаку на противоположную. Предсказать (угадать), сколько соударений будет между шарами за время T при заданной скорости V *
		Средней сложности	Предсказать (угадать) время T при заданной скорости V , при котором шары перестанут сталкиваться
		Сложная	Прямоугольник одной стороной вертикально поставили на землю, а на параллельной стороне расположили шары с массами M_1, m, M_2 . Шар с массой m в начальный момент начинает двигаться случайным образом влево или направо. Игра заканчивается, когда один из шаров M_1, M_2 упадет с прямоугольника и коснется земли. Выбрать скорость V , высоту и длину прямоугольника, чтобы время игры было бы не меньше T
8	Биллиард	Простая	Борта хоккейного поля имеют форму треугольника. Ворота, отрезок заданной длины, расположены внутри площадки и не касаются бортов. В начальный момент шайба находится в точке (x, y) . Задав начальную скорость V и направление движения шайбы, послать шайбу в ворота, но так, чтобы было хотя бы одно отражение от бортов V *
		Средней сложности	Борта хоккейного поля имеют форму круга. Центр отрезка-ворот совпадает с центром круга
		Сложная	Борта хоккейного поля имеют форму эллипса. Отрезок-ворота расположен на линии, соединяющий фокусы эллипса

* Параметры игры выбираются самостоятельно, согласуются с преподавателем и включаются в техническое задание.

для детей и значения придется подбирать по ходу игры), воспроизведением конкретной траектории движения (с помощью графиков, но лучше с помощью графических образов мяча, шайбы, самолета, ракеты) и фиксации результатов каждой попытки с помощью опять же графических образов (красный квадрат — неудача, зеленый — успех).

- При таком выборе игр мы можем дополнительно поговорить со студентами о
- математическом моделировании,
 - компьютерном моделировании,
 - алгоритмических языках и языках моделирования,
 - средах моделирования,
 - компьютерной графике.

3. ВЫБОР СРЕДЫ РАЗРАБОТКИ

В обсуждаемом курсе компьютерная игра должна быть реализована в среде визуального моделирования Rand Model Designer (www.mvstudium.com).

Обосновать этот выбор можно по-разному.

Во-первых, эта среда визуального моделирования разработана и сопровождается сейчас в том числе и сотрудниками, и аспирантами, и студентами нашего политехнического университета Петра Великого. И этого «довода» часто более чем достаточно для выбора программного продукта для учебных целей.

Во-вторых, что существеннее, — это среда, использующая технологию Объектно-Ориентированного Моделирования, может быть названа «условно свободно-распространяемой» — на сайте www.mvstudium.com есть ссылка на бесплатную версию среды. Условно-бесплатная версия будет работать в течение года, что вполне достаточно для учебного процесса, и ее можно обновлять, а только затем она начнет назойливо предлагать купить себя за небольшие деньги.

Сравнительный анализ [1] языков моделирования MVL (язык среды Rand Model Designer) и Modelica (<https://www.modelica.org/>) (его использует открытая среда OpenModelica и коммерческая Dymola) показал, что языки близки, и, зная один, легко перейти на другой. Более того, можно достаточно быстро написать конвертер с языка Modelica на язык MVL. Прототип конвертера создан студентами нашего университета. Сравнительный анализ цен профессиональных сред Rand Model Designer и Dymola говорит в пользу среды Rand Model Designer (дополненной даже студенческим конвертером).

Среда моделирования Rand Model Designer содержит Редактор моделей и Визуализатор поведения (компонент для отладки и выполнения моделей под управлением среды, анализа и обработки результатов вычислительных экспериментов).

Редактор моделей позволяет создавать модели непосредственно на языке моделирования MVL (текстовый режим), как если бы вы делали, используя среду OpenModelica, и с помощью редактора, содержащего графические компоненты — редакторы уравнений, карт поведения, компонентных моделей (графический режим).

Языки Modelica и MVL практически идентичны при описании однокомпонентных непрерывных динамических систем, компонентных систем с входами-выходами и контактами-потоками, но отличаются для событийно-управляемых (гибридных) систем и систем переменной структуры.

Отмечаемым всеми достоинством языка MVL является использование модификации машин состояния UML — карт поведения. Карты поведения допускают непрерывные DO_активности, представленные либо явно заданными функциями от времени, либо решениями алгебро-дифференциальных уравнений, то есть опять же функциями, зависящими от времени. В нашем случае такая модификация машин состояний чрезвычайно важна, так как позволяет просто и наглядно описать поведения событийно-управляемых динамических систем: дискретных, непрерывных, гибридных.

Дискретные и гибридные системы создаются с помощью редактора карт поведения, непрерывные — с помощью редактора уравнений. Следовательно, все приведенные примеры игр можно создавать с помощью графического языка высокого уровня. Разработанные модели доступны и в текстовой форме, что ускоряет изучение языка моделирования: создайте правильно работающую модель с помощью графического языка и посмотрите (отредактируйте) ее представление в текстовой форме.

Для понимания, как работает модель с непрерывным поведением, студенту достаточно знать, что для воспроизведения поведения непрерывных динамических систем, представленных как уравнениями, так и формулами, исполняемой модели (автоматически созданному средой программному коду) требуется уметь решать написанные студентом уравнения в заданный момент времени (вычислять значения формул) на промежутке $[0, T]$ и «двигать» модельное время, то есть последовательно выбирать точки t_1, t_2, \dots, t_n промежутка $[0, T]$ от начальной до конечной настолько близко друг от друга, чтобы построенное решение воспринималось как непрерывное.

Исполняемая модель с дискретным временем также автоматически двигает модельное время, но теперь уже дискретно, с заранее заданным шагом по времени, при этом значения интересующих нас переменных в очередной точке определяются с помощью алгоритмов, записанных на языке моделирования.

Двигатель гибридного времени продвигает как непрерывное время, так и дискретное, необходимое для упорядочивания событий во временных щелях [2].

И, наконец, гибридная или событийно-управляемая динамическая система меняет одно свое локальное непрерывное поведение на другое в момент наступления события. Смена непрерывных поведений может сопровождаться «мгновенными» действиями: действиями, длительностью которых можно пренебречь, по сравнению с длительностью непрерывных действий (активностей, поведений). Мгновенные действия либо завершают непрерывное поведение, либо предваряют его, либо сопутствуют произошедшему событию [2].

Событие становится еще одним важным ключевым понятием, которое нужно подробно обсудить с начинающими.

Понятия гибридной системы, длительного непрерывного поведения, мгновенного действия, события в этом курсе можно ввести неформально (рис. 1, 2).

<p>* «Из дома вышел человек, * С дубинкой и мешком, * И в дальний путь, * И в дальний путь, * Отправился пешком.</p> <p>* Он шел все прямо и вперед * И все вперед глядел. * Не спал, не пил, * Не пил, не спал, * Не спал, не пил, не ел.</p> <p>* И вот однажды на заре * Вошел он в темный лес, * И с той поры, и с той поры, * И с той поры, и с той поры, * И с той поры исчез.</p> <p>* Но если как-нибудь его * Случится встретить вам, * Тогда скорей, * Тогда скорей, * Скорей скажите нам. >></p> <p>* Стихотворение Хармса</p>	<p>* Начало отсчета времени: * «Из дома вышел человек» * Длительное действие: «Он шел все прямо и вперед» * Событие: * «И вот однажды на заре * Вошел он в темный лес» * Длительное действие: «И с той поры исчез» * Событие: * Но если как-нибудь его * Случится встретить вам * Мгновенное действие: * Скорей скажите нам * Конец</p> <p>* Описание в терминах действий и событий</p>
--	---

Рис. 1. Фрагменты презентации для лекции о языке моделирования MVL — события и непрерывные действия

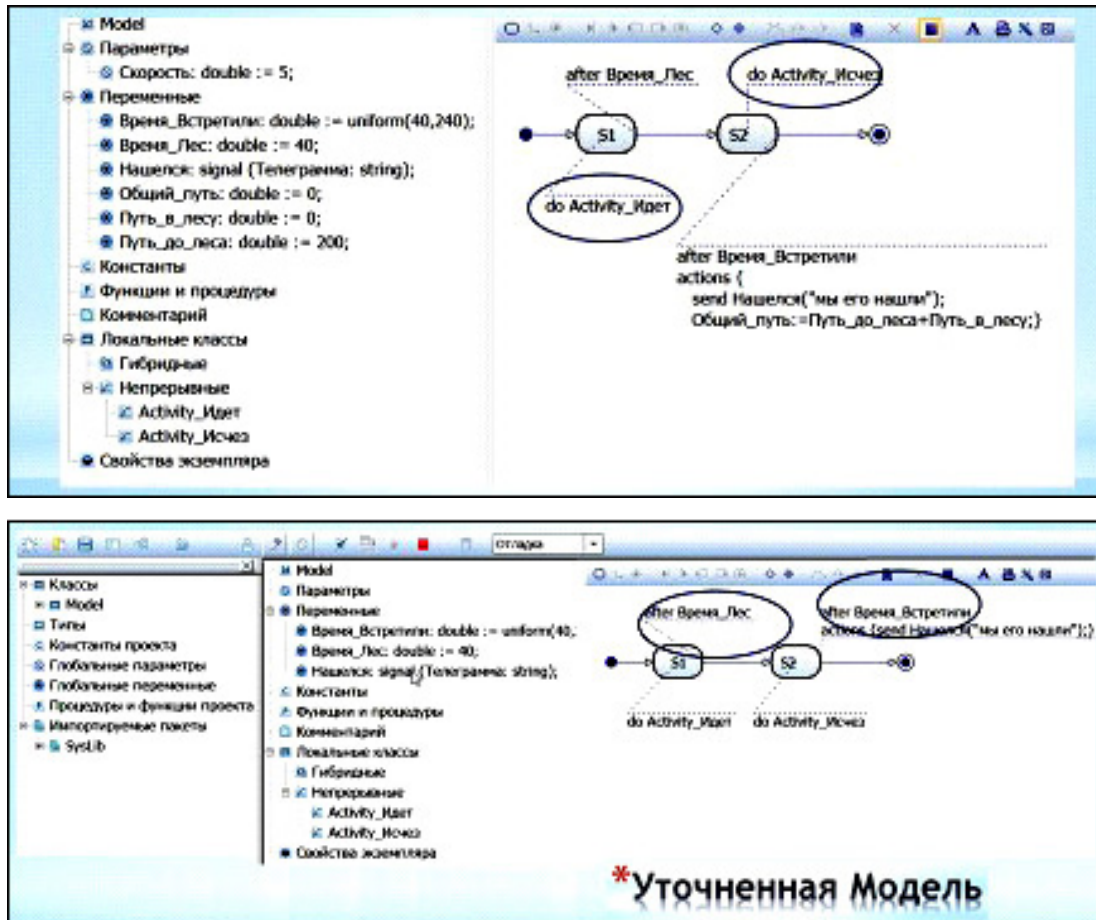


Рис. 2. Фрагменты презентации для лекции о языке моделирования MVL. Стихотворение Хармса на языке карт поведения

С помощью карт поведения можно моделировать и поведение дискретных систем (рис. 3, 4).



Рис. 3. Фрагменты презентации для лекции о циклах, событиях и дискретных динамических системах

Визуализатор поведения, он же — отладчик моделей, он же — среда для проведения вычислительных экспериментов — позволяет запустить исполняемую модель, управлять ее поведением, визуализировать результаты в различных формах и обрабатывать результаты экспериментов.

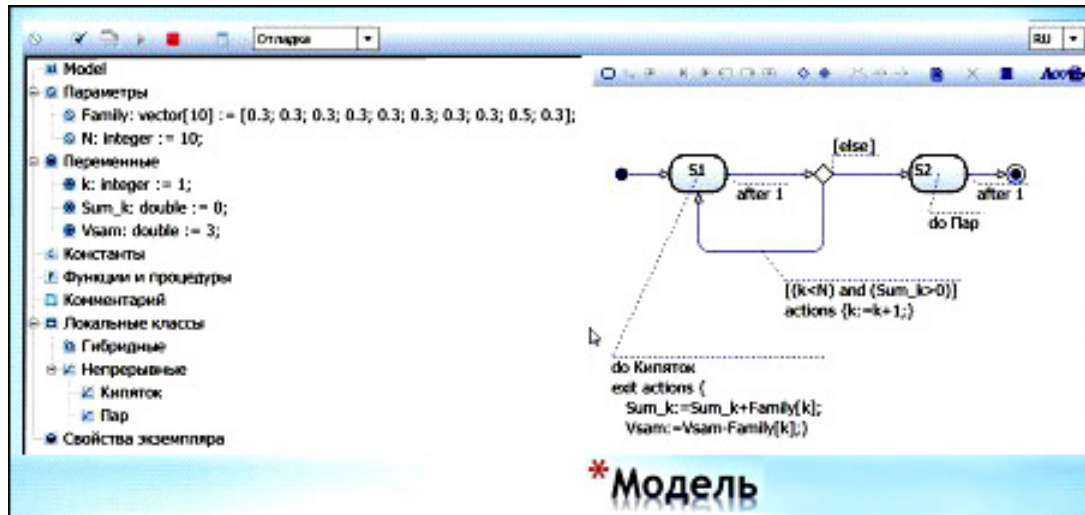


Рис. 4. Фрагмент презентации для лекции о языке моделирования MVL. Модель дискретного процесса «чаепитие» на языке карт поведения с мгновенным наполнением чаем чашки очередного представителя большого семейства и интервалом в одну единицу модельного времени между событиями

Выбор среды моделирования для начинающих определяется прежде всего сложностью языка моделирования. Компьютерное моделирование как пример профессиональной деятельности, в которой сочетается и математика, и теория алгоритмов, и численные методы, и многое, чему студенты будут учиться ближайшие четыре года, достаточно сильно ограничивает возможности выбора. Если добавить сюда требование, чтобы язык был объектно-ориентированным, то приходится выбирать между языком Modelica и MVL. Мы остановились на языке MVL и среде Rand Model Designer.

4. ПРИМЕРЫ ИЗ ПРЕЗЕНТАЦИЙ И ОТЧЕТОВ

Начнем с фрагментов из презентаций (студенческие доклады на лекциях (рис. 5, 6)).

Условие задачи

Мальчик поднимается по лестнице на горку. Затем скатывается с горки и, совершив упругий отскок, перелетает бассейн. Попадет он в воду или нет - вот вопрос, на который мы ответим с помощью программы.

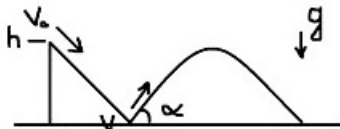


Рис. 5. Прыжок через бассейн



Рис. 6. Футбол. Удар по воротам

Создание компьютерной игры начинается с неформального описания, которое естественно дополнить рисунком (рис. 7).

Условие задачи

Смоделировать бросок мяча в корзину при заданном угле броска α и заданной скорости мяча V_0

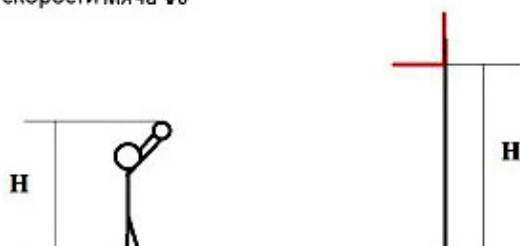


Рис. 7. Баскетбол. Бросок мяча

Основным и повторяющимся элементом игры является очередная попытка достигнуть цели. В случае броска мяча в корзину, нам понадобится хорошо известная первокурснику математическая модель (рис. 8).

Решение задачи

Из курса школьной физики известно, что движение по параболической траектории осуществляется по двум формулам: $x = x_0 + V_0 \cdot \text{time} \cdot \cos(\alpha)$; $y = h - \frac{g \cdot \text{time}^2}{2}$

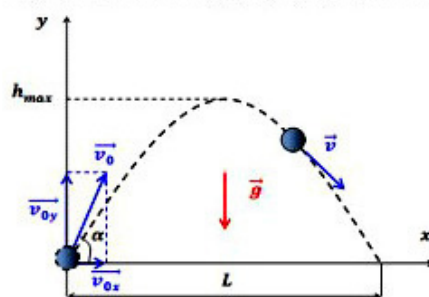


Рис. 8. Полет тела, брошенного под углом к горизонту в поле тяготения Земли

При составлении технического задания уточняется, что называется успешным броском (прямое попадание, попадание с отскоком от щита), неуспешным броском (мяч не долетел до щита и упал на землю, мяч перелетел щит, мяч не попал в корзину), сколько попыток разрешается сделать, какую информацию об очередной попытке запоминать, какую предоставлять игроку. Формальное описание создается с помощью карты поведения (рис. 9).

Отладчик среды позволяет следить за работой карты поведения пошагово. Выполнение программы можно приостановить в заданный момент времени, при наступлении указанного события или при срабатывании выбранного перехода. При остановке можно увидеть значения всех переменных модели.

Последним этапом разработки игры после отладки и тестирования алгоритма, обеспечивающего ожидаемую реакцию на введенные «правильные» и «неправильные» па-

Реализация в Rand Model Design. Алгоритм

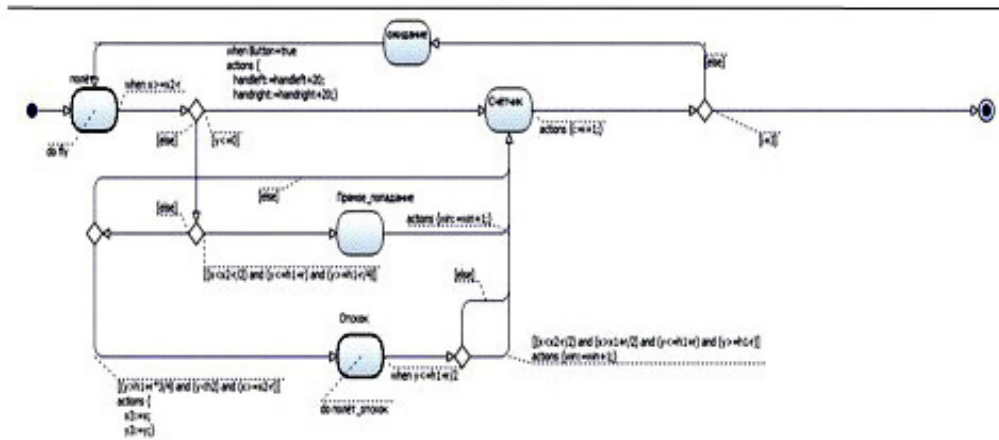


Рис. 9. Карта поведения, обеспечивающая реализацию трех попыток с различными исходами

раметры, является визуализация поведения и оснащение игрового экрана приборами управления (кнопками, слайдерами (рис. 10)).

Rand Model Design. Интерфейс взаимодействия

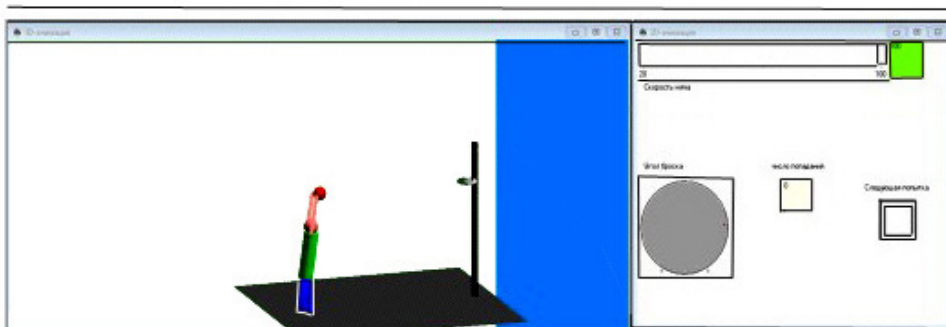


Рис. 10. Пользовательский экран игры. 3D-анимация, слайдеры, кнопки

В этом году практические занятия проходили раз в две недели, и уже в конце ноября большая часть программ была представлена и обсуждена на лекциях (модель конференции). Следует заметить, что если сами доклады проходили не всегда гладко (опыта представления результатов собственной работы у ребят первого курса маловато), то уровень презентаций был удовлетворительным.

С письменными отчетами дело обстояло хуже. Зачем представлять свою работу в виде доклада с использованием презентации, студентам было понятно, и выступать было интересно, а вот зачем учиться еще писать и отчеты — прообразы статей и книг, объяснять удавалось с трудом. Задание написать отчет чаще всего воспринималось как ненужное требование, преподавательская блажь. И все же приведем пример удачного отчета в авторской редакции.

ФРАГМЕНТ ОТЧЕТА

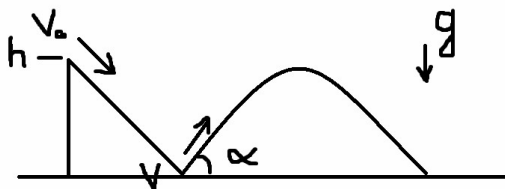
НАЧАЛО фрагмента

План:

1. Постановка задачи
2. Формализация задачи
3. Моделирование задачи
 - 3.1. Создание переменных
 - 3.2. Создание непрерывных локальных классов
 - 3.3. Карта состояний
4. Результат

1. Постановка задачи

Мальчик поднимается по лестнице, затем с заданной начальной скоростью скатывается с горки, отскакивает от батута и летит под углом к горизонту до момента касания земли. На некотором расстоянии от батута располагается бассейн. Задача заключается в том, чтобы установить начальную скорость, при которой мальчик не попадет в бассейн.



2. Формализация задачи

В задаче будем использовать только две координатные оси: Ox и Oy .

Сначала мальчик поднимается по лестнице. Подъем будем осуществлять дискретно, считая, что каждый шаг мальчик делает за 2 секунды. Поднимается мальчик на высоту, равную 10 метрам.

После того, как мальчик поднимется на верх лестницы, начнем осуществлять спуск. Он будет производиться с заданной пользователем скоростью. При движении вниз по горке скорость мальчика будет вычисляться как:

$$V = V_0 - g * time,$$

где V_0 — заданная скорость. Силой трения между мальчиком и горкой пренебрегаем.

После того как мальчик достигнет уровня земли, осуществим упругий отскок от батута; будем считать, что никакой энергии не теряется. При отскоке скорость мальчика будет равна конечной скорости, приобретенной при спуске с горки. Лететь мальчик будет по параболе, уравнения движения по осям согласно физическим законам движения под углом к горизонту:

$$x = x_1 + V_{end} * \cos(alpha) * time$$

$$y = y_1 + V_{end} * \sin(alpha) * time - g * t^2 / 2$$

При достижении мальчиком уровня земли оценим, попал ли он в бассейн: если финальная координата по оси Ox не будет попадать в диапазон координат бассейна, будем считать, что мальчик попал в воду и, соответственно, начальная скорость была задана неверно.

3. Моделирование задачи

3.1. Переменные

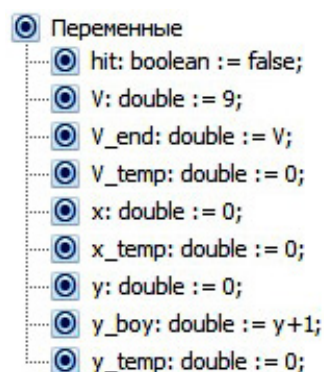


Рис. 1*. Переменные

hit — результат, попадание/не попадание в бассейн,

V — начальная скорость при скатывании с горки,

V_end — конечная скорость при скатывании с горки,

V_temp — дополнительная переменная, которая используется для промежуточных вычислений,

x — координата мальчика по оси Ox,

x_temp — дополнительная переменная, которая используется для промежуточных вычислений,

y — координата мальчика по оси Oy,

y_boy — координата мальчика по оси Oy для построения анимации,

y_temp — дополнительная переменная, которая используется для промежуточных вычислений.

3.2. Константы

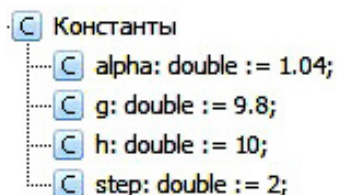


Рис. 2*. Константы

alpha — угол наклона горки,

g — коэффициент свободного падения,

h — высота, на которую должен подняться мальчик,

step — шаг, с которым поднимается мальчик.

3.3. Создание непрерывных локальных классов

В локальных классах мы описываем законы, по которым движется мальчик. В системах уравнений прописываются те уравнения, по которым изменяются координаты мальчика во время спуска с горки и полета после отскока. В искомым переменных мы указываем переменные, которые впоследствии применяются в построении модели поведения объекта, то есть его координаты.

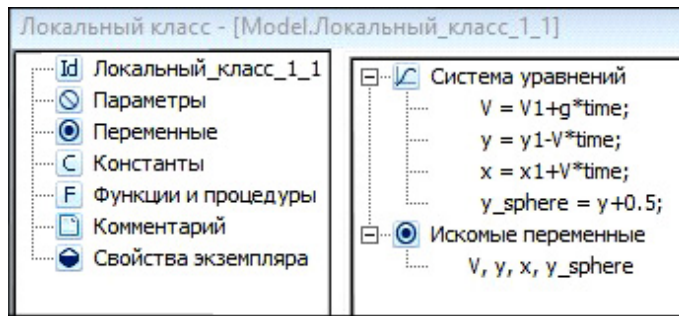


Рис. 3*. Локальный класс_1_1

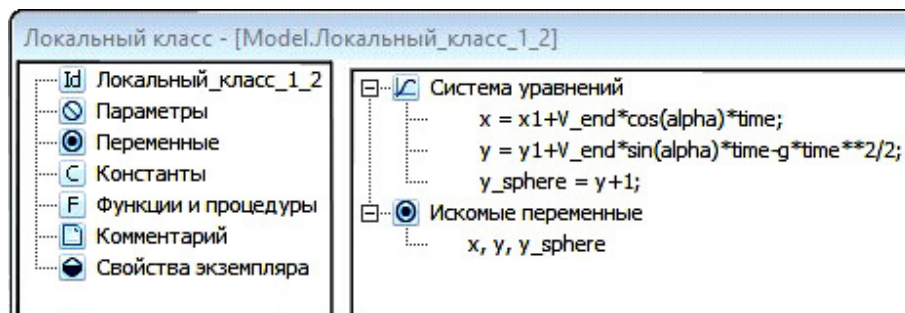


Рис. 4*. Локальный класс_1_2

3.4. Карта поведения

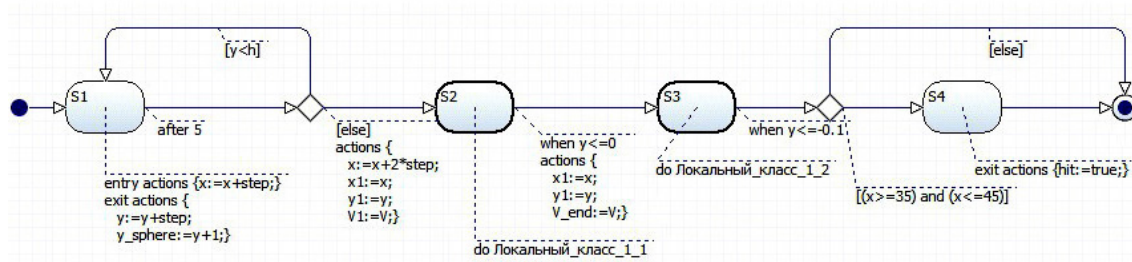


Рис. 5*. Карта поведения

S1 — поднимаемся по ступенькам до достижения нужной высоты.

S2 — спускаемся по горке до достижения земли.

S3 — летим по параболе до достижения земли.

S4 — определяем, попал ли мальчик в бассейн.

4. Результат

Мальчик представлен в виде шарика. При запуске программы он движется согласно фазовой диаграмме сначала по ступенькам, затем по горке, а после этого по параболе над бассейном. Пользователь выставляет начальную скорость спуска при помощи ползунка. После завершения игры цветовой индикатор загорается либо красным, либо зеленым: зеленый — если мальчик не попал в воду, красный — если попал в бассейн и намок.

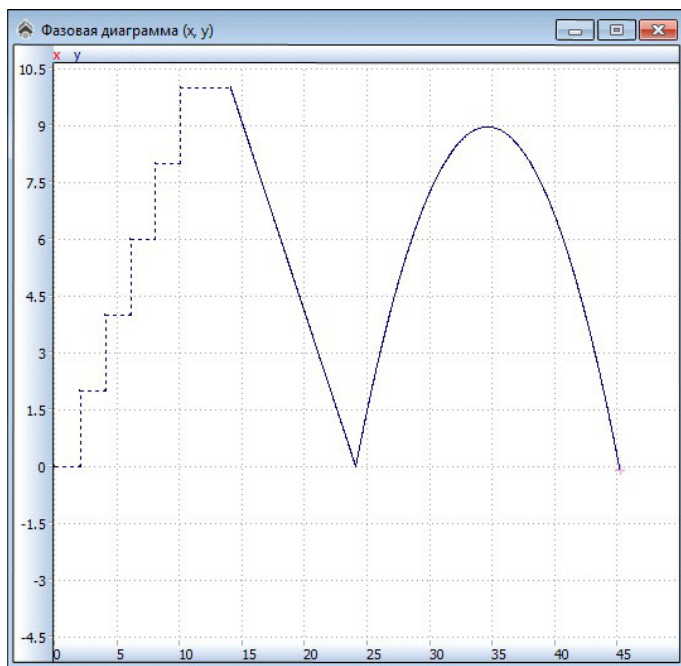


Рис. 6*. Фазовая диаграмма. Движение мальчика

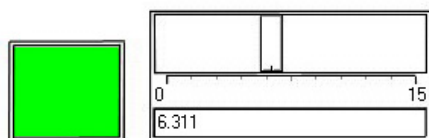


Рис. 7*. Пользовательский интерфейс

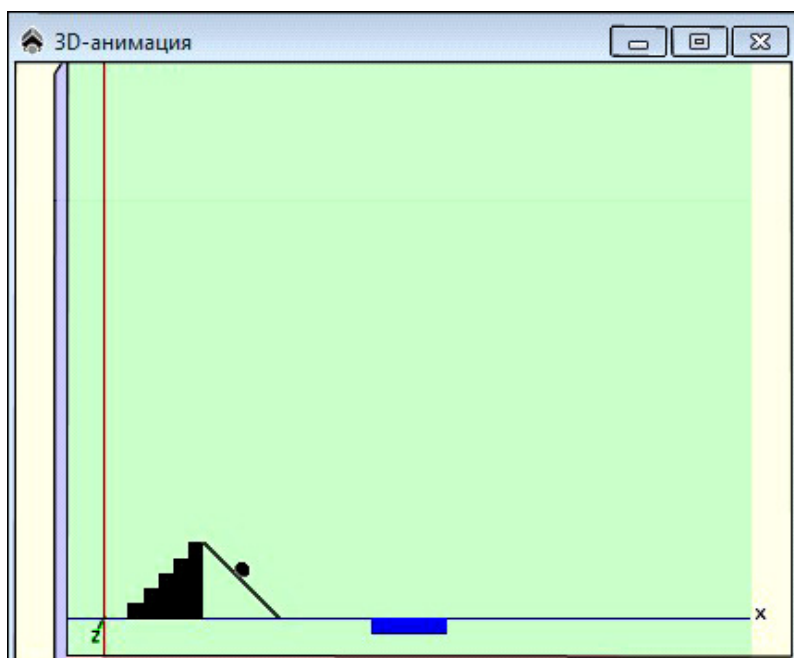


Рис. 8*. Анимация движения мальчика

КОНЕЦ фрагмента

5. ПРЕДВАРИТЕЛЬНЫЕ ОЦЕНКИ

Первые попытки использовать среды визуального моделирования для обучения начинающих были предприняты нами в конце прошлого века [4–6].

В то время среды визуального моделирования в основном использовали для создания виртуальных лабораторий [13–15], то есть для моделей реальных установок с соответствующей цели работы математической моделью и правдоподобным визуальным образом установки. Интерес к виртуальным лабораториям не пропал и сегодня, смотри, например, <https://www.compadre.org/physlets/>. Создано много компьютерных лабораторий по химии, физике (рис. 11).

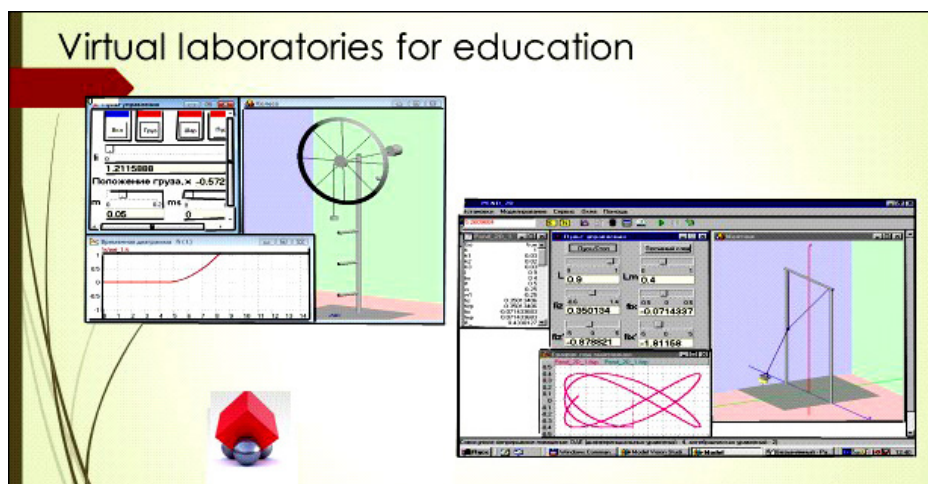


Рис. 11. Виртуальные лаборатории по физике (С.В. Бирюков). Машина Атвуда (<http://ito.edu.ru/2005/Moscow/II/1/II-1-5491.html>) и двойной маятник (<http://ito.edu.ru/2005/Moscow/II/1/II-1-5491.html>)

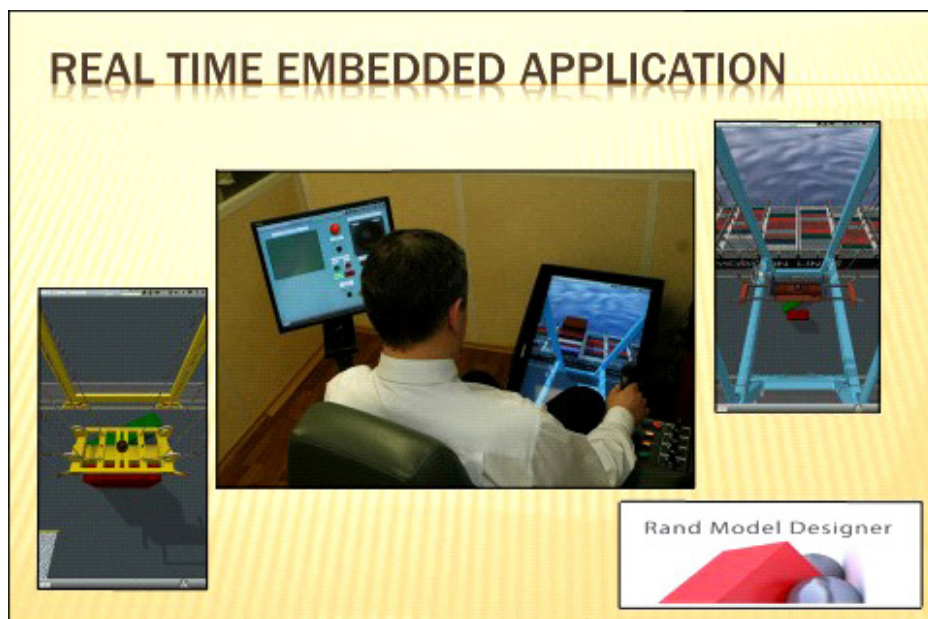


Рис. 12. Промышленные тренажеры. Программно-аппаратный комплекс фирмы TRANSAS

Не могу сказать, насколько такие лаборатории хороши для обучения той же физике — «слишком далеки они от народа»: работа с реальными физическими приборами намного полезнее, — обычно возражают физики. Однако компьютерные установки или тренажеры, с помощью которых приобретают предварительные «черновые» навыки работы с сложными реальными установками, используются достаточно широко.

Комбинация реальных приборов управления и модели, реальных объектов и модели системы управления, то есть аппаратно-программные комплексы, также широко используются сегодня в образовании (рис. 12).

Несомненным достоинством виртуальных лабораторий является возможность проверки правильности предлагаемых решений при конструировании новых механизмов, приборов, систем управления не на реальном макете, а на модели (рис. 13).

Все чаще в образовании используются различные имитационные модели, и тем самым расширяется список предметов, где целесообразно учить моделированию (рис. 14).

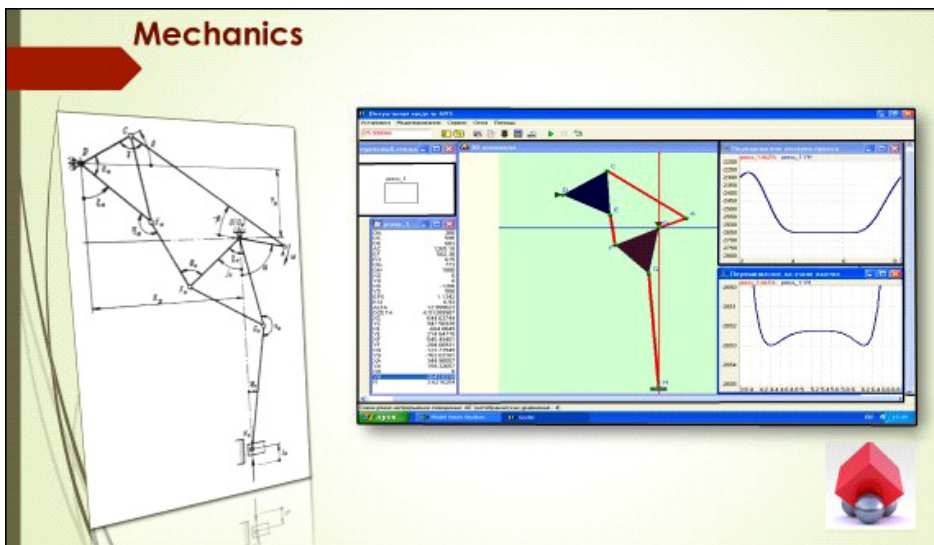


Рис. 13. Виртуальные лаборатории по механике. (Петров Г.Н., ТММ, СПбПУ Петра Великого)

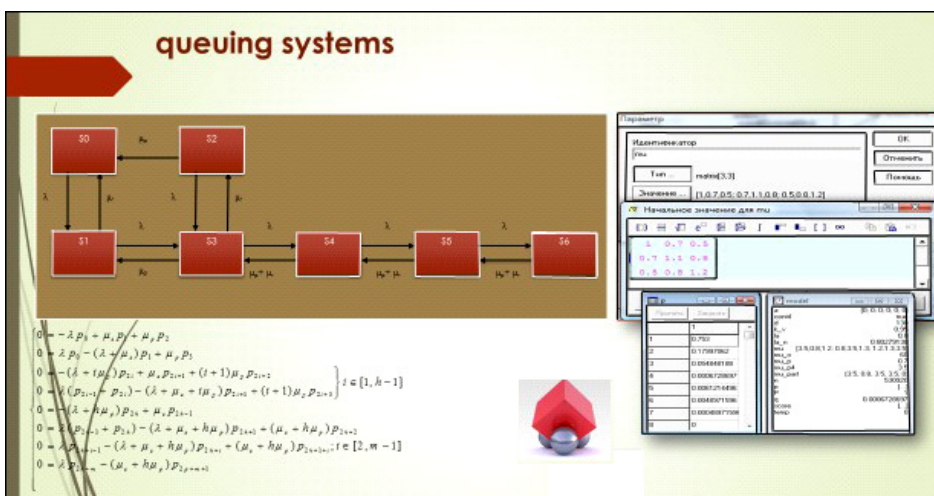


Рис. 14. Имитационная модель. Системы массового обслуживания

Есть много областей, где сложных математических моделей пока не применяют [16], но научить строить имитационные модели (модели, базирующиеся на простых локальных законах поведения и взаимодействиях участников процесса) с помощью современных сред визуального моделирования можно практически любого. Помимо использования виртуальных лабораторий в различных областях, можно говорить и об изучении языков моделирования и технологий компьютерного моделирования, включая конкретные среды визуального моделирования и их языки моделирования.

Языкам моделирования учат школьников [7–12], студентов [17–19], инженеров [3].

Попытки обучения школьников [20] на протяжении нескольких лет в разных формах — двухнедельная летняя практика; четыре дистанционных урока, дополненные летней практикой, показывают, что за столь короткое время невозможно добиться удовлетворительного результата, если речь идет о моделировании.

Ребятам-школьникам при этом, несомненно, интересно, но, думаю, что так же интересно им было бы на практике по любой дисциплине в любом университете.

Другое дело — первокурсники, будущие специалисты информационных технологий. Они хотят заниматься информатикой, соревнуются между собой, не боятся новых, не знакомых им программных продуктов. И в то же время первокурсники не рискуют браться за задачу в одиночку или вдвоем — в среднем группы большие (табл. 4, данные 2018 года, осенний семестр).

Таблица 4. Численность групп

Количество студентов в группе	Число групп	Всего студентов
1	3	3
2	4	8
3	7	21
4	6	24
Итого	20	59

Оценить индивидуальный вклад каждого в работе коллектива трудно, при том, что каждая учебная группа разбивается на две подгруппы. Проверять, консультировать, иногда помогать создавать и отлаживать модели в среднем пяти коллективам одновременно одному преподавателю сложно. Сказать «иди домой и подумай!» нельзя. Домой нужно идти с четким пониманием того, что и как следует сделать или исправить.

Игры можно упорядочить по сложности и трудоемкости (субъективная оценка студентов): сложные — бильярды (шары на прямой, в треугольнике, круге), проще — футболы и баскетболы с ударами без отскоков, горки. Это объяснимо: все, что выходит за рамки хорошо знакомой модели — тела, брошенного под углом к горизонту, сложно!

Таблица 5. Выбор игр

Игра	Количество групп
Самолет и письмо	3
Шары на прямой	1
Самолет и парашютист	4
Баскетбол	3
Футбол	2
Хоккей	1
Горка	5
Бильярд	1

Больше всего лекционного времени ушло на доклады — двадцать коллективов, по семь–десять минут на доклад. Первые доклады слушались с интересом — шел интенсивный сравнительный анализ. Постепенно интерес угасал — спасло коллективное обсуждение ошибок и недостатков.

6. ПОДДЕРЖКА КУРСА С ПОМОЩЬЮ СРЕДЫ SAKAI

И вот теперь мы переходим к главному. Чиновник не только придумывает дисциплины, которые сегодня не востребованы студентами или трудно реализуемы в сегодняшних экономических условиях, но и предлагает инновационные технологии обучения. Одна из «новых» технологий (ей всего то лет тридцать) — дистанционные курсы и дистанционное образование. Студент сам выбирает курс в любом университете мира, слушает его на любом языке, сам выполняет домашние задания. А раз сам выбирает, исходя из своих интересов, то и осваивает изучаемый материал лучше. Студента «тестируют» или он сдает экзамен, и все! Преподаватели не нужны — нужны разработчики курсов, консультанты (tutor), системы автоматического тестирования и интернет! Дешево! Структура университета упрощается — ректорат, бухгалтерия и гигантский деканат, где хранятся международные студенческие сертификаты о прослушанных курсах. Инновационно! Книжки, лекции, семинары устарели, нужны е-книжки, е-лекции, е-семинары. Мы переходим от образования к е-образованию с е-преподавателями! Сердито!

Как всегда, есть хотя бы одна заковырка в таких инновационных проектах: в данном случае, за чужие курсы и чужие сертификаты нужно платить свои деньги. Не беда, говорят нам, пусть студенты слушают курсы только в отечественных университетах, и тогда все можно уладить с помощью российских законов.

Компьютерные системы SAKAI или MOODLE помогут стандартизировать форму представления курсов.

Решили — сделали, в университетах стали создавать порталы дистанционных курсов и покупать оборудование для записи видео-уроков и лекций.

Позволю себе заметить, что SAKAI или MOODLE — это всего лишь системы поддержки курсов, а дистанционные курсы — форма заочного обучения.

Упрощаю — возможно, передергиваю — сознательно, однако делаю этого для того, чтобы рассказать о том, каким мне видится использование системы SAKAI в курсе «Введение в профессиональную деятельность» в сочетании с традиционными лекциями и лабораториями, обычными конспектами, учебниками и методичками для лабораторных работ.

Выбор между SAKAI или MOODLE — дело вкуса, возможности немного отличаются, но главное — использовать.

При глобальной системе идентификаторов в университете подключить любую группу к уже существующему сайту SAKAI или MOODLE не трудно. Если еще есть связь с деканатом, глобальный контроль за работой студента легко обеспечить.

Начнем с поддержки лабораторных работ.

Система поддержки курсов обеспечивает:

1. Автоматическую выдачу индивидуальных заданий для лабораторных работ.
2. Контроль за их своевременным выполнением.
3. Свободный доступ к методическим указаниям. Можно обеспечить дозированную выдачу дополнительных материалов, вплоть до прошлогодних отчетов по той же

теме, если студенту данное задание трудно выполнить. Истек срок сдачи отчета, становится доступным новая информация, помогающая выполнить задания, а оценка автоматически снижается. Если студенту выдается уже существующий отчет, от него требуется только прочесть, повторить, и уметь модифицировать готовую модель («удовлетворительно»).

4. Автоматическую загрузку отчетов с датами, что позволяет оценивать умение работать по плану.

Здесь отличий от традиционного способа проведения лабораторий нет.

Переходим к лекциям. В системе поддержки курсов можно разместить:

1. Лекционные презентации и комментарии лектора к ним (лекторский конспект). На мой взгляд, лекторский конспект, каким бы хорошим он ни был, всегда хуже студенческого.
2. Учебные пособия, правда, возникает конфликт с издательством. Очевидно, что, если это необходимо, будет куплена одна электронная версия учебника на всех. К сожалению, даже это встречается не часто — в интернете можно найти все бесплатно.

При наличии презентаций и лекторского конспекта, лекцию можно превратить в обсуждение заранее прочитанного материала (мне, честно сказать, это удается редко, так как к дискуссии обычно готов только я).

Если дополнительно записывать лекции на камеру, курс становится заочным, со всеми вытекающими отсюда недостатками, в частности, коммерческими (часто это трактуется как достоинство), и создавать сайты таких курсов разумно с помощью специальных групп технической поддержки. Здесь уместно сравнение создания сайтов с изданием книг. Книга создается автором и издательством совместно, и тогда это книга, а не рукопись, изданная по оригинал-макету автора.

7. ЗАКЛЮЧЕНИЕ

Эту статью можно рассматривать как черновик учебного рабочего плана по курсу «Введение в профессиональную деятельность» для студентов, изучающих информатику. Математическое и компьютерное моделирование постепенно становятся фундаментальными инженерными дисциплинами, и хочется надеяться, что такой пример профессиональной деятельности оправдан. Отсутствие учебника — временный недостаток, особенно если найдутся соавторы.

P.S. Прошу рассматривать мои суждения о современных чиновниках как оценочные суждения и не судить меня строго за мою советскую модель общества, в котором мирно сосуществовали различные функционеры и различные обыватели — от академиков до простых советских тружеников, в одном лице.

Благодарность



Co-funded by the
Erasmus+ Programme
of the European Union

Список литературы

1. *Martin-Villalba, C., Urquia, A., Senichenkov, Y., Kolesov, Y.* Two approaches to facilitate virtual lab implementation // *Computing in Science and Engineering*. 2014. Vol. 16 (1). P. 78–86.
2. *Martin-Villalba, C., Urquia, A., Senichenkov, Y., Kolesov, Y.* Hybrid systems. Preliminary Comparative Analysis of Modelica and Model Vision Language // *Университетский научный журнал*. 2014. № 8. С. 87–96.
3. *Колесов Ю. Б., Сениченков Ю. Б.* Объектно-ориентированное моделирование в среде Rand Model Designer 7. М.: Проспект, 2016.
4. *Колесов Ю. Б., Сениченков Ю. Б.* Как быстро и качественно «приготовить» интерактивную модель динамической системы // *Компьютерные инструменты в образовании*. 1998. № 3-4. С. 52–61.
5. *Колесов Ю. Б., Сениченков Ю. Б.* Математическое моделирование в картинках или рисунки поведения динамических систем с помощью «MODEL VISION» // *Компьютерные инструменты в образовании*. 1998. № 5. С. 45–52.
6. *Колесов Ю. Б., Сениченков Ю. Б.* Компьютерное моделирование в научных исследованиях и образовании // *Exponenta Pro. Математика в приложениях*. 2003. № 1. С. 4–11.
7. *Макарова Н. В., Сениченков Ю. Б., Титова Ю. Ф.* Школа моделирования-2003. Занятие 1 // *Компьютерные инструменты в образовании*. 2003. № 1. С. 39–48.
8. *Макарова Н. В., Сениченков Ю. Б., Титова Ю. Ф.* Школа моделирования-2003. Занятие 2. Моделирование движения объекта в среде MVS // *Компьютерные инструменты в образовании*. 2003. 2. С. 75–83.
9. *Макарова Н. В., Сениченков Ю. Б., Титова Ю. Ф.* Школа моделирования-2003. Занятие 3. Возможности компьютерного эксперимента в среде MVS // *Компьютерные инструменты в образовании*. 2003. № 3. С. 55–63.
10. *Макарова Н. В., Сениченков Ю. Б., Титова Ю. Ф.* Школа моделирования-2003. Занятие 4. Моделирование систем, или может ли компьютер моделировать собачью радость? // *Компьютерные инструменты в образовании*. 2003. № 4. С. 61–68.
11. *Макарова Н. В., Сениченков Ю. Б., Титова Ю. Ф.* Школа моделирования-2003. Занятие 5. Моделирование систем. маленькая модель большого универсама // *Компьютерные инструменты в образовании*. 2003. № 5. С. 51–55.
12. *Колесов Ю. Б., Макарова Н. В., Сениченков Ю. Б., Титова Ю. Ф.* Школа моделирования-2003. Занятие 6. Визуализация моделей // *Компьютерные инструменты в образовании*. 2003. № 6. С. 65–70.
13. *Бутиков Е. И.* Лаборатория компьютерного моделирования // *Компьютерные инструменты в образовании*. 1999. № 5. С. 24–39.
14. *Бутиков Е. И.* Интерактивные компьютерные модели в преподавании физики / «Моделирование–2003». Труды 4-й международной научно-технической конференции. СПб. 23–28 июня 2003 г. С. 50–52.
15. *Кавтрев А. Ф.* Компьютерные модели в школьном курсе физики // *Компьютерные инструменты в образовании*, 1998, № 2. С. 41–47.
16. *Козырева Д. Д., Ампилова Н. Б.* Математические модели в социологии и методы их исследования // *Компьютерные инструменты в образовании* 2016, № 5. С. 5–16.
17. *Колесов Ю. Б., Сениченков Ю. Б.* Моделирование систем. Динамические и гибридные системы. Учебное пособие. СПб.: БХВ-Петербург, 2005.
18. *Колесов Ю. Б., Сениченков Ю. Б.* Моделирование систем. Объектно-ориентированный подход. Учебное пособие. СПб.: БХВ-Петербург, 2005.
19. *Колесов Ю. Б., Сениченков Ю. Б.* Моделирование систем. Практикум по компьютерному моделированию (CD-ROM). СПб.: БХВ-Петербург, 2006.
20. *Сениченков Ю. Б., Бакулева Е., Быков И., Зубань Л.* Компьютерное моделирование для школьников // *Компьютерные инструменты в школе*. 2016 № 4. С. 32-40.

Поступила в редакцию 27.06.2018, окончательный вариант — 02.08.2018.

Computer tools in education, 2018

№ 5: 16–38

<http://ipo.spb.ru/journal>

[doi:10.32603/2071-2340-2018-6-16-38](https://doi.org/10.32603/2071-2340-2018-6-16-38)

A COURSE ON “INTRODUCTION TO PROFESSIONAL ACTIVITY” or How to Breathe Life into What an Official Has Touched

Senichenkov Yu. B.

Peter the Great Saint-Petersburg Polytechnic University, Saint Petersburg, Russia

Abstract

A variant of the course «Introduction to Professional Activity» for first-year students in the field of Informational Technologies is discussed. Students are encouraged to develop a computer game using the domestic environment of visual modeling named Rand Model Designer gradually, going through all the stages of creating software. The possibilities of using the SAKAI system for the support of training courses are discussed.

Keywords: *mathematical and computer modeling courses, object-oriented modeling, computer modeling technologies, computer modeling for beginners.*

Citation: Yu. B. Senichenkov, “A Course on “Introduction to Professional Activity” or How to Breathe Life into What an Official Has Touched”, *Computer tools in education*, no. 6, pp. 16–38, 2018 (in Russian). [doi:10.32603/2071-2340-2018-6-16-38](https://doi.org/10.32603/2071-2340-2018-6-16-38)

Received 27.06.2018, the final version — 02.08.2018.

Yuriy B. Senichenkov, St. Petersburg Peter the Great Polytechnic University, Institute of Computer Science and Technology, Professor, doctor of science; 194021 Saint-Petersburg, Polytehnicheskaya st., 21, ICST, senyb@dcn.icc.spbstu.ru



Наши авторы, 2018.

Our authors, 2018.

**Сениченков Юрий Борисович,
доктор технических наук, профессор
высшей школы «Программная инженерия»
Санкт-Петербургского политехнического
университета Петра Великого;
194021 Санкт-Петербург,
ул. Политехническая, д. 21, ВШ ИКНиТ,
senyb@dcn.icc.spbstu.ru**