

## Технология адаптивного тестирования в среде R

*Национальный аэрокосмический университет им. Н. Е. Жуковского  
«Харьковский авиационный институт»*

В настоящее время вопросам дистанционного обучения уделяется много внимания. Актуальными являются задачи проектирования и разработки систем адаптивного тестирования, позволяющие не только эффективно оценивать уровень усвоения знаний, но и организовывать гибкий процесс дистанционного обучения, ориентированного на индивидуальные способности обучаемого. Существующие системы компьютерного адаптивного тестирования являются достаточно дорогостоящими. Бесплатная программно-инструментальная среда R даёт возможность реализовать ряд функций адаптивного тестирования, а пакет *mirtCat*, основанный на функциях таких пакетов, как *shiny* и *mirt*, позволяет реализовать механизм ввода-вывода тестовых заданий через веб-интерфейс и делает возможным проведение анализа теста на основе одномерных и многомерных моделей теории IRT.

**Ключевые слова:** дистанционное обучение, массовые онлайн-курсы, компьютерное адаптивное тестирование, тестовые задания, качество теста.

### Введение

В настоящее время в связи с развитием массовых онлайн-курсов и систем дистанционного обучения широко применяется технологии компьютерного тестирования [1–2]. Это обусловлено тем, что компьютерное тестирование является эффективным и, часто, единственно возможным способом контроля процесса дистанционного обучения. Методам и подходам компьютерного адаптивного тестирования (КАТ) уделяется много внимания [3–5]. В основном методы КАТ базируются на моделях современной теории тестирования IRT [6–7], основанных на вычислении параметров модели методом максимального правдоподобия. В работах [8–10] предлагается для реализации КАТ использовать модели сплайнов, которые позволяют получать характеристики заданий теста в автоматизированном режиме с более высокой точностью. Однако, несмотря на быстрые темпы развития методов и подходов в педагогическом тестировании, использование методов компьютерного адаптивного тестирования все еще ограничено. Это обусловлено рядом факторов:

- сложностью в понимании и реализации компьютерных адаптивных методик тестирования, базирующихся на моделях современной теории тестирования IRT;
- необходимостью создания объемного банка калиброванных тестовых заданий;
- отсутствием эффективных методик адаптивного тестирования для различных учебных дисциплин и различных популяций испытуемых.

Это далеко не весь перечень проблем, которые приводят к трудностям использования адаптивных тестов при компьютерном оценивании. Кроме того, существуют проблемы, связанные с идентификацией обучаемых, проходящих тестирование. В массовых онлайн-курсах возникают проблемы, касающиеся контекстуальных характеристик обучаемых (пол, возраст, страна проживания, уровень образования и т.д.), которые приводят к низким результатам прохождения тестов и досрочному окончанию обучения. Основной же проблемой при

использовании КАТ является сложность математических и алгоритмических моделей и методов, положенных в основу работы адаптивного теста, что, в свою очередь, приводит к невозможности реализации процесса КАТ обычным преподавателем для своих дистанционных курсов. Например, некоторые проблемы, связанные с использованием моделей в процессе КАТ, представлены в работах [11–12].

Поэтому актуальными являются обоснование выбора инструментальных средств, а также разработка методики компьютерного адаптивного тестирования, которая даёт возможность оперативно и качественно реализовать процесс КАТ на базе современных компьютерных средств.

Цель данной работы – анализ современных инструментальных средств, позволяющих реализовать механизм адаптивного тестирования, и разработка методики запуска КАТ на базе HTML-интерфейса на примере тестов по информатике.

### 1. Анализ существующих систем и функций, реализующих процесс компьютерного адаптивного тестирования

Общий алгоритм КАТ показан на рис. 1.

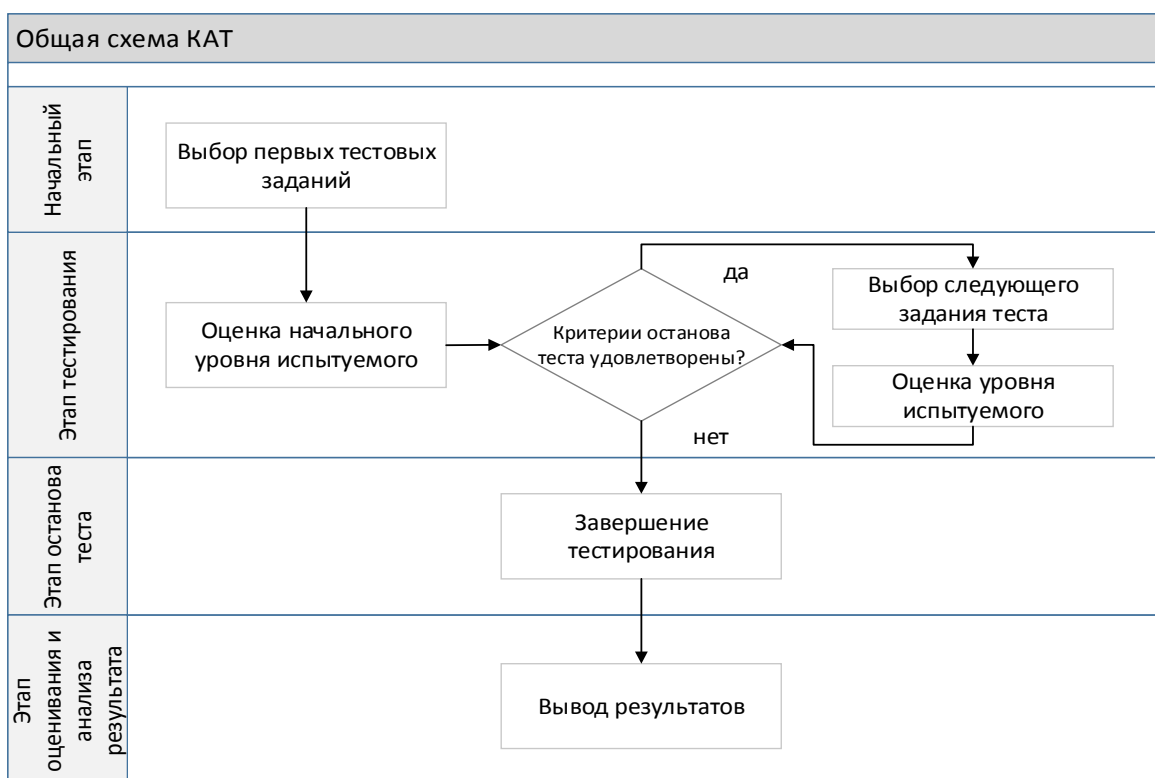


Рис. 1. Схема процесса КАТ

Процесс КАТ может быть реализован только на базе откалиброванного банка тестовых заданий (БТЗ) и может быть разделен на четыре основных этапа. Первый этап является начальным и заключается в выборе одного или нескольких соответствующих заданий в качестве первых заданий в процессе тестирования. Обычно выбирают задания средней сложности. Вторым этапом является непосредственное тестирование, которое заключается в том, что задания

последовательно выбираются из банка тестовых заданий и повторно оценивается уровень способности испытуемых после каждого ответа. На данном этапе, если испытуемый не отвечает на вопрос, то ему дается более легкое задание, если же отвечает, то более сложное. Данные шаги повторяются до тех пор, пока не достигнут критерий останова теста. На этапе останова теста определяются правила завершения тестирования. На заключительном этапе дается окончательная оценка способностей испытуемого и, возможно, другая информация об обучаемом.

Среди существующих в настоящее время систем, позволяющих реализовать адаптивное тестирование, можно выделить две группы: коммерческие и бесплатные. К коммерческим относятся CATSim (Assessment Systems Corporation, 2012 [13]), Adaptest [14], Assessment Center [15], Winsteps [16] и др.. Механизмы, реализованные в этих пакетах, позволяют смоделировать процесс адаптивного тестирования, создать банк тестовых заданий, откалиброванных согласно моделям IRT, получить полную статистику по результатам тестирования. Однако использование таких пакетов ограничено высокой стоимостью. Бесплатные инструментари в основном front-end разрабатывают с применением таких платформ разработки C++, C#, Java и других, а в качестве back-end инструментария используют функции программно-инструментальной среды R, которая является открытой бесплатной средой для обработки статистических данных [17]. К этим системам относятся Firestar [18], CatIrt [19], CatR [20]. Среди последних некоммерческих разработок, позволяющих реализовать механизм адаптивного тестирования с помощью веб-интерфейса, можно выделить пакет mirtCat [21].

Пакет mirtCat основан на функциях таких пакетов, как shiny (позволяющий реализовать механизм ввода-вывода через веб-интерфейс) и mirt (пакет анализа дихотомических и мультиномиальных тестовых заданий на основе одномерных и многомерных моделей теории IRT).

Для реализации системы адаптивного тестирования были проанализированы основные функции пакета, описание которых приведено в таблице 1.

Таблица 1

## Описание функций пакета mirtCat

<b>Функция</b>	<b>Описание</b>
computeCriteria	Возвращает массив оценок критериев для каждого соответствующего задания в тесте
createShinyGUI	Инициализирует графический интерфейс пользователя путем вызова shinyApp. Для размещения приложения в открытый доступ обычно используется ресурс ( <a href="http://www.shinyapps.io/">http://www.shinyapps.io/</a> ). Подробно пакет shiny описан в [22]
extract.mirtCAT	Функция извлекает значения элементов по параметрам (текущее состояние прохождения теста и уровня подготовки испытуемого). Вызывается вместе с функцией customNextItem (см. mirtCAT для подробной информации [21])
findNextItem	Возвращает следующее задание в компьютерном адаптивном тесте. Используется вместе с функциями updateDesign и customNextItem

Функция	Описание
generate.mirt_object	Функция, с помощью которой рассчитывают параметры модели IRT, передающиеся затем в функцию mirtCAT для запуска KAT
getPerson	Эта функция требуется для запуска функции createShinyGUI. Возвращает результаты адаптивного теста для указанного пользователя
generate_pattern	Создает шаблон компьютерного адаптивного теста по различным параметрам для моделирования процесса KAT (подробнее см. в документации [21])
mirtCAT	Основная функция пакета. Предоставляет инструмент для создания HTML-интерфейса для реализации одномерного и многомерного компьютерного адаптивного теста с использованием математической теории IRT
mirtCAT_preamble	Функция вызывается, как внутренняя функция mirtCAT, для более эффективной связи с внешними интерфейсами веб-хостинга (например, <a href="http://www.shinyapps.io/">http://www.shinyapps.io/</a> ). Для получения дополнительной информации о сохранении результатов удаленно при использовании shiny (см.[21])
updateDesign	Функция обновляет объект, полученный в результате вызова findNextItem

Основной функцией пакета является функция mirtCAT, которая по сути предоставляет инструмент для создания HTML-интерфейса для реализации адаптивного теста, используя пакет shiny. Параметры функции определяют методы обработки данных согласно алгоритму KAT (рис.1). Формат функции mirtCAT следующий:

```
mirtCAT(df = NULL, mo = NULL, method = "MAP", criteria = "seq", start_item = 1,
local_pattern = NULL, design_elements = FALSE, cl = NULL, progress = FALSE, primeCluster =
TRUE, design = list(), shinyGUI = list(), preCAT = list(), ...)
```

Описание параметров функции mirtCAT приведено в табл. 2. Для вывода результатов адаптивного тестирования, отображения итоговых статистик прохождения теста и построения графика выбора тестовых заданий испытуемым используют функции print(x, ...), summary(object, sort = TRUE, ...) и plot(x, pick\_theta = NULL, true\_thetas = TRUE, ...) соответственно.

Таблица 2

Описание основных аргументов функции mirtCAT

Аргументы функции	Описание
df	data.frame или список, содержащий данные для указания форматов ответов (Type), вопросов (Question), настроек (Option), правильных ответов (Answer). Все, что необходимо для генерации вопросов, используя GUI Shiny. Например, под типом формата ответа поддерживаются такие типы, как кнопки radio-buttons ('radio'), раскрывающийся список ('select'),

Аргументы функции	Описание
	('checkbox') для выбора множественных ответов, ('text') поле, требующее ввода пользовательского ответа, и многое другое
mo	Аргумент для определения коэффициентов уравнения модели IRT. Используется в том случае, если задания теста подбираются согласно адаптивному алгоритму. В этом случае, если известны параметры популяции испытуемых, то тест может быть сгенерирован на основе банка тестовых заданий функцией generate.mirt_object согласно модели той или иной модели IRT. Если используют неадаптивную форму теста (обычный опросник), то такой аргумент опускается
method	Этот параметр определяет метод прогнозирования скрытой характеристики уровня подготовки $\theta$ (тета). По умолчанию используется метод "MAP", именно с помощью этого метода рассчитывается финальное значение оценки $\theta$ . Также применяются "EAP", "ML", "WLE", "EAPsum", "fixed"
criteria	Критерий выбора следующего вопроса в адаптивном тестировании. Для одномерных адаптивных тестов возможны такие параметры: MI поддерживаются вопросы с максимальной информативностью, MERV с минимальной ожидаемой постериорной дисперсии, MPWI расчет информативности проводится с помощью взвешенного метода максимального правдоподобия, MEI с максимальной ожидаемой информативностью и др. При многомерном адаптивном тестировании используются: Drule для максимального определителя информационной матрицы, Trule для максимальной (потенциально средневзвешенная) трассировки информационной матрицы, Erule для минимального значения информационной матрицы и др. К обоим случаям (одномерного и многомерного тестирования) применимы методы: KL для метода поточечной дивергенции Kullback-Leibler и KLn – поточечный метод Kullback-Leibler с уменьшением значения дельта ( $\delta \cdot \sqrt{n}$ ), где $n$ – это число предыдущих вопросов, на которые есть ответы
start_item	Критерий выбора начального вопроса в тесте. Это одно число, указывающее на номер вопроса, с которого начинается тест. По умолчанию параметр равен единице для вывода первого вопроса в тесте. Если это значение пропущено, то вопрос будет выбран одним из доступных методов, описанных в пункте criteria
local_pattern	Символьная или числовая матрица шаблонов ответа, используемых для запуска приложения KAT без создания графического интерфейса. Если используется data.frame, то определяется матрица символов (с одной строкой), где устанавливается, как отвечал участник на тест, в ином случае возвращается целочисленная матрица для определения ответов множества участников. Если было установлено графическое приложение, то шаблоны будут генерироваться в автономном режиме
design_elements	Логическое значение; возвращает объект, содержащий тест или элементы дизайна. Применяется для функции findNextItem. Логическое значение, указывающее, следует ли возвращать список, который содержит ответы участника
design	Список параметров для контроля адаптивных тестов: критерий остановки адаптивного теста при изменении коэффициента

Аргументы функции	Описание
design	<p><math>\theta</math> (delta_thetas), начальное значение <math>\theta</math> (thetas.start), минимальное количество вопросов, на которые необходимо ответить для завершения теста (min_items), максимальное количество вопросов, на которые необходимо ответить для завершения теста (max_items), Минимальная стандартная ошибка, которая должна быть достигнута для скрытой переменной <math>\theta</math> до того, как тест будет остановлен (min_SEM). Quadpts – число квадратурных точек, используемых как размерность для интеграции (если это необходимо). theta_range – верхняя и нижняя границы для интеграционной сетки theta. Используется совместно с quadpts для создания равномерно распределенной квадратурной сетки. По умолчанию установлено с(-6,6)</p>
shinyGUI	<p>Список GUI параметров для моделирования HTML-интерфейса адаптивного теста: заголовок теста (title), имя автора адаптивного теста (authors), описание трех страниц, указывающее, как использовать GUI (instructions), первая страница shiny GUI (firstpage), текст, отображаемый на странице до начала теста (begin_message), страница для сбора информации об абитуриенте (demographics), максимальное время для прохождения теста, измеряется в секундах (max_time), последняя страница, указывающая, что тест завершен (lastpage). Demographics – личная информационная страница, используемая в GUI для сбора демографической информации. Генерируется с использованием настройки из пакета shiny.</p> <p>demographics_inputIDs – вектор символов, необходимый при использовании входа по демографической особенности пользователя. По умолчанию эта опция отключена.</p> <p>temp_file – символьный вектор, указывающий, где сохраняется временный файл .rds, что содержит информацию с ответами.</p> <p>css – символьная строка, определяющая CSS элементы для модификации в GUI элементов представления.</p> <p>forced_choice – параметр логического типа; указывает, требовать ли ответ на каждый вопрос. Для опросников (на KAT) должно быть установлено только false.</p> <p>password a data.frame object – указание имени пользователя (необязательно) и пароля необходимых перед началом проведения KAT. Возможны такие варианты: No User Information – одна строка в data.frame. Каждый столбец, приведенный в этом случае, будет связан с подходящим паролем для всех тестируемых. Естественно, что если был установлен только один столбец, то это означает один глобальный пароль для всех пользователей. User Information Pairing – многостроковая data.frame, где первый столбец представляет собой имя пользователя и все остальные столбцы, такие же, как описано в первом варианте.</p> <p>stopApp – параметр логического типа; указывает, нужен ли вызов stopApp () после того, как тест был завершен. По умолчанию установлено true. Тем не менее, при размещении приложения на удаленном сервере необходимо установить значение false, чтобы позволить более корректное завершение работы (в этом случае последняя страница будет отображаться до тех пор, пока вкладка браузера не будет закрыта).</p>

Аргументы функции	Описание
	ui – функция shiny UI, используемая для установки интерфейса
preCAT	<p>Это объект списка, который может быть использован для указания предварительного блока КАТ, в котором могут применяться различные свойства теста до начала сеанса КАТ. Если список пустой, то блок preCAT не будет применяться. Все последующие элементы необходимо использовать как входные параметры preCAT :</p> <p>min_items – минимальное число вопросов для администрирования теста до начала сессии КАТ. По умолчанию равно 0.</p> <p>max_items – максимальное число вопросов для администрирования до начала сессии КАТ.</p> <p>criteria – выбор критерия (см. выше). По умолчанию установлено 'random'.</p> <p>method – критерий оценивания (см. выше). Как правило, рекомендуется выбрать метод, который используется с паттернами ответов (все или ничего), такими, как 'EAP' или 'MAP', или в многомерном случае 'DPrule' или 'TPrule'. По умолчанию установлено 'MAP'.</p> <p>response_var – параметр логического типа. Необходимо ли завершить этап preCAT, когда наблюдается изменчивость в структуре ответа (т.е. когда оценка максимального правдоподобия содержит потенциальный оптимум)? По умолчанию установлено false</p>
...	Дополнительные аргументы, которые передаются через mirt, fscores, runApp, или lattice
x object	Объекты класса 'mirtCAT'
sort	Параметр логического типа. Необходимо ли шаблоны ответов (паттернов) сортировать согласно порядку их администрирования? Если установлено – false, то для вопросов, которые не были заданы, будут возвращены значения NAs в необработанных шаблонах ответов (паттернов)
pick_theta	Число, указывающее, для какого показателя $\theta$ строить график (применяется только в случае многомерных тестов). По умолчанию для набора каждой $\theta$ строится один график, но для построения графика только для первого фактора необходимо установить pick_theta = 1
SE	Размеры стандартных ошибок для графика. По умолчанию установлено 1, и, следовательно, этот график представляет собой график погрешностей. Для того, чтобы получить 95%-ный интервал, использовать SE = 1.96 (для z-распределения)
main	Заголовок графика. По умолчанию для 'CAT Standard Errors' или 'CAT ##% Confidence Intervals' в зависимости от входа SE
par.strip.text par.settings	Аргументы графики, передаваемые через lattice

Полное описание всех функций и их аргументов можно найти в инструкциях к пакетам mirtCat и shiny [21].

## 2. Реализация процесса компьютерного адаптивного тестирования с помощью пакета mirtCAT

С использованием пакета был разработан тест по дисциплине «Информатика» для студентов первых курсов Национального аэрокосмического университета «ХАИ», позволяющий проводить адаптивное тестирование.

На первом этапе были загружены дополнительные пакеты, содержащие набор функций, которые расширяют базовые возможности языка R:

```
library('mirtCAT')  
library('data.table')
```

Для того, чтобы при записи данных в data.frame каждое символьное значение в таблице воспринимался как значение, а не фактор, необходимо присвоить логическому параметру stringsAsFactors значение false:

```
options(stringsAsFactors = FALSE)
```

На следующем этапе необходимо зафиксировать начальное значение для генерации случайных чисел:

```
set.seed(1234)
```

Затем считываются данные. Функция data.table::fread() считывает данные из файла по указанному пути к файлу с тестом и создает из него таблицу данных (data.frame). Значение аргумента sep = "/" предполагает, что значения переменных в считываемом файле разделены символом "/". Значение параметра header = TRUE позволяет записывать в заголовки столбцов первую строку из считываемого файла. Данные сохраняются как объект класса, что позволяет в последующем реализовать объектно-ориентированный доступ к данным:

```
x<- data.table::fread("E:/R/informTest.txt", sep="/", header=TRUE)  
x <- as.data.frame(x)  
class(x)
```

Далее задается количество вопросов теста с помощью метода nrow():

```
nitems <- nrow(x)
```

Моделируются шаблоны ответов согласно моделям IRT. Сначала определяются заголовки столбцов в таблицы с моделируемыми ответами. Название заголовка таблицы – 'Item.' + номер вопроса в тесте. Затем генерируется матрица a с параметрами тестовых заданий (эта матрица отвечает за такой параметр модели IRT, как дифференцирующая способность). Для генерации используем функцию логнормального распределения (rlnorm). Параметры nitems – это число наблюдений (в нашем случае 10), 0.2 и 0.3 – параметры для закона (значения среднего и стандартного отклонения для распределения на логарифмической шкале).

```
itemnames <- paste0('Item.', 1:nitems)
```



```
a <- matrix(rlnorm(nitems, .2, .3))
```

На следующем шаге задаем начальные коэффициенты матрицы  $d$  (эта матрица отвечает за такой параметр модели IRT, как сложность (трудность) тестового задания. В идеальном случае значения матрицы равны 0,5, что означает, что кандидат с определенным уровнем знаний ответит на данный вопрос верно с 50%-ной вероятностью. Для данной матрицы используем функцию генерации ответов на основе нормального закона распределения (`rnorm`), где `nitems` – число вопросов, среднее и стандартное отклонение распределения 0 и 1 соответственно (установлены по умолчанию).

```
d <- matrix(rnorm(nitems))
```

С помощью функции `simdata()` моделируем шаблоны ответов для моделей MIRT на основе матриц  $a$  и  $d$ , где  $N$  – объем выборки, `itemtype` – способ выбора начальных значений в новой модели, для оптимальной скорости расчетов выберем метод 'dich'.

```
dat <- simdata(a, d, N = 501, itemtype = 'dich')
```

На следующем этапе определяем модель, согласно которой будет реализован процесс КАТ. В данной работе использовалась однофакторная модель IRT. Выбор данной модели обусловлен тем, что тесты по информатике являются гомогенными, т.е. ответы на тестовые задания нечасто коррелируют между собой. Вероятность ответа на  $j$ -е задание согласно данной модели определяется как.

$$P(X_j = 1 | \theta) = g_j + (u_j - g_j) \frac{1}{1 + \exp(-a_j(\theta - d_j))}, \quad (1)$$

где  $X_j$  – результат ответа на тестовое задание: 1 – верный ответ, 0 – неверный ответ,  $g_j$  – вектор уровней способностей обучаемых,  $a_j$  – параметр дискриминации тестового задания,  $d_j$  – параметр сложности тестового задания,  $u_j = 1, g_j = 0$  – параметры угадывания и невнимательности соответственно.

На данном шаге используем функцию `mirt`, которая позволяет сгенерировать шаблоны ответов согласно модели (1). Аргументами функции являются `data` – шаблон ответов для модели MIRT (`data=dat`), `model` – определяет количество скрытых факторов (`model` равно 1), `itemtype` – тип логистической модели IRT (в данном случае 2PL). Подробности о пакете `mirt` можно найти в [27].

```
mod <- mirt(data = dat, model = 1, itemtype = '2PL')
```

Извлекаем показательные коэффициенты из модели `mod` с помощью функции `coef`, где `simplify` – логический параметр, указывающий, как извлекать коэффициенты, в данном случае с упрощением:

```
coef(mod, simplify=TRUE)
```

Из таблицы данных, в которую был записан тест, создадим векторы: Question – вектор с вопросами, Option – вектор с возможными ответами для каждого вопроса, Answer – вектор с правильными ответами для каждого вопроса:

```
questions <- answers <- character(nitems)
options <- matrix("", nitems, 4)
spacing <- floor(d - min(d)) + 1
for (i in 1:nitems) {
  ans <- x[i, 2]
  questions[i] <- paste0(x[i, 1])
  answers[i] <- as.character(ans)
  v <- sample(1:4, 4, replace = F)
  ch <- sample(c(-4:-1, 1:4) * spacing[i, ], 4)
  ch[v[1]] <- ans
  ch[v[2]] <- x[i, 3]
  ch[v[3]] <- x[i, 4]
  ch[v[4]] <- x[i, 5]
  options[i, ] <- as.character(ch)
}
```

Укажем параметры для функции shinyGUI, которая создает визуальную оболочку адаптивного теста.

```
title <- "Многомерный адаптивный тест"
authors <- " М.С. Мазорчук, В.С. Добряк, А.Р. Кочура"
instructions <- c("Инструкции:", "Для ответа на следующий вопрос щелкните по
кнопке ниже", "Далее")
firstpage <- list(h2("Тест по информатике"), h5("Дайте ответы на вопросы в меру
своих возможностей. Результаты теста полностью анонимны, используются в целях
исследования."))
begin_message <- "Если Вы готовы начать тест, щелкните по кнопке ниже 'Далее'."
lastpage <- function(person)
  return(list(h5("Спасибо за завершение теста. Пожалуйста, щелкните 'Далее',
чтобы сохранить Ваш результат")))
shinyGUI_list <- list(title = title, authors = authors, instructions=instructions, firstpage =
firstpage, begin_message = begin_message, lastpage = lastpage, stopApp = TRUE)
df <- data.frame(Question = questions, Option = options, Answer = answers, Type =
"radio")
```

Определим значения для аргумента design, в котором содержится список параметров для контроля адаптивного теста, где delta\_thetas – критерий остановки адаптивного теста при изменении коэффициента  $\theta$ , min\_items – минимальное количество вопросов, на которые необходимо ответить для завершения теста:

```
design <- list(delta_thetas = 0.095, min_items = 2)
```

На последнем этапе смоделируем многомерный адаптивный тест с параметрами: `df` – data.frame, содержащий векторы с данными для реализации тестирования (вопросы, варианты ответов), `mo` – модель факторного анализа (определенная на предыдущих шагах), `method` – критерий обновления показателей теста в момент прохождения теста (именно с помощью этого метода рассчитывается финальное значение  $\theta$ , в данном случае с помощью метода модельного базиса 'MAP', который будет максимизировать функцию апостериорного распределения уровня способности), `criteria` – критерий выбора последующих тестовых заданий в многомерном адаптивном тесте (в данном случае критерий выбора – 'Drule', ориентированный на поиск максимального определителя информационной матрицы), `shinyGUI` – визуальная оболочка адаптивного теста.

```
result <- mirtCAT(df = df, mo = mod, method = 'MAP', criteria = 'Drule', shinyGUI = shinyGUI_list, design = design)
```

Вывод результата для многомерного адаптивного теста:

```
print(result)
```

Вывод графика, по которому можно отследить изменения  $\theta$  в зависимости от ответов на тестовые задания (координата  $x$  – номер задания, координата  $y$  – диапазон значений  $\theta$ ):

```
plot(result)
```

### 3. Результаты моделирования КАТ

Интерфейсы пользователя, отображающие этапы работы с тестом по информатике, изображены на рис. 2 – 5.

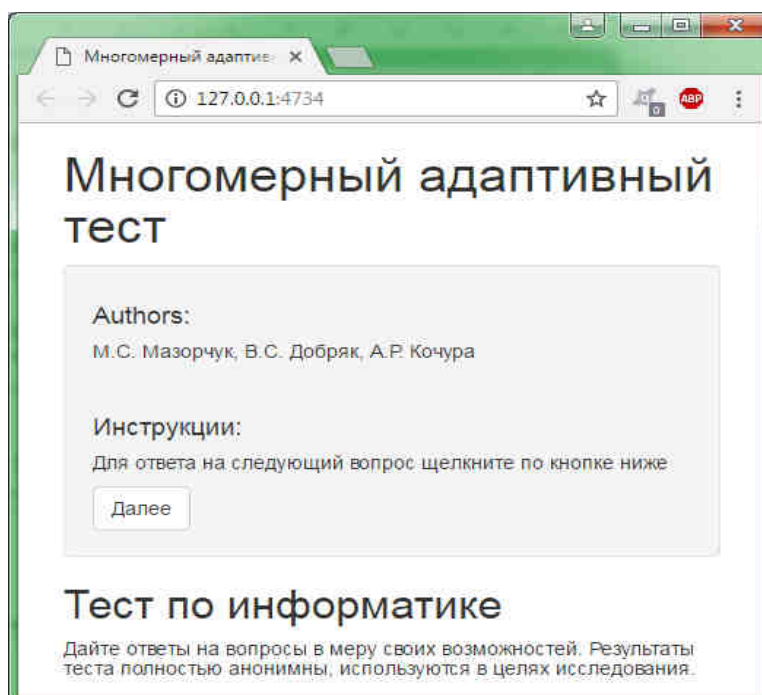


Рис. 2. Первая страница с описанием адаптивного теста



Рис. 3. Страница теста с инструкциями

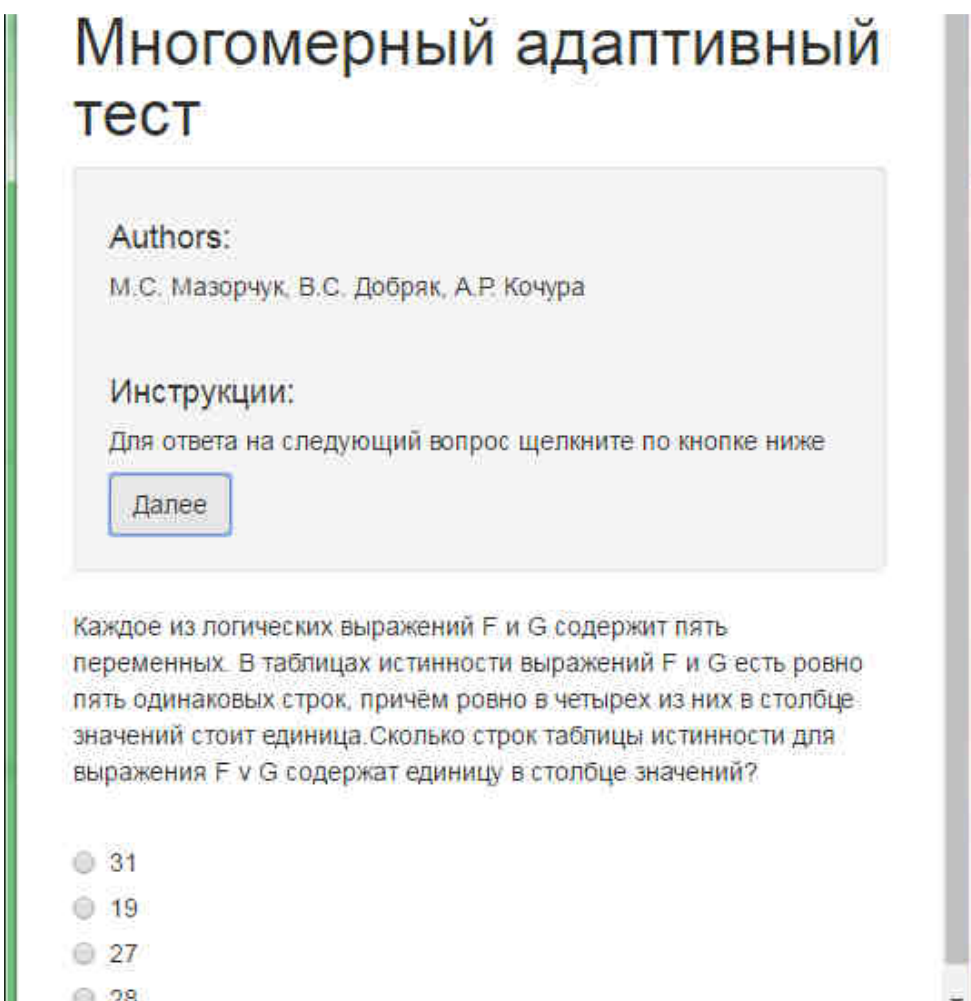


Рис. 4. Страница с одним из вопросов адаптивного теста

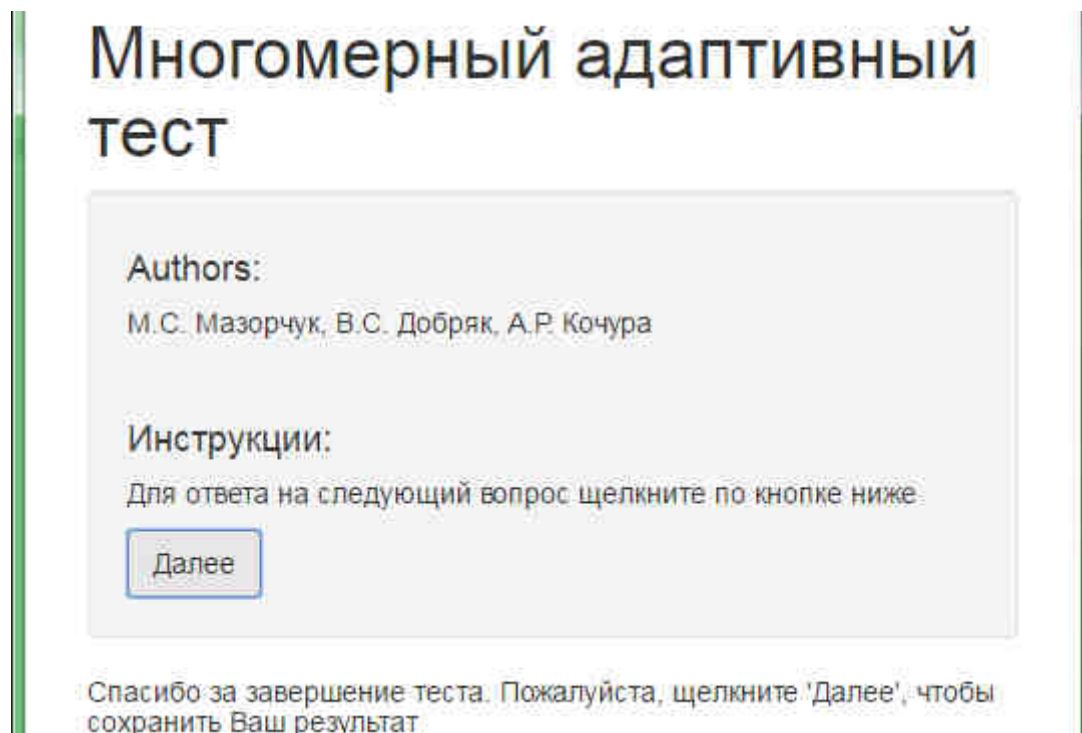


Рис. 5. Последняя страница, указывающая, что адаптивный тест завершен

Далее осуществляется сохранение теста и результаты можно просмотреть в среде R. Вывод результатов для многомерного адаптивного тестирования изображен на рис. 6 – 9.

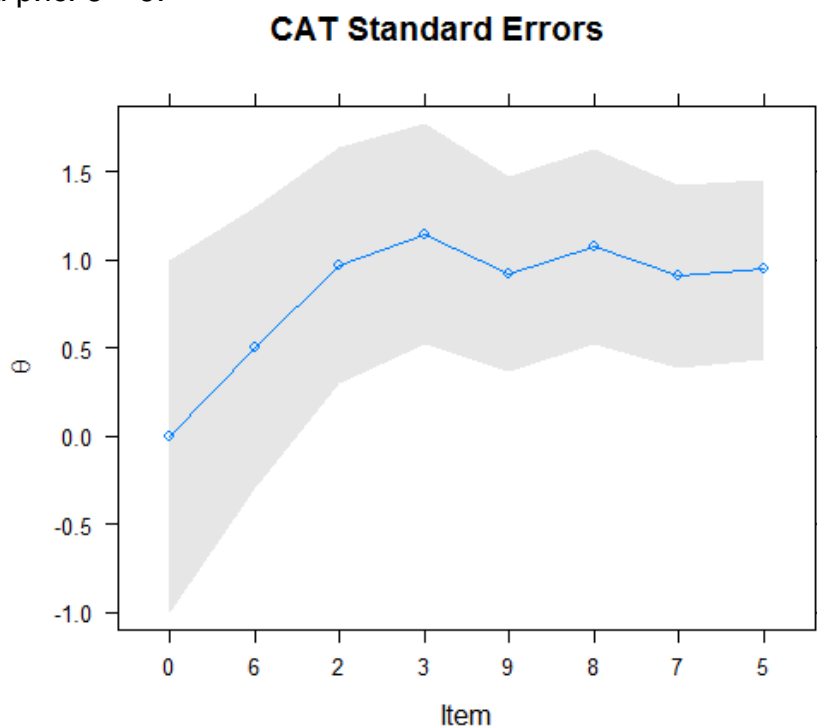


Рис. 6. График, по которому можно отследить изменения  $\theta$  в зависимости от ответов на вопросы

```
$scored_responses  
[1] 1 1 1 0 1 0 1
```

Рис. 7. Последовательность ответов на вопросы в адаптивном тесте  
(0 – не правильно, 1 – правильно)

```
$thetas_history  
Theta_1  
[1,] 0.0000000  
[2,] 0.5008426  
[3,] 0.9660461  
[4,] 1.1485124  
[5,] 0.9166839  
[6,] 1.0748985  
[7,] 0.9083317  
[8,] 0.9466111
```

Рис. 8. Обновления показателей  $\theta$  в момент прохождения теста

```
> print(result)  
n.items.answered Theta_1 SE.Theta_1  
7 0.9466111 0.5101026
```

Рис. 9. Результат многомерного адаптивного теста (значения  $\theta$  – уровня подготовки испытуемого (Theta\_1) и средней ошибки (SE.Theta\_1))

## Выводы

Таким образом, изложенная методика компьютерного адаптивного тестирования в среде R позволяет разработать адаптивный тест (на основе шаблонов тестовых заданий, откалиброванных согласно многомерной модели IRT) и реализовать его через HTML-интерфейс. Технология разработки теста не является полностью прозрачной, так как используемые функции пакетов *mirtCAT*, *shiny*, *mirt* и других требуют определенного понимания процесса КАТ и моделей, которые используются для задания критериев начала теста, процесса тестирования и останова. Однако гибкий функционал позволяет реализовать процесс КАТ практически с любыми начальными условиями. Использование функций пакета *mirtCAT* позволит реализовать функциональную часть систем КАТ, разработанных в других программных средах.

## Список литературы

1. Clark, Ruth C. e-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning [Text] / Ruth C. Clark, Richard E. Mayer. – 4th Edition. – 528 p.
2. Jordan, K. Massive Open Online Course Completion Rates Revisited: Assessment, Length and Attrition [Electronic resource] // The international review of research in open and distributed learning. – 2015. – Way of access: <http://www.irrodl.org/index.php/irrodl/article/view/2112/3340> June, 2015. – Title from the screen.
3. Ying, C. Computerized adaptive testing [Text] / C. Ying // New developments and applications, University Of Illinois At Urbana-Champaign, 2008. – 92 p.

4. Wainer, Howard Computerized Adaptive Testing: A Primer. [Text] / Howard Wainer. – MahWah, 2000. – 2 Edition. – 364 p.
5. Данг, Х. Ф. Метод разработки алгоритмов адаптивного тестирования [Текст] / Х. Ф. Данг, В. А. Камаев, О. А. Шабалина // Известия Волгоградского государственного технического университета. – 2012. – Вып. – № 13, Т. 4. – С.107-112.
6. Baker, Frank B. The Basics of Item Response Theory by Frank B. Baker [Text] / Frank B. Baker // ERIC Clearinghouse on Assessment and Evaluation, 2001. – 172 p.
7. T-SKIRT [Электронный ресурс]: Online Estimation of Student Proficiency in an Adaptive Learning System. – Режим доступа: [https://dsp.rice.edu/sites/dsp.rice.edu/files/tskirt\\_camera\\_ready\(1\)\\_3](https://dsp.rice.edu/sites/dsp.rice.edu/files/tskirt_camera_ready(1)_3). – 25.12.2016 г.
8. Дубан, Р. М. Слайн-моделі профілів складності питань та знань респондентів в тестовому контролі знань [Текст] / Р. М. Дубан, І. В. Шелевицький // Автоматизированные системы управления и приборы автоматики: всеукр. межведомств. научно-техн. сб. – Х., – 2011. – Вып. 156. – С. 71-77.
9. Дубан, Р. М. Ермітів кубічний сплайн з фіксованими краями – модель профілів IRT [Текст] / Р. М. Дубан // Комп'ютерно-інтегровані технології: освіта, наука, виробництво: науковий журнал. – Луцьк. – 2012. – № 10. – С. 115-122.
10. Дубан, Р. М. Застосування інтегралу від лінійного B-сплайна в якості моделі IRT [Текст] / Р. М. Дубан, І. Ф. Бойко // Електроніка та системи управління. – К. – 2012. – №1(31). – С. 131-138.
11. Ronald, K. Comparison of classical test theory and item response theory and their applications to test development [Электронный ресурс] // Educational Measurement: issues and practice. – 1993. – Режим доступа: [http://www.internationalgme.org/Resources/Pubs/ITEMS\\_Module\\_16.pdf](http://www.internationalgme.org/Resources/Pubs/ITEMS_Module_16.pdf) – Title from the screen.
12. Loken, E. Estimation of a Four-Parameter Item Response Theory Model [Text] / E. Loken, K. L. Rulison // British Journal of Mathematical and Statistical Psychology. – 2010. – № 63. – P. 509–525.
13. Assessment systems for good measure [Электронный ресурс]: CATSim. – Режим доступа: <http://www.assess.com/product/catsim/>. – 25.12.2016 г.
14. Vector Psychometric Group [Электронный ресурс]: Adaptest®. – Режим доступа: [https://www.vpgcentral.com/software/computerized\\_adaptive\\_testing/](https://www.vpgcentral.com/software/computerized_adaptive_testing/). – 25.12.2016 г.
15. Assessment CenterSM [Электронный ресурс]: WHAT IS ASSESSMENT CENTER SM. – Режим доступа: <https://www.assessmentcenter.net>. – 25.12.2016 г.
16. WINSTEPS [Электронный ресурс]: WINSTEPS & Facets Rasch Software. – Режим доступа: <http://www.winsteps.com/>. – 25.12.2016 г.
17. CRAN [Электронный ресурс]: The Comprehensive R Archive Network. – Режим доступа: <https://cran.r-project.org/>. – 25.12.2016 г.
18. Seung, W. Choi Firestar: Computerized Adaptive Testing Simulation Program for Polytomous Item Response Theory Models [Электронный ресурс] / Choi W. Seung. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.457.6717&rep=rep1&type=pdf>. – 25.12.2016 г.
19. Package 'catIrt' [Электронный ресурс]: An R Package for Simulating IRT-Based Computerized Adaptive Tests. – Режим доступа: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.368.67&rep=rep1&type=pdf>. – 25.12.2016 г.
20. Package 'catR' [Электронный ресурс]: Generation of IRT Response Patterns under Computerized Adaptive Testing. – Режим доступа: <https://cran.r->

project.org/web/packages/catR/catR.pdf. – 25.12.2016 г.

21. Package 'mirtCAT' [Электронный ресурс]: Computerized Adaptive Testing with Multidimensional Item Response Theory. – Режим доступа: <https://cran.r-project.org/web/packages/mirtCAT/mirtCAT.pdf>. – 25.12.2016 г.

Поступила в редакцию 27.01.2017

## Технологія адаптивного тестування у середовищі R

У цей час багато уваги приділяється питанням дистанційного навчання. Актуальним є завдання проектування і розроблення систем адаптивного тестування, що дозволяють не тільки ефективно оцінювати рівень навчання, але й організувати гнучкий процес дистанційного навчання, орієнтованого на індивідуальні здібності учня. Існуючі системи комп'ютерного адаптивного тестування є дорогими. Безкоштовне програмно-інструментальне середовище R дає можливість реалізувати ряд функцій адаптивного тестування, а пакет mirtCat, оснований на функціях таких пакетів, як shiny і mirt, дозволяє реалізувати механізм введення-виведення тестових завдань через веб-інтерфейс і робить можливим проведення аналізу тесту на основі одновимірних і багатовимірних моделей теорії IRT.

**Ключові слова:** дистанційне навчання, масові онлайн-курси, комп'ютерне адаптивне тестування, тестові завдання, якість тесту.

## Adaptive Testing Technology In R

Currently much attention is paid to distance learning. The problem of design and development of adaptive testing systems that are effective not only in assessing the level of training, but in organizing a flexible process of distance learning based on the student's individual abilities is relevant. Existing systems of computer adaptive testing are quite expensive. Open-source software environment R allows implementing a number of features of adaptive testing and mirtCat package based on functions of such packages as shiny and mirt, allows realizing the mechanism of input-output test items through the web interface, and doing a test analysis possible on the basis of one-dimensional and multidimensional IRT models theory.

**Key words:** distance learning, massive open online courses, computer adaptive testing, test items, test quality.

### Сведения об авторах:

**Мазорчук Мария Сергеевна** – к.т.н., доцент кафедры информатики Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ»; e-mail: mazorchuk\_mary@inbox.ru.

**Добряк Виктория Сергеевна** – к.т.н., доцент кафедры информатики Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ»; e-mail: viktoriya.dobryak@gmail.com.

**Кочура Анастасия Романовна** – студентка 365м группы Национального аэрокосмического университета им. Н. Е. Жуковского «ХАИ»; e-mail: skwep@gmail.com.