

С. М. Кашаев,
Нижегородский государственный лингвистический университет,

Л. В. Шерстнева, Д. С. Гладских,
Нижегородский государственный технический университет

АЛГОРИТМ СОСТАВЛЕНИЯ РАСПИСАНИЯ УЧЕБНЫХ ЗАНЯТИЙ

Аннотация

Рассматривается алгоритм для автоматизации составления расписания занятий в учебном заведении. В качестве критерия качества расписания предлагается использовать сумму условных штрафов при несоблюдении тех или иных желательных ограничений (пожеланий к расписанию). Приведенный алгоритм был реализован практически в приложении Microsoft Excel с использованием программирования на VBA.

Ключевые слова: учебное заведение, расписание занятий, аудиторный фонд, VBA, Excel, составление расписания.

Составление расписания занятий в учебном заведении традиционно является трудоемким процессом. Эта работа непосредственно связана с качеством учебных занятий, а также с эффективной загрузкой аудиторного фонда. Составление расписания учебных занятий можно трактовать как оптимизацию использования таких ограниченных ресурсов, как преподавательский состав и аудиторный фонд [3].

Основная задача и одновременно основная трудность составления расписания учебных занятий касается необходимости формирования удобного расписания для всех преподавателей и групп учащихся без исключения. В этом плане можно отметить требование составления расписания «без разрывов» как для учебных групп, так и для преподавателей, ограничение по минимальному и максимально возможному числу занятий в течение дня, выбор определенного времени и дня недели для ряда занятий, пожелание выбора определенного учебного корпуса (а часто преподаватели предпочитают и определенные аудитории) и т. д. К процессу составления расписания добавляется требование эффективного

использования аудиторного фонда (аудитории для занятий следует подбирать по численности групп). Все это делает процесс составления расписания решением сложной задачи оптимизации с большим количеством ограничений.

Известно, что задача планирования расписания учебных занятий — это задача комбинаторного типа, характерными особенностями которой являются огромная размерность и наличие большого числа ограничений сложной формы. Фактически в настоящее время не существует универсальных методов решения таких задач. Математическая (классическая) теория расписаний [1, 4, 5] охватывает лишь узкий круг хорошо формализуемых проблем, которые обычно сводятся к задаче коммивояжера, транспортной задаче и т. п. Прямое применение данных методов к задаче составления расписания учебных занятий не представляется возможным. В настоящей работе предложен критерий оптимальности расписания учебных занятий и алгоритм формирования расписания в соответствии с этим критерием, а также приведена реализация данного алгоритма

Контактная информация

Кашаев Сергей Михайлович, канд. тех. наук, доцент, доцент кафедры математики и информатики Нижегородского государственного лингвистического университета; *адрес:* 603155, г. Нижний Новгород, ул. Минина, д. 31а; *телефон:* (831) 416-60-44; *e-mail:* cashaev@rambler.ru

Шерстнева Людмила Владимировна, канд. физ.-мат. наук, зам. директора Института радиоэлектроники и информационных технологий Нижегородского государственного технического университета; *адрес:* 603155, г. Нижний Новгород, ул. Минина, д. 24; *телефон:* (831) 436-93-47; *e-mail:* sherlv@rambler.ru

Гладских Дарья Сергеевна, студентка Института радиоэлектроники и информационных технологий Нижегородского государственного технического университета; *адрес:* 603155, г. Нижний Новгород, ул. Минина, д. 24; *телефон:* (831) 436-93-47; *e-mail:* inland.holland@gmail.ru

S. M. Kashaev,
Linguistic University of Nizhny Novgorod,
L. V. Sherstneva, D. S. Gladskih,
Nizhny Novgorod State Technical University

ALGORITHM OF MAKING A CLASS SCHEDULE

Abstract

An algorithm to automate the scheduling of classes in the school is considered in the article. As a criterion of the quality of the schedules is proposed to use the sum of the conditional penalties if the failure to comply with desired constraints (requirements and timetable). The algorithm was practically implemented in Microsoft Excel using VBA programming.

Keywords: educational institution, class schedule, classroom fund, VBA, Excel, making schedule.

в приложении Microsoft Excel с использованием программирования на VBA.

Критерий качества расписания

Очевидно, что некоторые ограничения при составлении расписания являются обязательными (например, для конкретного преподавателя необходимо поставить занятия только в определенные дни недели), а другие — желательными (или необязательными) для выполнения в итоговом расписании (например, одному преподавателю желательно ставить занятия не раньше второй пары, а другому желательно поместить все занятия в два любых дня недели). Таких необязательных ограничений при составлении расписания может быть много и факт их невыполнения предлагается отмечать условным *штрафом*. В этом случае определенные недостатки в расписании будут измеряться бóльшим штрафом, а какие-то менее значительные недостатки — меньшим. Сумма таких штрафов по всему расписанию и говорит о его качестве. В идеальной ситуации эта сумма штрафов должна быть равна нулю, и тогда расписание удовлетворяет всем ограничениям (как обязательным, так и необязательным). И *цель приводимой далее алгоритмизации процесса составления расписания заключается в том, чтобы обеспечить минимизацию штрафов по составленному расписанию*.

Пусть для составления расписания учебных занятий мы располагаем M заявками на проведение занятий. Обозначим через i номер заявки S_i в составляемом расписании. При этом каждая заявка на учебное занятие S_i характеризуется рядом параметров:

$$S_i = f(P_i, k_i, t_i, d_i, a_i, g_i, \bar{n}_i), \quad (1)$$

где:

P_i — преподаватель, указанный в заявке;
 k_i — название дисциплины;
 t_i — время начала пары;
 d_i — день планируемого занятия;
 a_i — аудитория для занятия;
 g_i — учебная группа;
 $\bar{n}_i$ — набор номеров недель для указанного занятия;
 i — принимает значения от 1 до M .

Здесь $P_i, k_i, g_i, \bar{n}_i$ являются исходными данными при составлении заявки, а t_i, d_i, a_i требуется подобрать при соблюдении всех обязательных ограничений и максимально полном выполнении всех необязательных ограничений.

Что касается *обязательных* ограничений, то тут все понятно — они должны быть обязательно отражены в расписании. Если это по каким-либо причинам не удается сделать, то данная заявка на обслуживание становится отклоненной (дожидается ручного анализа диспетчером по расписанию).

Наибольшая сложность связана с выполнением (или невыполнением) *необязательных* ограничений. Как было сказано выше, невыполнению каждого необязательного ограничения предлагается поставить в соответствие условный штраф. В случае выполнения данного необязательного ограничения штраф

равен нулю, в противном случае — это определенное число, большее нуля. При этом в зависимости от важности того или иного необязательного ограничения значения штрафа будут различаться. В результате сумма всех штрафов по определенной заявке говорит о качестве ее обслуживания.

Оформим сказанное в виде критерия качества i -й заявки по необязательным ограничениям:

$$Q_i = \sum_{l=1}^{L_i} q_l(i), \quad (2)$$

где:

Q_i — сумма всех штрафов по необязательным ограничениям для i -й заявки;

$q_l(i)$ — значение штрафа для i -й заявки по l -му необязательному ограничению;

L_i — общее число необязательных ограничений для i -й заявки.

В этом случае оптимально обслужить i -ю заявку следует так, чтобы подобранные ее параметры t_i, d_i, a_i приводили к минимизации Q_i :

$$Q_i \rightarrow \min. \quad (3)$$

В идеале $Q_i = 0$, что говорит о том, что все необязательные ограничения для i -й заявки в расписании соблюдены. Этот критерий мы и будем использовать в алгоритме составления расписания занятий. При этом понятно, что должны быть соблюдены и обязательные ограничения.

Соотношение (2) определяет величину штрафа по отдельно взятой i -й заявке из расписания. Для того чтобы охарактеризовать качество расписания в целом, мы должны получить величину штрафа для всего расписания:

$$Q = \sum_{i=1}^M Q_i = \sum_{i=1}^M \sum_{l=1}^{L_i} q_l(i). \quad (4)$$

И для наилучшего расписания характерна минимальная величина штрафа:

$$Q \rightarrow \min. \quad (5)$$

Таким образом, в соответствии с рассматриваемым критерием мы должны рассмотреть все возможные варианты обслуживания заявок (проанализировать параметры t_i, d_i, a_i для всех занятий) и выбрать тот, для которого соотношение (4) соответствует минимуму. Учитывая большой объем заявок на занятия, а также множество возможных значений для t_i, d_i, a_i , решение данной задачи переборным способом не представляется возможным. К тому же на практике ситуация с формированием расписания еще больше усложняется. Так, в связи с тем, что реально перед началом составления расписания не удастся учесть все ограничения, в уже готовом расписании всегда остаются неформализованные условия, которые в нем не соблюдены. Скажем, определенной группе учащихся основные образовательные программы в понедельник необходимо завершить до 14:00, так как в этот день недели планируется большинство факультативов. Таких ситуаций может быть достаточно много, и это выясняется после составления расписания, что требует последующей доработки автоматизированного расписания ручным способом

диспетчером с целью максимального учета этих новых факторов.

Таким образом, требуется предложить алгоритм, который в соответствии с соотношениями (2) и (4) будет обеспечивать автоматизированный подбор параметров t_i , d_i , a_i имеющихся заявок.

Описание алгоритма

В процессе обслуживания заявок в составляемом расписании занятий важную роль играет их порядок. Каждая заявка на проведение занятий характеризуется некой свободой выбора по дню недели, времени начала занятия (это время фактически дискретное — начало пары, например, 9:55, 11:35 и т. д.) и аудитории. Это можно трактовать как *вариативность* или *степень свободы занятия*. В каких-то заявках эта вариативность минимальная. Например, определенному преподавателю (скажем, это приглашенный специалист из известной фирмы) нужно поставить занятие только в понедельник в 9:55 в аудитории 3105 (в ней находится необходимая для проведения занятия техника). В других ситуациях степень свободы занятия больше — ограничений по дням, времени и аудиториям меньше. Скажем, другому преподавателю обязательно ограничить занятия только по дням (понедельник или пятница). В большинстве ситуаций такие ограничения в заявках не такие значительные, а в некоторых вообще отсутствуют. Численно степень свободы занятия можно определить так:

$$V_i = N_z(i) \cdot N_a(i), \quad (6)$$

где:

$N_z(i)$ — общее количество учебных пар по учебному заведению в течение всей недели, которые подходят для обслуживания i -й заявки (например, пять учебных дней и шесть пар по расписанию в течение дня приводят к $N_z(i) = 30$);

$N_a(i)$ — общее число учебных аудиторий, которые подходят для проведения данного занятия (с точки зрения вместимости, наличия необходимого оборудования, требований преподавателя и т. д.).

Таким образом, на первом этапе формирования расписания составляются заявки на занятия с указанием преподавателя, дисциплины, группы учащихся и номеров учебных недель в течение семестра. После этого определяются ограничения по аудиториям и времени проведения занятий, а на основании этих ограничений вычисляется степень свободы занятия V_i для каждой i -й заявки. Далее заявки на занятия пересортировываются в порядке возрастания V_i с целью обслуживания сначала наименее вариативных. После этого производится последовательное обслуживание пересортированных заявок. Обслуживание заключается в выборе параметров t_i , d_i , a_i для i -й заявки таким образом, чтобы и соблюсти выполнение обязательных ограничений, и обеспечить минимизацию штрафов по необязательным ограничениям в соответствии с (3).

Необязательных ограничений в учебных организациях достаточно много, и к типичным можно отнести следующие:

- лекционные занятия должны проводить на первых двух парах;

- в занятиях у студентов не должно быть окон;
- занятия по дополнительным программам могут начинаться не ранее определенной пары (например, не ранее 15:00);
- в расписании преподавателей не должно быть окон;
- у преподавателей должно быть не менее одного свободного дня.

Одно из вышеуказанных необязательных ограничений («в занятиях у студентов не должно быть окон») требует пояснения. Дело в том, что мы проводили тестирование данной методики для учебного заведения, где это допускается (хотя и является исключительной ситуацией). В большинстве же учебных организаций такое не допускается. В этом случае данное ограничение убирается из перечня необязательных ограничений и переносится в категорию обязательных.

В такой редакции задачу составления расписания уже можно решать программными методами. При этом, как уже говорилось, необходимо обеспечить возможность последующей доработки автоматизированного расписания ручным способом диспетчером с целью максимального учета новых, не учтенных до этого факторов.

Таким образом, можно сказать, что задача составления расписания имеет много ограничений и нечеткое множество факторов. Решение таких задач, как правило, осуществляется в два этапа: получение оптимального (с точки зрения используемых критериев) варианта и его последующая доработка человеком (диспетчером) с целью максимального учета неформализованных факторов.

В качестве дальнейшего совершенствования приведенного алгоритма составления расписания представляется возможным переход от соотношений (2) и (3) к «более качественной» оптимизации — минимизации выражения:

$$\bar{Q}_i = \sum_{n=1}^{M(i)} Q_n = \sum_{n=1}^{M(i)} \sum_{l=1}^{L_i} q_l(n), \quad (7)$$

где:

$\bar{Q}_i$ — совокупная сумма штрафов как по i -й заявке, так и по всем предыдущим, уже обработанным заявкам, которые имеют общие параметры с i -й заявкой (одна и та же группа учащихся, один и тот же преподаватель);

$M(i)$ — часть заявок из совокупности от S_1 до S_i , которые имеют общие параметры с i -й заявкой (по группе учащихся и преподавателю).

В этом случае обслуживание заявки S_i предполагает возможную коррекцию предыдущих, уже обслуженных заявок, которые нашли место в расписании. При этом коррекция касается только тех заявок из уже обслуженных, которые имеют общие параметры с текущей заявкой S_i .

Программная реализация

Описанный выше алгоритм составления расписания учебных занятий был реализован в приложении Microsoft Excel с использованием программирования на VBA. При этом, как уже говорилось выше, на завершающем этапе составления расписания (после

выполнения автоматизации) требуется обеспечить ручную доводку. Это связано с тем, что некоторые начальные заявки на занятия не могут быть обслужены в расписании из-за совокупности наложенных на них ограничений. Например, требуется поставить несколько компьютерных занятий на определенное время. Однако, если число компьютерных классов меньше числа заявок, такое невозможно. Также часто преподаватели ограничивают свои пожелания в плане занятий небольшим количеством вариантов по времени и дням недели. Кроме того, в течение учебного семестра зачастую требуется вносить различные коррективы в первоначально составленное расписание. Все это требует обеспечения для диспетчера удобного механизма коррекции уже составленного автоматически расписания.

Поясним данную разработку, которая представляет собой книгу Microsoft Excel. В нее включено несколько листов с данными и несколько листов управления с необходимыми интерфейсными элементами управления. Исходными данными для программы является справочная информация, которая предварительно вносится в данную книгу на специально отведенный для этого лист (это список аудиторий с указанием вместимости, список учебных групп, дисциплин, перечень начала учебных пар и т. д.). Для внесения информации о заявках на занятия предназначен отдельный лист (в нем каждая заявка располагается в одной строке, в которой указаны P_i, k_i, g_i, n_i). На другом листе (рис. 1) формируется информация по перечню обязательных и необязательных ограничений для заявок по времени, дням и аудиториям.

После этого вычисляются степени свободы каждой заявки (6) и производится пересортировка заявок по возрастанию V_i . В результате в верхней части листа заявок (рис. 2) оказываются те, которые имеют наименьшую степень свободы занятия. И именно в этом порядке будет проводиться их обслуживание (подбор времени, дня, аудитории).

При автоматическом обслуживании i -й заявки будем выбирать параметры $\{t_i, d_i, a_i\}$ таким образом, чтобы величина штрафа по данной заявке:

$$Q_i = \sum_{l=1}^{L_i} q_l(i) \text{ была минимальна.}$$

В выполненной программной реализации было использовано пять необязательных ограничений ($L = 5$). Эти пять ограничений и значения штрафов при невыполнении ограничения (указаны в скобках) выглядят следующим образом:

- отсутствие разрывов в расписании учебных групп ($q_1 = 10$);
- отсутствие переходов между корпусами в течение дня ($q_2 = 8$);
- лекционные занятия должны проводиться на первых двух парах ($q_3 = 7$);
- не более четырех занятий для группы учащихся в течение дня ($q_4 = 3$);
- отсутствие разрывов в расписании преподавателей ($q_5 = 1$).

В результате обслуживания заявок в соответствии с данным алгоритмом мы получаем в основном обработанный лист заявок — для тех заявок, для которых это возможно, подобраны время, день и аудитория.

На этом этап автоматического формирования расписания завершен, и теперь мы должны предоставить диспетчеру средства для коррекции расписания. Наиболее просто и эффективно выполнять технические действия по коррекции и дополнению расписания на предназначенном для этого листе «Расписание» (рис. 3). Данный лист заполняется динамически (в нем отображаются те группы, с расписанием которых мы в данный момент работаем, — сам этот перечень формируется на отдельном листе). Лист «Расписание» организован традиционным для его составления способом — сетка с указанием дней недели и времени начала занятий. Это некий аналог листа бумаги, на котором мы составляем расписание, и технические действия заключаются во внесении в ячейки названий предметов. Основная работа сотрудника (диспетчера) на листе «Расписание» выполняется в стиле «копировать — вставить». Это означает создание новой заявки на занятие на основании уже имеющейся. Параметры заявки-основания при этом копируются в новую заявку. Далее пользователь может поменять параметры этой новой заявки, например, название дисциплины и преподавателя. Также в процессе этого действия на экране отображается форма для подбора аудито-

Номер заявки	День-время	День	Время	Аудитория	Вкл	Описание			
121	1	0	0	0	Да	Пон	9-55		
275	1	0	0	0	Да	Вт	11-35	Среда	15-00
389	0	1	0	0	Нет	Пон	Среда		
312	0	0	1	0	Нет	8-15	15-00		

Рис. 1

3	Номер	Заявитель	Преп.	День	Время	Кол ст	Обс	Номер ауд-и	Группа	Дисц	0	1	2	3	4	5
4	ФВО-1	ИСУ	Рязанова Н.В.			11			10ДВ	Проф лексика-зач+кр и ДП-зач		*	*			
5	ФВО-1	ИСУ	Воронков Ю.В.			11			10ДВ	Проф лексика-зач+кр и ДП-зач			*	*	*	
6	ФВО-1	ИСУ	Акимов С.С.			10			10ИВ	Инвест.анализ		*			*	
7	ФВО-1	ИСУ	Каданина А.Л.			10			10ИВ	Инвест.анализ		*			*	
8	ФВО-1	МИИ	Стулова О.Е.			10			10ИВ	Инвест.анализ		*			*	
9	ФВО-1	Воронков Ю.В.	Акимов С.С.			10			10ИВ	Ист.зар.иск. и ар. ч.15 (19-20вв.)		*			*	

Рис. 2

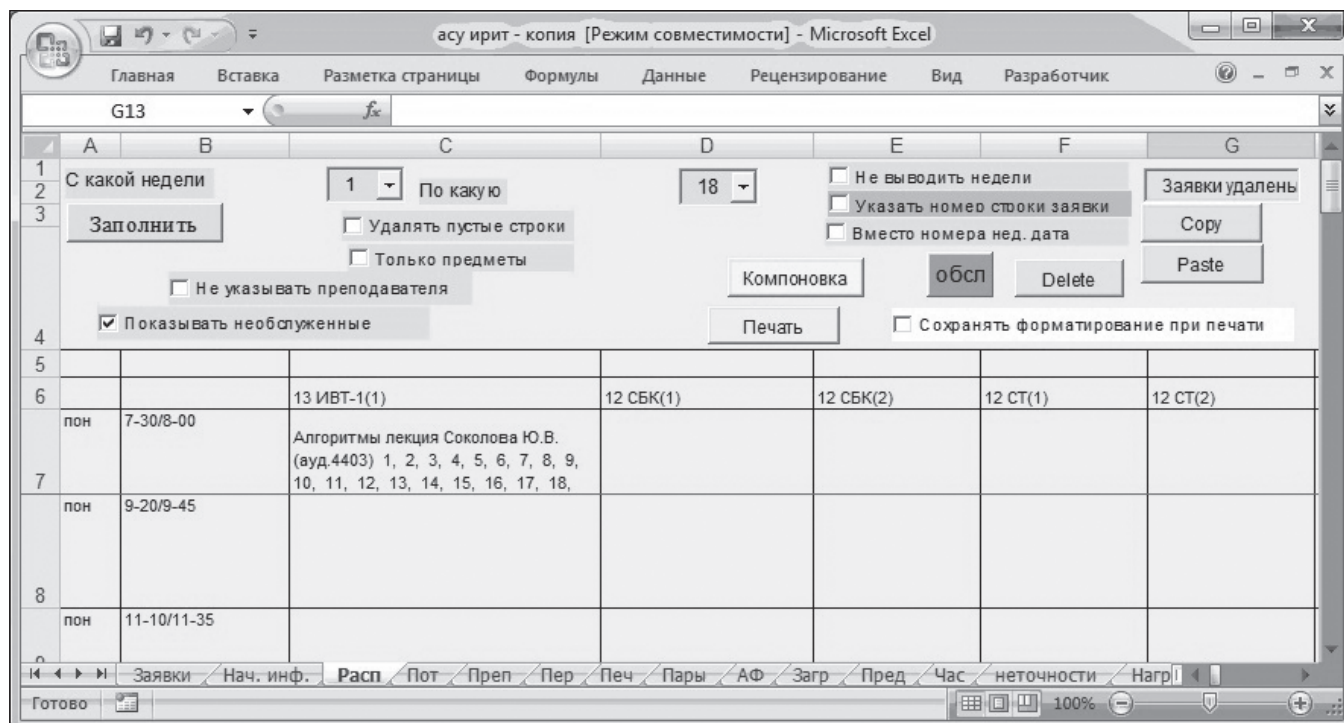


Рис. 3

рии (аудиторию можно указать самостоятельно либо предоставить это программе). В результате таких визуальных действий автоматически формируется новая заявка на листе «Заявки» (см. рис. 2). При этом обеспечивается полный контроль корректности введенной заявки на предмет отсутствия накладок.

Обсудим процесс ручного подбора аудитории для занятия. Электронная форма подбора аудитории

показана на рисунке 4. В этом случае предусмотрен контроль вместимости аудиторий и проверка занятости аудитории. При попытке постановки занятия в уже занятую аудиторию программа не блокирует данное действие, а только предупреждает об этом. Это связано с тем, что иногда необходимо разместить разные занятия в одной аудитории. При автоматическом поиске аудитории для занятия можно учесть

Рис. 4

пожелания преподавателей по тому или иному корпусу. В случае последовательных по времени занятий у одной группы учащихся можно задать автоматическое размещение этих занятий в одной аудитории. Таким образом, действия составителя расписания аналогичны традиционным, когда на листе бумаги мы вписываем нужные занятия и стираем те, которые надо убрать из расписания. Здесь процесс вполне аналогичен, только он выполняется с помощью щелчков мышью.

В разработку заложен ряд отчетов, которые помогают принять правильные управленческие решения для руководящих структур. В первую очередь, это отчет о занятости аудиторного фонда различными подразделениями в течение конкретной недели [3]. Другой отчет представляет собой сведения по занятости аудиторного фонда в определенном интервале недель. В этом случае можно достаточно быстро визуально оценить занятость аудиторного фонда и найти свободный ресурс для проведения занятий. Кроме того, на данном листе заложен функционал по переносу занятий в другое место и на другое время. Занятия можно переносить из одной аудитории в другую и целиком, и в определенном интервале недель. В этом случае все изменения приводят к коррекции на листе «Заявки». Как показала практика использования, фактически вся работа по управлению расписанием занятий в учебном заведении может быть выполнена с помощью представленной разработки.

В заключение кратко остановимся на объеме вычислительных действий при применении данной методики составления расписания, а конкретно — на первом ее этапе, когда производится автоматическая обработка имеющихся заявок. В процессе использования данной методики при составлении реального расписания рассматривалось около 100 групп учащихся, расписание составлялось на семестр (18 учебных недель), и в среднем было 14–16 занятий в неделю в каждой группе. В этом случае обработка заявок занимала несколько десятков минут, что вполне приемлемо, так как фактически это разовое действие, после которого наступает этап ручной доводки полученного «полуфабриката» расписания занятий. Он требует аналитической работы диспетчера и занимает доминирующее время в общем процессе составления расписания.

Литература

1. Гэри М., Джонсон Д. Вычислительные машины и труднорешаемые задачи / пер. с англ. М.: Мир, 1982.
2. Кашаев С. М. Офисные решения с использованием Microsoft Excel и VBA. СПб.: Питер, 2009.
3. Кашаев С. М., Шерстнева Л. В., Гладских Д. С. Управление расписанием занятий и загрузкой аудиторного фонда в учебных заведениях // Информатика и образование. 2014. № 6.
4. Танаев В. С., Гордон В. С., Шафранский Я. М. Теория расписаний. Одностадийные системы. М.: Наука, 1984.
5. Танаев В. С., Шкурба В. В. Введение в теорию расписаний. М.: Наука, 1975.

НОВОСТИ

Предприятиям хотят разрешить включать в расходы на обучение траты на сетевое образование

Комиссия по законопроектной деятельности одобрила законопроект, направленный на совершенствование системы профессионального образования. Об этом сообщается на официальном сайте Правительства РФ.

Предварительно участники комиссии высказали свои замечания и предложения. Планируется внести изменения в статью 264 Налогового кодекса, по которым расходами организации на обучение сотрудников будут также считаться расходы, связанные с договорами о сетевой форме реализации образовательных программ.

Соответственно, предприятие, заключающее контракты с образовательными организациями с сетевым обучением, может быть работодателем для выпускника.

«Это будет способствовать развитию системы профессионального образования, обеспечит мотивацию предприятий к участию в практико-ориентированной модели подготовки высококвалифицированных кадров», — говорится в сообщении.

Законопроект подготовлен Министерством финансов РФ и будет рассмотрен на заседании Правительства.

(По материалам федерального портала «Российское образование»)

Создана электронная замена стикеров — программируемые «точки»

Устройство под названием Dot разработано в качестве цифровой замены стикеров — «точка» отправит на ваш смартфон заранее запрограммированное напоминание, как только вы мимо нее пройдете. С помощью Dot можно оставлять цифровые записки членам семьи или напоминания самому себе о выполнении каких-либо обязанностей, например, о том, что нужно вынести мусор. Вместо показа напоминания устройство можно заставить открыть какое-либо приложение. Dot работает, пользуясь технологией Apple iBeacon. Светодиод на устройстве можно использовать для передачи каких-нибудь сведений.

Например, красный свет может напомнить о том, что на улице жара, а зеленый — сообщить об отсутствии заторов на дорогах. В одной комнате может находиться сразу несколько устройств. При этом одна и та же «точка» может одновременно быть запрограммирована на выполнение нескольких функций при условии, что они не конфликтуют друг с другом. Расстояние до Dot смартфон определяет по силе сигнала Bluetooth. Проблему, связанную с ослаблением сигнала по мере подсадки батареи, разработчики планируют решить программным способом.

(По материалам международного компьютерного еженедельника «Computerworld Россия»)