



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.01 Информатика и вычислительная техника

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Анализ систем тестирования знаний языков
программирования

Студент

ИУ6-72Б

(Группа)

(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Руководитель

(Подпись, дата)

Т.А. Ким

(И.О. Фамилия)

Консультант

(Подпись, дата)

С.С. Семенов

(И.О. Фамилия)

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

З А Д А Н И Е
на выполнение научно-исследовательской работы

по теме Анализ систем тестирования знаний языков программирования

Студент группы ИУ6-72Б

Астахов Сергей Викторович

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)

исследовательская

Источник тематики (кафедра, предприятие, НИР) кафедра

График выполнения НИР: 25% 4 нед., 50% 7 нед., 75% 11 нед., 100% 14 нед.

Техническое задание: выполнить анализ систем тестирования знаний языков программирования, осуществить выбор способов тестирования знаний языков программирования и на их основе сформировать функциональные требования к системе тестирования знаний языков программирования

Оформление научно-исследовательской работы:

- 1) Расчетно-пояснительная записка на 25-30 листах формата А4.
- 2) Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)
Необходимый иллюстративный графический материал включить в качестве рисунков в расчетно-пояснительную записку
- 3) Приложение А. Техническое задание на ВКРБ на 5-8 листах формата А4.

Дата выдачи задания « 1 » сентября 2021 г.

Руководитель

(Подпись, дата)

Т.А. Ким

(И.О. Фамилия)

Студент

(Подпись, дата)

С.В. Астахов

(И.О. Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

РЕФЕРАТ

Отчет X с., X рис., X. ист., X прил.

ТЕСТИРОВАНИЕ ЗНАНИЙ, ЯЗЫКИ ПРОГРАММИРОВАНИЯ,
ОБРАЗОВАТЕЛЬНЫЕ ПОРТАЛЫ, ХАКАТОНЫ, CTF, LMS.

Объектом исследования являются системы тестирования знаний, ориентированные на передачу и проверку знаний в области информационных технологий и, в частности, языков программирования.

Цель работы — получение описания бизнес-процессов, (idef0) архитектурной модели (с4) и функциональных требований к подсистеме тестирования знаний языков описания аппаратуры.

В процессе работы был проведен анализ бизнес-процессов и принципов организации имеющихся систем тестирования знаний в области информационных технологий. Затем был проведен анализ полученных моделей бизнес-процессов, в результате доработки которых было получено описание бизнес-процессов, а затем и архитектурная модель разрабатываемой системы.

Актуальность тут или дальше?

СОДЕРЖАНИЕ

ВЕДЕНИЕ	6
1 Проблематика и методология	7
1.1 Проблематика изучения языков описания аппаратуры	7
1.2 Методология функционального моделирования IDEF0	8
1.3 Модель С4	9
2 Методы тестирования знаний	10
2.1 Классификация методов	10
2.2 Классические методы тестирования знаний	11
2.2.1 Тестирование с ответом в закрытой форме	11
2.2.2 Тестирование с коротким ответом и ответом в форме эссе	13
2.3 Тестирование на написание программы	15
2.3.1 Проверка по референсным значениям	15
2.3.2 Автоматическое тестирование на проверяющей стороне + ?С4?	17
? 2.3.3 Про систему Paradigm ?	18
2.4 Сравнительный анализ методов	18
3 Система оценивания + Paradigm CTF + ?статистика?	18
4 Бизнес-процессы и архитектура приложения мечты	18
ЗАКЛЮЧЕНИЕ	19
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	20

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

Бизнес процесс

Язык описания аппаратуры

ПЛИС

HDL

VCD

Tetsbench

IDEF0

C4

Язык описания предметной области / DSL

БД

ПО

СДО

СТФ

Unit-тестирование

Интеграционное тестирование

LMS

CAT

ВЕДЕНИЕ

— актуальность, высокий порог входа, моральный барьер, сложный софт, мало курсов, IT – круто, ?санкции?, профориентация

Основной целью научно-исследовательской работы является формирование функциональных требований, описание бизнес-процессов и архитектурной модели программной подсистемы тестирования знаний языков описания аппаратуры.

Такая подсистема предназначена для интеграции в архитектуру образовательной платформы для предоставления функций тестирования знаний (т.е. для проведения контрольных мероприятий), основные из которых:

- добавление, редактирование, удаление заданий;
- проверка правильности решения;
- хранение и обработка информации о решенных заданиях и др.

Актуальность исследования обусловлена тем, что несмотря на активный в последние годы рост популярности и числа образовательных онлайн-платформ и курсов, связанных с изучением информационных технологий, вплоть до настоящего момента существует дефицит ресурсов, направленных на практическое освоение языков описания аппаратуры.

Примерный план:

- 1 Проблематика освоения HDL, бла-бла-бла про системный и функциональный анализ, iverilog, c4
- 2 Аналоги, курсы, их модели iverilog и возможно c4
- 3 Обобщение и сравнительный анализ — критерии, достоинства и недостатки
- 4 Выявление best practices, устранение недостатков, специфика работы с HDL-языками (временные диаграммы и тест-бенчи, ?синтезируемый код?), iverilog и c4 системы нашей мечты

1 Проблематика и методология

1.1 Проблематика изучения языков описания аппаратуры

Несмотря на наличие большого числа теоретических материалов на данную тематику, наблюдается дефицит и низкое качество организации ресурсов, ориентированных на практическое освоение языков описания аппаратуры. Абсолютное большинство таких ресурсов предоставляет лишь теоретические данные и набор практических упражнений, которые пользователю предлагается выполнить самостоятельно в стороннем программном обеспечении.

Такой подход может быть довольно сложен для новичка в силу описанных ниже проблем.

Первая проблема — установка стороннего программного обеспечения. Наиболее часто используемые для работы с HDL среды: Quartus и Xilinx. Обе они требуют большого объема как постоянной, так и оперативной памяти. Кроме того, для приобретения начальных навыков функциональность этих сред избыточна, так как значительная ее часть ориентирована на адаптацию проекта под конкретную аппаратную базу для дальнейшей прошивки в ПЛИС. Избыточная функциональность (с точки зрения рассматриваемой задачи) требует дополнительных вычислительных ресурсов и усложняет работу пользователя с этими средами.

Альтернативой Xilinx и Quartus являются такие инструменты, как Icarus Verilog. Это легковесная среда симуляции и синтеза устройств, описанных на языке Verilog. Взаимодействие с пользователем осуществляется через консольный интерфейс, результаты симуляции записываются в VCD-файл и затем отображаются графически через такие утилиты, как GTKWave. Основным недостатком в этом случае являются непривычный для новичка консольный интерфейс.

Вторая проблема — отсутствие внешнего контроля и системы оценивания. Безусловно, большинство людей в состоянии объективно оценить

правильность функционирования описанного ими устройства по временным диаграммам, полученным в результате запуска Testbench-файлов, прикрепленных к заданию. Однако, обучение на основе только таких заданий не позволяет закрепить теоретические знания, которые можно было бы проверить, например, тестовыми заданиями. Кроме того, такая система усложняет контроль человека за освоением курса в целом, утрачивается ощущение объективности оценки собственного прогресса, ухудшается качество обучения [1].

Стоит отметить, что полноценное освоение языков описания аппаратуры в принципе затруднительно без знаний в области цифровой схемотехники, архитектуры ЭВМ и т.п. Однако, цель образовательных платформ, посвященных этой тематике состоит прежде всего именно в формировании базовых знаний и навыков работы с HDL для людей, интересующихся HDL, например, в качестве хобби или с целью продолжить обучение в ВУЗе и т.п.

Данная научно-исследовательская работа позволяет подойти к решению описанных выше проблем посредством формирования функциональных требований, описания бизнес-процессов (idef0) и архитектурной модели (c4) подсистемы тестирования знаний языков описания аппаратуры, полученных на основе анализа имеющихся систем тестирования знаний языков программирования.

1.2 Методология функционального моделирования IDEF0

IDEF0 — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов.

Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ) [2].

Диаграмма IDEF0 выглядит как набор функциональных блоков. Каждая сторона блока имеет свое назначение: левая — для входов, правая — для выходов, верхняя — для управления, нижняя — для механизмов (рисунок 1).

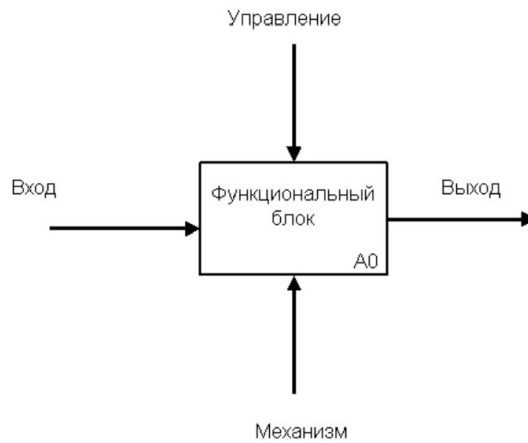


Рисунок 1 — функциональный блок в нотации IDEF0

Основным достоинством IDEF0 является высокая степень формализации, которая позволяет легко интегрировать фрагменты разрабатываемой модели в единую систему [3].

В рамках данной научно-исследовательской работы IDEF0 используется для описания бизнес-процессов, реализующих различные методы тестирования знаний. Диаграммы получены с помощью DSL IDEF0-SVG, в силу чего наблюдаются некоторые несущественные упрощения нотации.

1.3 Модель С4

Модель С4 — простой метод графической записи для моделирования архитектуры программных систем. Он основан на структурной декомпозиции системы на контейнеры и компоненты и опирается на существующие методы моделирования, такие как Unified Modeling Language (UML) или ER-модель (ERD), для более детальной декомпозиции архитектурных блоков.

В модели С4 выделяют 4 уровня диаграмм [4]:

- диаграммы контекста (уровень 1): показывают систему в масштабе ее взаимодействия с пользователями и другими системами;
- диаграммы контейнеров (уровень 2): разбивают систему на взаимосвязанные контейнеры. Контейнер - это исполняемая и развертываемая подсистема;
- диаграммы компонентов (уровень 3): разделяют контейнеры на взаимосвязанные компоненты и отражают связи компонент с другими контейнерами или другими системами;

- диаграммы кода (уровень 4): предоставляют дополнительные сведения о дизайне архитектурных элементов, которые могут быть сопоставлены с программным кодом. Модель C4 на этом уровне опирается на существующие нотации, такие как UML.

Для уровней с 1 по 3 в модели C4 используются 5 основных элементов диаграмм: пользователи, программные системы, контейнеры, компоненты и отношения.

Модель C4 облегчает совместную работу над созданием архитектуры ПО и доработку архитектуры в контексте команд разработки работающих с применением гибкой методологии разработки, в которой более формальные методы документирования и предварительное архитектурное проектирование нежелательны.

В данной работе используется диаграмма контейнеров (уровень 2), позволяющая в общих чертах описать архитектуру проектируемой программной подсистемы тестирования знаний языков описания аппаратуры.

2 Методы тестирования знаний

2.1 Классификация методов

Перед моделированием бизнес-процессов, реализующих различные методы тестирования знаний, необходимо ввести их классификацию.

В качестве основы была взята подобная классификация для СДО Moodle [5]. Она была дополнена с учетом функциональных особенностей таких СДО, как Huawei University, Coursera, Stepik, Ethernaut, а также CTF Paradigm [6]. Сформированная классификация приведена в таблице 1.

Таблица 1 — классификация методов тестирования знаний

№	Тип	Подтип	Пояснение
1	Тестирование с ответом в закрытой форме	1.1 Выбор одного ответа 1.2 Выбор множественных ответов 1.3 Сопоставление	-
2	Тестирование с	2.1 С автоматической проверкой	-

	коротким ответом	2.2 С проверкой преподавателем 2.3 С перекрестной проверкой	
3	Тестирование с ответом в форме эссе	3.1 С проверкой преподавателем 3.2 С перекрестной проверкой	-
4	Тестирование на написание программы	4.1 С проверкой по референсным значениям 4.2 Автоматическое тестирование на проверяющей стороне 4.3 Другие	-

2.2 Классические методы тестирования знаний

2.2.1 Тестирование с ответом в закрытой форме

Тестирование с ответом в закрытой форме применяется практически во всех системах тестирования знаний. IDEF0-модель соответствующего бизнес-процесса представлена на **рисунке 1**.

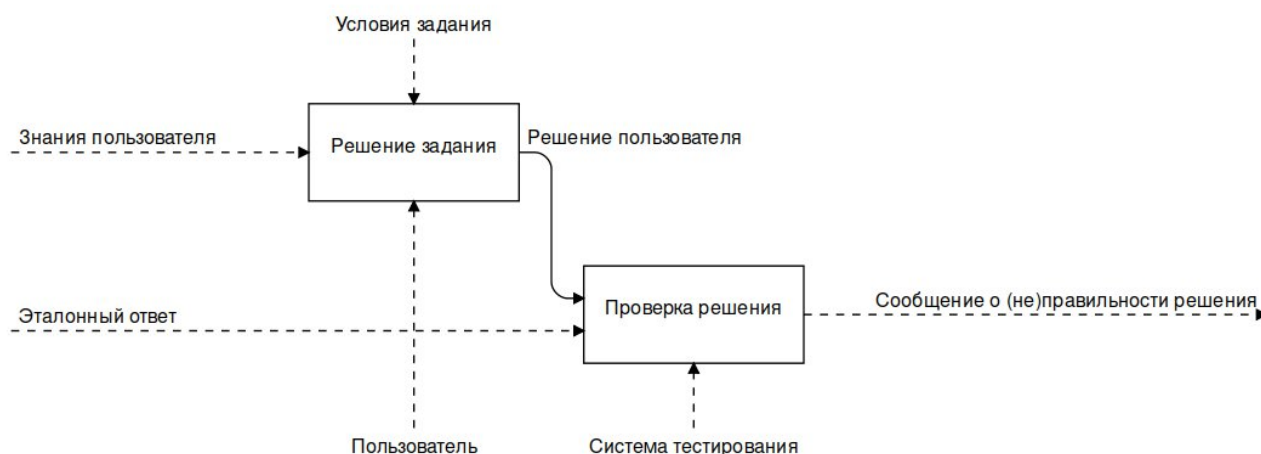


Рисунок 1 — бизнес-процесс тестирования с ответом в закрытой форме

Основным недостатком такой реализации тестирования с ответом в закрытой форме является невозможность получить содержательную обратную связь в случае неверного решения. Возможные формы обратной связи для различных подтипов заданий с закрытым ответом показаны в таблице 2.

Таблица 2 — формы обратной связи для тестирования с ответом в закрытой форме

Подтип тестирования	Форма обратной связи
Выбор одного ответа	Пояснение причин некорректности ответа
Выбор множественных ответов	Сообщение о выборе избыточного/недостаточного числа вариантов
Сопоставление	Сообщение о количестве неправильно выбранных пар
	Подсветка некорректно выбранных пар

В случае заданий на выбор множественных ответов сообщение о выборе избыточного/недостаточного числа вариантов неинформативно и не позволяет пользователю повторно проанализировать задание с его учетом. При этом такой вид обратной связи позволяет пользователю сократить число вариантов для перебора ответов при повторном решении задания. По этим причинам использование обратной связи в заданиях этого подтипа не всегда желательно.

В случае заданий на сопоставление сообщение о количестве неправильно выбранных пар менее информативно, но не сокращает число вариантов перебора ответа. Подсветка некорректно выбранных пар содержит полезную для повторного анализа задания информацию, но сокращает число вариантов перебора ответа.

Еще одной проблемой тестирования с ответом в закрытой форме, уже затронутой выше, является проблема перебора ответов. Данная проблема не возникает в случае проведения контрольных мероприятий, где количество попыток прохождения тестирования ограничено. Однако, в случае открытых онлайн-курсов, число попыток прохождения тестирования, как правило, не

ограничивается. В таком случае одним из решений проблемы является ограничение времени до возможности повторно пройти тестирование.

Бизнес-процесс прохождения тестирования с ответом в закрытой форме с учетом предложенных улучшений представлен на **рисунке 2**.

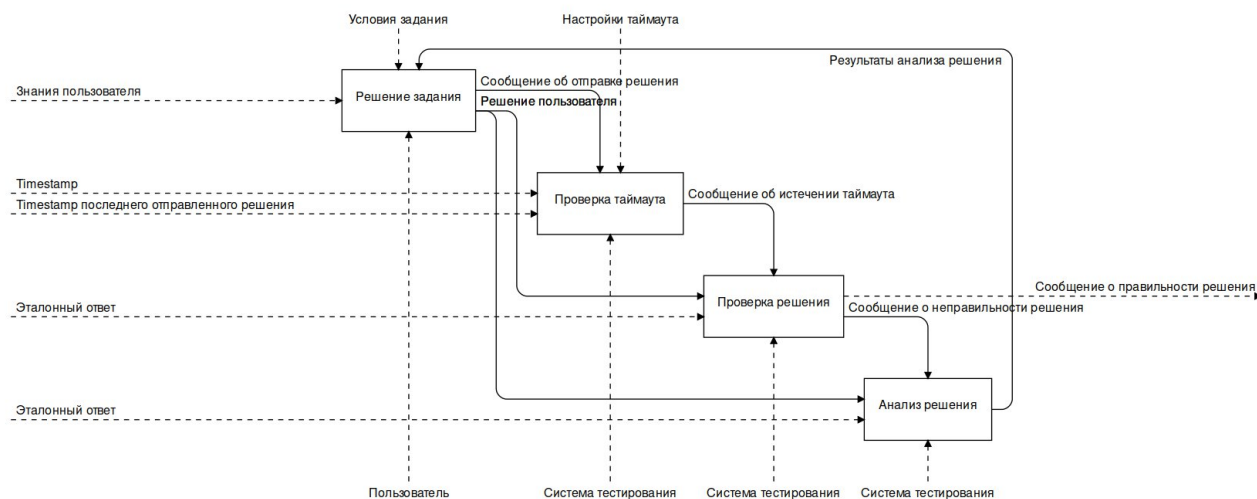


Рисунок 2 — усовершенствованный бизнес-процесс тестирования

2.2.2 Тестирование с коротким ответом и ответом в форме эссе

Тестирование с коротким ответом может быть проверено автоматически, преподавателем или участниками тестирования перекрестно. Автоматическая проверка зачастую не учитывает все возможные формы слова, синонимы, грамматические ошибки в ответе и т.п. Однако, она не требует вовлечения преподавателя или перекрестной оценки, которая может быть субъективной в силу тех или иных причин. Поэтому, в случае открытых онлайн-курсов, ориентированных на большое число участников, как правило используется автоматическая проверка таких вопросов. В случае же каких-либо контрольных мероприятий в рамках, например, СДО ВУЗа, рекомендуется использовать проверку ответов преподавателем, чтобы избавиться от технических ошибок при проверке задания в автоматическом режиме.

Кроме того, стоит отметить, что данный тип тестирования, используемый, например, для проверки решения задач по математике, не позволяет предоставить пользователю информативную обратную связь о его ошибках.

Тестирование в форме эссе применяется как правило для контрольных мероприятий в рамках СДО ВУЗа и т.п. В этом случае задание проверяется преподавателем. В открытых онлайн-курсах такие задания как правило не применяются, так как преподавательского ресурса недостаточно для проверки заданий всех пользователей, число которых может быть очень велико. В редких случаях, когда использование такой формы тестирования все же необходимо (например, из-за гуманитарной тематики курса или при отсутствии технической возможности проверить задачу на программирование), прибегают к системе перекрестной проверки. В таком случае требования к ответу стараются максимально формализовать, чтобы пользователи могли более объективно оценить друг друга. При перекрестном тестировании в качестве итоговой оценки, как правило, выставляется среднее или медианное значение результатов нескольких проверок [7].

Пример интерфейса проверки задания с перекрестным оцениванием на платформе Stepik приведен на рисунке 3.

The screenshot displays the 'Рецензия' (Review) interface on the Stepik platform. On the left, the review details include the step name 'Шаг: PyTest — параметризация, конфигурирование, плагины Шаг 9', the solution ID 'Решение: #787527998', and the date '23 октября 2022 г., 22:04, верно'. Below this is a section for the review condition 'Условие:' with a 'Развернуть' (Expand) button. The solution ID 'Решение #787527998' is repeated, followed by a reminder to submit the solution for review and a text box containing the URL 'https://github.com/ekaterinatest/stepik_auto_tests_course2.git'. On the right, two criteria are listed for evaluation. 'Критерий 1' describes a test in a repository that can be run with 'pytest --language=es' and successfully passes. 'Критерий 2' describes a code functionality check for different languages, mentioning 'time.sleep(30)' and 'Ajouter au panier'. Each criterion has an 'Оценка' (Evaluation) section with radio buttons for '0' and '1', and a text box for 'Развернутое объяснение вашей оценки' (Expanded explanation of your evaluation). A note indicates that '0' is a mandatory field.

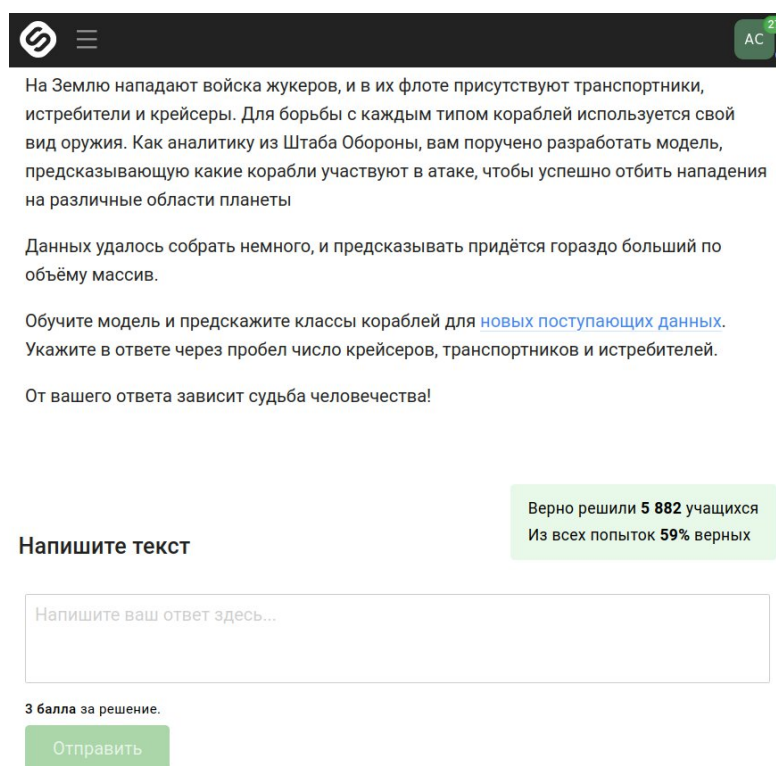
Рисунок 3 — интерфейс проверки задания с перекрестным оцениванием на платформе Stepik

2.3 Тестирование на написание программы

2.3.1 Проверка по референсным значениям

Зачастую, так как разработчики онлайн-портала не обладают достаточными ресурсами для создания подсистемы автоматического тестирования пользовательских программ, либо сама архитектура проверяемой программы не позволяет протестировать ее автоматически по техническим причинам (например, программа пользователя связана с автоматическим тестированием, программа связана с машинным обучением и потребляет много вычислительных ресурсов и т.д.)

Пример таких заданий приведены на **рисунках 4 и 5**.



На Землю нападают войска жуков, и в их флоте присутствуют транспортники, истребители и крейсеры. Для борьбы с каждым типом кораблей используется свой вид оружия. Как аналитику из Штаба Оборона, вам поручено разработать модель, предсказывающую какие корабли участвуют в атаке, чтобы успешно отбить нападения на различные области планеты

Данных удалось собрать немного, и предсказывать придётся гораздо больший по объёму массив.

Обучите модель и предскажите классы кораблей для [новых поступающих данных](#). Укажите в ответе через пробел число крейсеров, транспортников и истребителей.

От вашего ответа зависит судьба человечества!

Верно решили **5 882** учащихся
Из всех попыток **59%** верных

Напишите текст

Напишите ваш ответ здесь...

3 балла за решение.

Отправить

Рисунок 4 — проверка задания на машинное обучение по референсным значениям

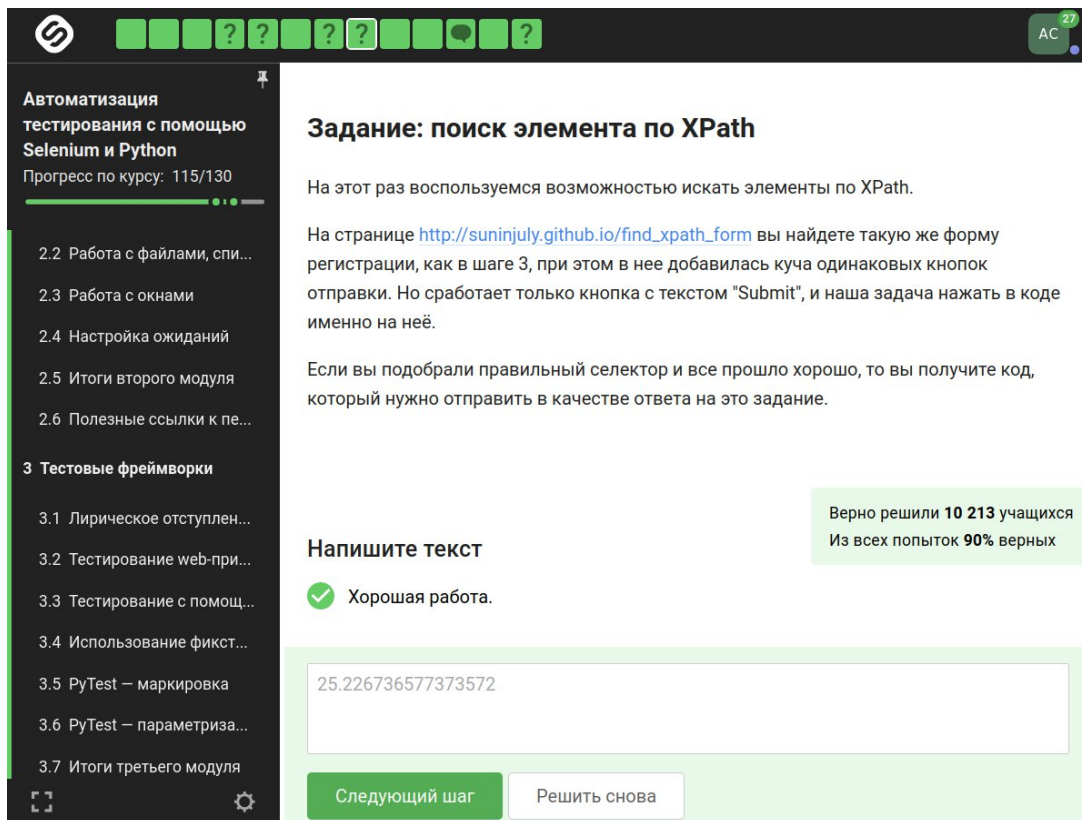


Рисунок 5 — проверка задания на автоматизированное тестирование по референсным значениям

Бизнес-процесс прохождения тестирования на написание программы с проверкой по референсным значениям приведен на **рисунке 6**.



Рисунок 6 — бизнес-процесс прохождения тестирования на написание программы с проверкой по референсным

При анализе представленного бизнес-процесса становятся очевидны ряд очевидных недостатков такого типа тестирования.

Первым недостатком является отсутствие информативной обратной связи, которое затрудняет пользователю поиск семантических ошибок в логике программы.

Вторым недостатком является необходимость установки дополнительного ПО (текстового редактора и компилятора, либо среды разработки) со стороны пользователя. Это не только повышает входной порог, но и лишает пользователя возможности проходить обучение и тестирование без своего компьютера (например, с мобильного устройства в общественном транспорте, во время командировки или путешествия).

Третьим недостатком является необходимость составления таких заданий и подбор таких входных данных, результаты которых достаточно сложно или невозможно рассчитать без написания требуемой программы. Зачастую, этот процесс может быть затруднительным и в итоге потребует от пользователя написания более сложной программы, чем в случае если бы задание было нацелено исключительно на формирование и проверку целевого навыка.

2.3.2 Автоматическое тестирование на проверяющей стороне + ?С4?

Наиболее **каноничным** способом проверки заданий по программированию является автоматическое тестирование на проверяющей стороне. Бизнес-процесс прохождения тестирования на написание программы с автоматической проверкой показан на **рисунке 7**.



Рисунок 7 — бизнес-процесс прохождения тестирования на написание программы с автоматической проверкой

Положительными сторонами такого подхода к проверке заданий на программирования являются [8]:

- объективность оценки (оцениваемые программы проходят через одинаковый набор тестов или эквивалентные между собой наборы тестов);
- скорость оценки;
- наглядность оценки (хотя процесс тестирования проходит в режиме черного ящика, результат тестирования, при должной подготовки САТ и самих тестов, нагляден и способен в некоторой степени оповестить об ошибках в тестируемых приложениях);
- возможность применения для большого числа пользователей.

? Что-то про сложность реализации и применимость не ко всем задачам?

? С4 ?

? 2.3.3 Про систему Paradigm ?

2.4 Сравнительный анализ методов

3 Система оценивания + Paradigm CTF + ?статистика?

4 Бизнес-процессы и архитектура приложения мечты

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. <https://cyberleninka.ru/article/n/rol-kontrolya-v-obuchenii-studentov-vuza>
2. https://www.researchgate.net/profile/Adrien-Presley/publication/2447898_The_Use_of_IDEF0_for_the_Design_and_Specification_of_Methodologies/links/554a57490cf29f836c964ddf/The-Use-of-IDEF0-for-the-Design-and-Specification-of-Methodologies.pdf
3. https://www.researchgate.net/profile/Adrien-Presley/publication/2447898_The_Use_of_IDEF0_for_the_Design_and_Specification_of_Methodologies/links/554a57490cf29f836c964ddf/The-Use-of-IDEF0-for-the-Design-and-Specification-of-Methodologies.pdf
4. <https://c4model.com/>
5. <http://www.ssi.magtu.ru/doc/mpos-2011-3.pdf#page=167>
6. <https://support.stepik.org/hc/ru/articles/360000159673-%D0%9F%D1%80%D0%B0%D0%BA%D1%82%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B5-%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D1%8F>
7. <https://support.stepik.org/hc/ru/articles/360000159713-%D0%A1%D0%BE%D1%81%D1%82%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5-%D0%B7%D0%B0%D0%B4%D0%B0%D0%BD%D0%B8%D0%B9-%D1%81-%D1%80%D0%B5%D1%86%D0%B5%D0%BD%D0%B7%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5%D0%BC>
8. <https://science-education.ru/ru/article/view?id=24719>
- 9.