

Адабияттар

1. Айтматов Ч. И дольше века длится день. Плаха. Пегий пес, бегущий краем моря. – Ф.: Кыргызстан, 1981. – 656 с.
2. Алимбеков А. Мектеп окуучусунун этномаданий компетенциялары// Шоокум. – Б., 2012. – №2.- 24-25-бб.
3. Бекбоев И., Алимбеков А. Азыркы сабакты даярдап өткөрүүнүн технологиясы.-Б.: Бийиктик, 2011.-192 б.
4. Волков Г.Н. Этнопедагогика: Учеб. для студ. сред. и высш. пед.учеб. заведений. – 2-е изд., испр. и доп. – М.: Издательский центр «Академия», 2000. – 176 с.
5. Муратов А. Ч. Айтматов жазуучу –этнопедагог. Бишкек.- 2006.
6. Салимова К. , Нана Доде. Педагогика народов мира: История и современность.- М.:Педагогическое общество России, 2001.-576 с.

УДК 004.9

АЛГОРИТМЫ, МЕТОДЫ И МОДЕЛИ ТЕСТИРУЮЩЕГО МОДУЛЯ АВТОМАТИЗИРОВАННОЙ ОБУЧАЮЩЕЙ СИСТЕМЫ ПО ПРОГРАММИРОВАНИЮ

Исмаилов Бактыбек Искакович, д.т.н., профессор, Национальная Академия Наук КР, 720010, г. Бишкек, пр. Чуй 265а, Тел: 0771358645, e-mail: bismailov47@gmail.com.

Каткова Светлана Николаевна, ст. преподаватель, КГТУ им. И. Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Мира 66, Тел:0550437100, e-mail: goodday54@yandex.ru

Аннотация. В статье показано, как сделать процесс обучения программированию удобным для студента, развивающим не только теоретические основы, но и практические умения, навыки в разработке программ, т.е. компетенций, а также как обеспечить объективность оценки его знаний с помощью системы автоматизированной обучающей (АОС).

Ключевые слова: онтология, концепция модульности, модель компетенций студента, компетентностный подход, эталонные модели.

THE ALGORITHMS, METHODS AND MODELS OF AUTOMATED TESTER MODULE LEARNING PROGRAMMING SYSTEM

Ismailov Bakytbek Iskakovich, Ph.D., Professor, National Academy of Sciences of the Kyrgyz Republic, 720010, Bishkek, Chui Avenue 265a, Phone: 0771358645,

e-mail: bismailov47@gmail.com.

Svetlana Katkova, Senior Lecturer, KSTU name after I. Razzakov, Kyrgyzstan, 720044 Bishkek, Mira 66, Phone.: 0550437100, e-mail: goodday54@yandex.ru.

Abstract. The article shows how to make learning programming convenient for the student to develop not only the theoretical basis, but also practical skills, skills in developing programs, ie competences, as well as how to provide an objective assessment of his knowledge with the help of the automated training system (ATS).

Keywords: ontology, the concept of modularity, the model student competence, competence approach, reference models.

В настоящее время онтологии находят широкое применение в программировании [1]. Рассматриваются подходы к использованию онтологий и средств разработки онтологий в

учебном процессе [2]. Широко известной обучающей системой, разработанной на основе онтологического подхода, является система БиГОР [3]. Онтологический подход в сочетании с концепцией модульности позволил реализовать в ней синтез индивидуальных маршрутов обучения.

Онтологии применяются и при обучении программированию, хотя работ в этой области чрезвычайно мало, например, в [4] онтология языка С представлена в виде семантической сети, в [5] описан процесс разработки онтологии и «образовательная» онтология для изучения программирования на языке С. Эта онтология разработана авторами для применения в обучении программированию, однако ее использование в какой-либо автоматизированной обучающей системе не предполагается. В известной системе ECOLE используется онтологический подход при обучении дисциплине «Экспертные системы и логическое программирование».

Ни в одной из этих автоматизированных систем практически отсутствуют исследования, связанные с формированием *модели компетенций студента*, отражающей его способности применять знания, умения, навыки и личностные качества для успешной деятельности в конкретной профессиональной области. В связи с интенсивным внедрением в образование компетентного подхода, разработка таких моделей представляется весьма актуальной.

Цель статьи - обосновать необходимость ввода малоизученного адаптивного элемента в образовательную технологию, без которого невозможно обеспечить индивидуальный характер обучения студентов, а именно, *эталонную модель* тестового задания на программирование.

Для достижения этой цели необходимо решить следующие задачи:

1. Разработать подробный пошаговый алгоритм работы тестирующего модуля АОС, в котором используются *эталонные модели* для оценивания качества обучения студентов.
2. Подобрать математический аппарат, значительно повышающий объективность оценки знаний студентов на основе *эталонных моделей*.
3. Описать формальную *модель компетенций студента*.

Компетентный подход в образовании предлагает при обучении делать упор на развитии компетенций студентов. Программирование относится к таким дисциплинам, в которых крайне необходимо развивать практические навыки и умения в программировании. Без практического опыта студент не сможет стать профессиональным программистом.

Для развития компетенций перспективна *адаптивная технология* обучения, подстраивающаяся под каждого студента, с учетом фактических знаний и опыта студента на момент начала обучения. На практике реализация компетентного подхода и адаптивной технологии возможна только на основе АОС.

При традиционном, не автоматизированном обучении программированию осуществляется три вида деятельности.

1. *Информационная* деятельность предоставляет студенту в том или ином виде учебный материал (лекции, тексты, рисунки, графики и т. п.).
2. *Тренирующая* деятельность направлена на выработку необходимых практических умений и навыков по составлению программ, т.е. компетенций.
3. *Тестирующая* деятельность обеспечивает обязательную проверку и оценивание знаний студента преподавателем. Она играет особую роль в обучении программированию. Оценивание качества разработанной студентом программы представляет собой неформальный процесс, и выставаемая оценка является в значительной мере субъективной. Именно поэтому объективное оценивание работы студента является актуальной задачей, которая должна быть решена в автоматизированной обучающей системе по программированию.

АОС должна обеспечивать все три вида деятельности процесса обучения с помощью соответствующих программных модулей. Кроме того, система должна осуществлять гибкое

управление процессом обучения, адаптируясь к уровню знаний и умений конкретного обучающегося. Алгоритм адаптивного процесса обучения представлен на Рис.1.

Чтобы обеспечить индивидуальное обучение студентов и подстроить образовательный процесс под его уровень знаний и умений, в статье предлагается ввести в тестирующий модуль АОС малоисследованную *эталонную модель* задания при определении качества его выполнения студентом. (Рис.3).

С помощью пошагового алгоритма подробно опишем процесс оценки уровня сложности L заданной для тестирования знаний студента программы, а также процесс определения степени знаний $I(S)$ студента на основе уровня сложности L задания. Алгоритм состоит из 9 шагов:

1 шаг: Создаем модель тестирования. Модель тестирования программы M_t представляет собой три множества:

$T_t = \{ t_1, t_2, \dots, t_k \}$ – множество тестов для проверки знаний, $R_t = \{ r_1, r_2, \dots, r_k \}$ – множество эталонных ответов на множество тестов T_t , $D_t = \{ d_1, d_2, \dots, d_k \}$ – множество диагностических сообщений о знаниях.

С помощью модели тестирования система автоматически должна проверять правильность выполнения разработанного студентом программного кода. Результат выполнения теста сравнивается с соответствующим эталонным результатом. Результаты сравнения представляют собой булевский вектор $B = (b_1, b_2, \dots, b_n)$, каждый элемент которого равен либо 0 либо 1.

2 шаг: На основе этого вектора вычисляем оценку за *выполнение* задания O_v по формуле:

$$O_v = 0,6 \times \frac{\sum_{i=1}^k b_i}{k} \quad (1)$$

Таким образом, если все тесты пройдены, то оценка за выполнение максимальна и равна 0,6. Если какие-то тесты не выполнены, то оценка уменьшается.

3 шаг: Если оценка за выполнение $O_v = \max$, для повышения объективности оценки, кроме оценки O_v , следует учитывать качество задания O_k .

4 шаг: Для повышения объективности оценки качества задания обучающая система должна создавать статическую модель M_S выполненного обучающимся задания Рис.2.

5 шаг: Следует использовать метод, повышающий объективность оценки, это *метод сравнения статических моделей* программного кода. В соответствии с этим методом сравнивается модель M_S с эталонной, созданной преподавателем статической моделью M_s . Оценка качества программы вычисляется при сравнении двух этих моделей. Сравнение построенной статической модели и эталонной сводится к сравнению деревьев, которое можно выполнить *эвристическим рекурсивным алгоритмом полного перебора*. Статические модели представляют собой деревья небольшого размера, поэтому оценка O_k вычисляется как «расстояние» статической модели студента MS от эталонной статической модели M_s :

$$D = f(MS, M_s) \quad (2)$$

6 шаг: Вычисляется *разовая оценка* за выполнение одного задания $O(e)$ на основе модели задания по формуле:

$$O(e) = (O_v + O_k) \times w, \quad (3)$$

Где: e – номер задания, O_v – оценка за *выполнение* задания, O_k – оценка за качество выполненного задания. $w \in (0, 1]$ – вес (важность) задания для усвоения единицы знаний, к которой «привязано» данное задание.

7 шаг: Из разовых оценок выполняемых заданий формируется вектор оценок, а вернее – *разовый вектор знаний* студента, который является *моделью знаний студента*.

$$K(s) = (p_1, \dots, p_n), \quad (4)$$

Каждый элемент p_i вектора $K(s)$ выражает степень знания студентом единицы знаний (одного задания по программированию). Эту модель можно назвать *моделью компетенций студента*, так как её конечное формирование проходит через множество повторных этапов тренирующего и тестирующего процессов, вектор переформируется на каждой новой итерации тестирования. Различают пять уровней усвоения учебного материала: понимание, узнавание, воспроизведение, применение, творческая деятельность. Уровням усвоения для

каждой единицы знаний можно поставить в соответствие степень знания студентом единицы знаний: нет знаний (A), новичок (N), начинающий (B), продвинутый (F), эксперт (E).

8 шаг: Определяется *уровень сложности задания*. Единица знаний может интерпретироваться как дискретная случайная величина со значениями A, N, B, F, E . Для вычисления распределения вероятностей для каждой единицы знаний (случайной величины) используется байесовская сеть доверия. Сеть представляет собой ациклический граф отношения.

9 шаг: Определяется степень знания обучаемого.

Уровень сложности задания L соответствует степеням знания ученика:

$L < 0,1$ – нет знаний (A), $0,1 \leq L \leq 0,4$ – для новичка (N), $0,41 \leq L \leq 0,7$ – для начинающего (B), $0,71 \leq L \leq 0,9$ – для продвинутого (F), $0,91 \leq L \leq 1$ – для эксперта (E).

На основании вектора знаний определяется интегральный $I(s)$ уровень обученности студента. Интегральный уровень обученности вычисляется в соответствии с количеством элементов вектора $K(s)$ соответствующего уровня. Общий принцип таков: студент считается достигшим уровня L ($L = A, N, B, F, E$), если количество элементов уровня L в векторе знаний не меньше 70 %, причем остальные 30 % представляют собой предыдущий уровень.

Выводы. В статье обоснована необходимость введения в тестирующий модуль АОС малоизученного адаптивного элемента *эталонной модели* благодаря решению задач:

1. разработке алгоритмов работы АОС и тестирующего модуля в виде блок-схем и пошагового алгоритма работы тестирующего модуля, использующего *эталонные модели*;
2. подбору математического аппарата, базирующегося на *эталонных моделях* и дающего возможность значительно повысить объективность оценки знаний студентов;
3. определению *формальной модели компетенций студента*, при реализации которой в тестирующем модуле АОС, студент приобретает практические навыки и умения в процессе разработки и тестирования кодов, разработанных им программ.

В итоге отметим, что система алгоритмов, моделей и методов, представленная в статье, дает реальную возможность создать на её основе качественную *онтологию* тестирующего модуля (Рис. 4) – составляющую базы знаний, а также эффективную и правдивую адаптивную АОС по обучению программированию в дальнейшем.

Рисунки для статьи:

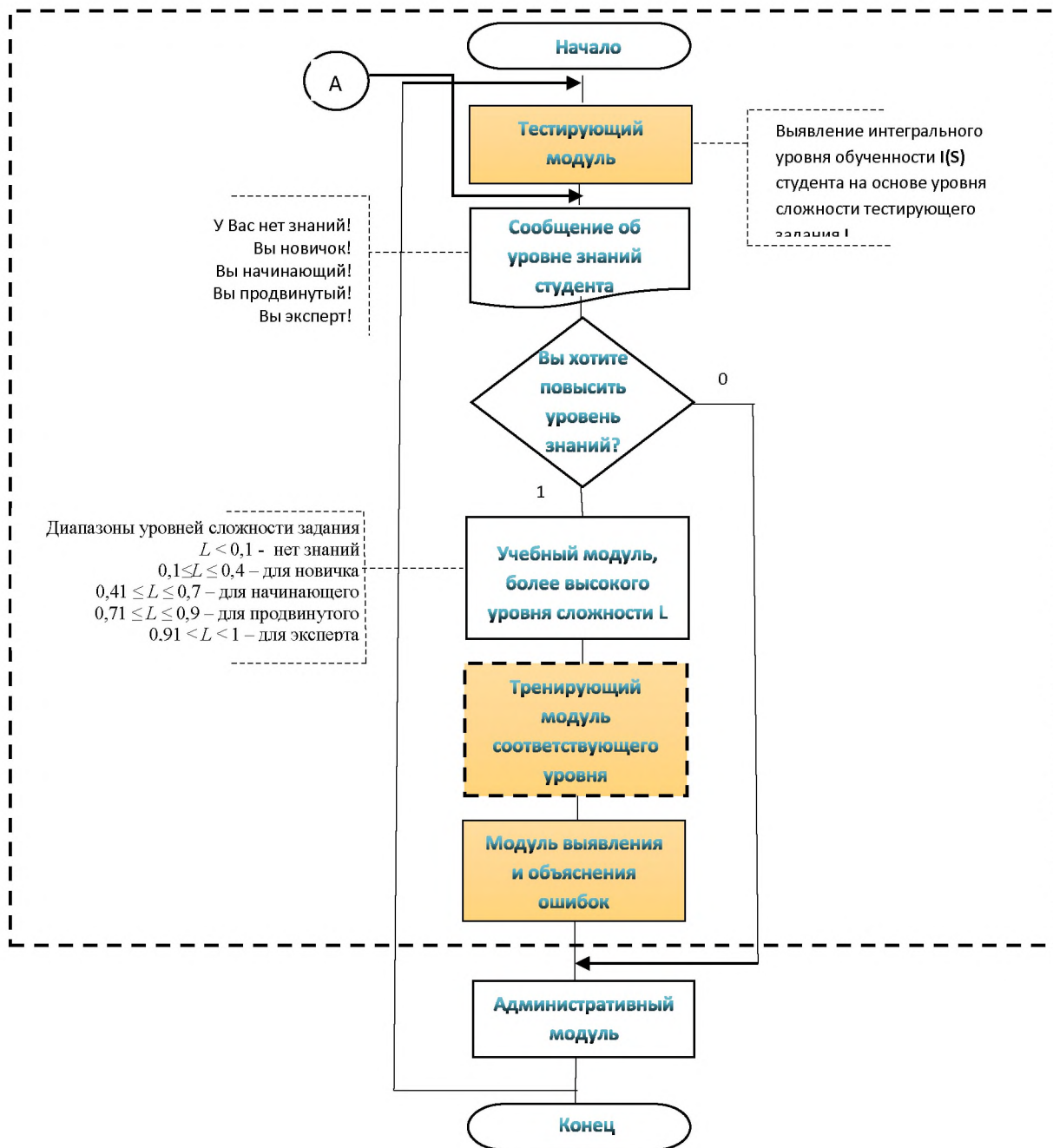


Рис. 1. Алгоритм адаптивного обучения программированию

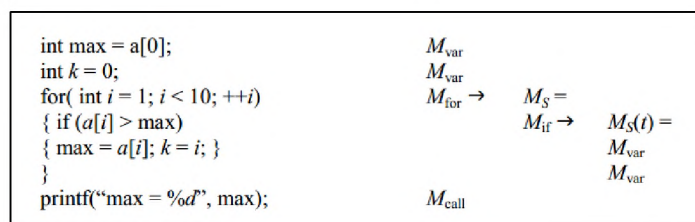


Рис. 2. Пример статической модели MS программы в виде корневого бинарного дерева

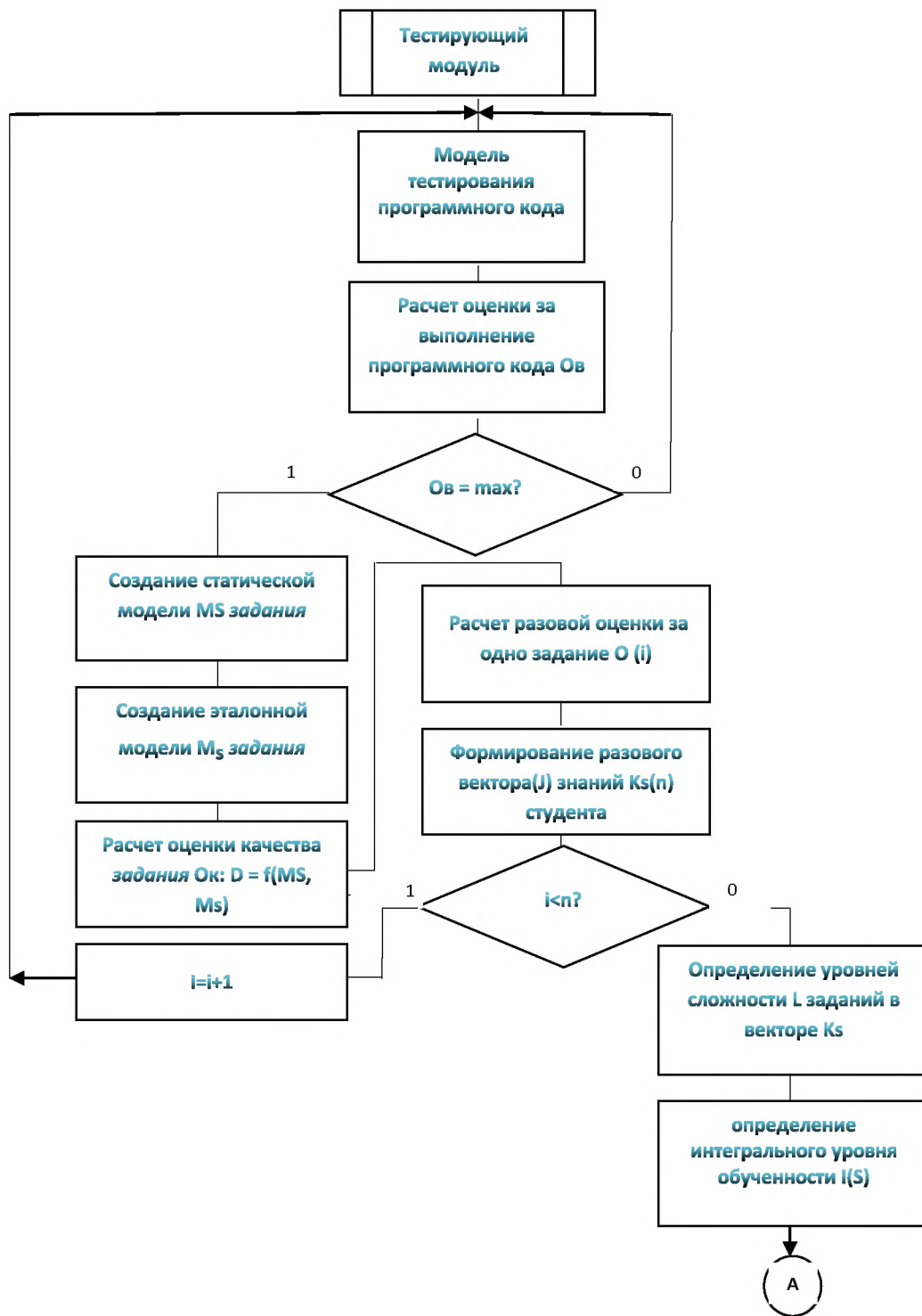


Рис.3. Алгоритм определения степени обученности студента по уровню сложности задания

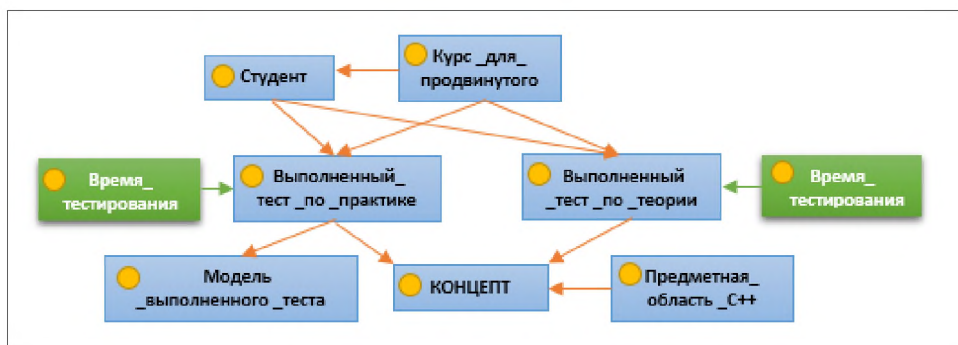


Рис.4. Онтология верхнего уровня модуля тестирования с уровнем сложности тестов «для продвинутого»

Список литературы

1. Клещев А. С. Роль онтологий в программировании. Ч. 1. Аналитика // Информационные технологии. – 2008. – № 11. – С. 28–33.
2. Кафтаников И. Л., Коровин С. Е. Перспективы использования WEB-онтологий в учебном процессе // Educational Technology & Society. – 2003. – № 6 (3). – С. 134–138.
3. Норенков И. П., Уваров М. Ю. База и генератор образовательных ресурсов // Информационные технологии. – 2005. – № 9. – С. 60–65.
4. Онтологии и тезаурусы: модели, инструменты, приложения: учеб. пособие / Б. В. Добров, В. В. Иванов, Н. В. Лукашевич, В. Д. Соловьев. – М.: Интернет-университет информационных технологий; БИНОМ. Лаборатория знаний, 2009. – 173 с.
5. Константинова Н. С., Митрофанова О. А. Онтологии как системы хранения знаний. – <http://www.ict.edu.ru/ft/005706/68352e2-st08.pdf>.

УДК 621.38(075.8)

ИНТЕГРИРОВАННЫЙ УЧЕБНО-ЛАБОРАТОРНЫЙ КОМПЛЕКС ПО ЭЛЕКТРОНИКЕ

Кармышаков Аскарбек Камалдинович, к.т.н., доцент, КГТУ им. И.Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Мира 66, e-mail: askar1969@mail.ru

Канаев Болотбек Бакаевич, вед. инженер СКБ ИЭТ при КГТУ им. И.Раззакова, Кыргызстан, 720044, г. Бишкек, пр. Мира 66, e-mail: tech-astra-kg@mail.ru

Цель статьи - рассмотрение особенностей технической разработки и реализации лабораторных систем и их применение в учебном процессе. Авторами предложена современная, универсальная и легко перенастраиваемая учебно-лабораторная система для обучения студентов по основам электроники. Разработаны и созданы опытные образцы лабораторных комплексов для исследования основных элементов электронной техники.

Ключевые слова: преобразование сигналов, частота сигнала, система измерения, электронные элементы и приборы, микроконтроллер.

INTEGRATED TRAINING AND LABORATORY SYSTEM FOR ELECTRONICS

Karmyshakov Askarbek Kamaldinovich, PhD (Engineering), Associate professor, KSTU named after I. Razzakova, Kyrgyzstan, 720044, Bishkek, Kyrgyzstan, e-mail: askar1969@mail.ru

Kanaev Bolotbek Bakaevich, Lead Engineer STB IET at KSTU named after I. Razzakova, Kyrgyzstan, 720044, Bishkek, Kyrgyzstan, e-mail: tech-astra-kg@mail.ru