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)

2022 г.

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 функциональных возможностей и бизнес-процессов систем тестирования знаний языков программирования, на основе результатов анализа сформировать функциональные требования и спроектировать архитектурную модель подсистемы тестирования знаний языков описания аппаратуры*_____

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 _____ Т.А. Ким
(Подпись, дата) (И.О. Фамилия)

Студент _____ С.В. Астахов
(Подпись, дата) (И.О. Фамилия)

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

РЕФЕРАТ

Отчет 24 с., 10 рис., 3 табл., 7 ист.

ТЕСТИРОВАНИЕ ЗНАНИЙ, ЯЗЫКИ ПРОГРАММИРОВАНИЯ, ОБРАЗОВАТЕЛЬНЫЕ ПОРТАЛЫ, ХАКАТОНЫ, СДО.

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

Цель работы — анализ функциональных возможностей существующих платформ обучения языкам программирования, получение описания их бизнес-процессов (с использованием нотации IDEF0), формирование на их основе функциональных требований и архитектурной модели (на основе нотации C4) подсистемы тестирования знаний языков описания аппаратуры.

В процессе работы был проведен анализ и функциональное моделирование бизнес-процессов имеющихся систем тестирования знаний языков программирования. В результате доработки полученных изначально моделей было получено описание бизнес-процессов, а затем и архитектурная модель проектируемой системы.

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 образовательных ресурсов, направленных на практическое освоение языков описания аппаратуры.

Все существующие на данный момент интернет-порталы, посвященные данной тематике предлагают лишь теоретические знания и задания, требующие установки стороннего программного обеспечения и предполагающие самопроверку. Ни в одном из существующих на данный момент порталов не представлена функция автоматизированной проверки кода.

СОДЕРЖАНИЕ

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

C4 — context container component code

IDEF0 — integrated computer aided manufacturing definition

VCD — value change dump

СДО — система дистанционного образования

ВЕДЕНИЕ

Основной целью научно-исследовательской работы является анализ функциональных возможностей существующих платформ обучения языкам программирования, получение описания их бизнес-процессов формирование на их основе функциональных требований и архитектурной модели 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 об активности пользователей.

Так как большинство существующих интернет-ресурсов, посвященных изучению языков описания аппаратуры, не обладают достаточным набором функциональных возможностей, в качестве ближайших аналогов были рассмотрены образовательные платформы, предназначенные для изучения «классических» языков программирования (например, Stepik, Coursera, Udemy). Несмотря на то, что языки описания аппаратуры имеют ряд отличий от обычных языков программирования, для тестирования знаний в этих областях могут применяться общие методики и решения.

1 Проблематика изучения языков описания аппаратуры и методология моделирования программных систем

1.1 Проблематика изучения языков описания аппаратуры

Несмотря на наличие большого числа теоретических материалов, посвященных языкам описания аппаратуры, наблюдается дефицит и низкое качество организации ресурсов, ориентированных на их практическое освоение. Абсолютное большинство таких ресурсов (marsohod.org, portal-ed.ru, asic-world.com) предоставляет лишь теоретические данные и набор практических упражнений, которые пользователю предлагается выполнить 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 языками описания аппаратуры среды: Quartus и Xilinx. Обе они требуют большого объем 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 программируемую логическую интегральную схему. 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 языками описания аппаратуры для людей, интересующихся ими, например, в качестве хобби или с целью продолжить обучение в университете 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 Нотация 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).

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).

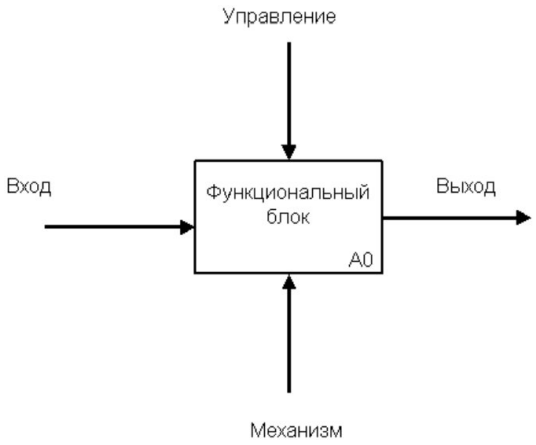

Рисунок 1 — функциональный блок в нотации IDEF0

Основным достоинством IDEF0 является высокая степень формализации, которая позволяет легко интегрировать фрагменты разрабатываемой модели в единую систему [2].

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

1.3 Модель визуализации программной архитектуры (нотация C4)

Модель C4 — 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 C4 выделяют 4 уровня диаграмм [3]:

- 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 программного обеспечения 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 тестирования знаний

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

В качестве основы была взята подобная классификация для системы дистанционного обучения (далее — СДО) Moodle [4]. Она была дополнена с учетом функциональных особенностей таких СДО, как Huawei University, Coursera, Stepik, Ethernaut, а также хакатона Paradigm CTF [5]. Сформированная классификация приведена в таблице 1.

Таблица 1 — классификация методов тестирования знаний

№	Тип	Подтип
1	Тестирование с ответом в закрытой форме	1.1 Выбор одного ответа 1.2 Выбор множественных ответов 1.3 Сопоставление
2	Тестирование с коротким ответом	2.1 С автоматизированной проверкой 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 исходного кода	4.1 С проверкой по референсным значениям 4.2 Автоматизированное 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.

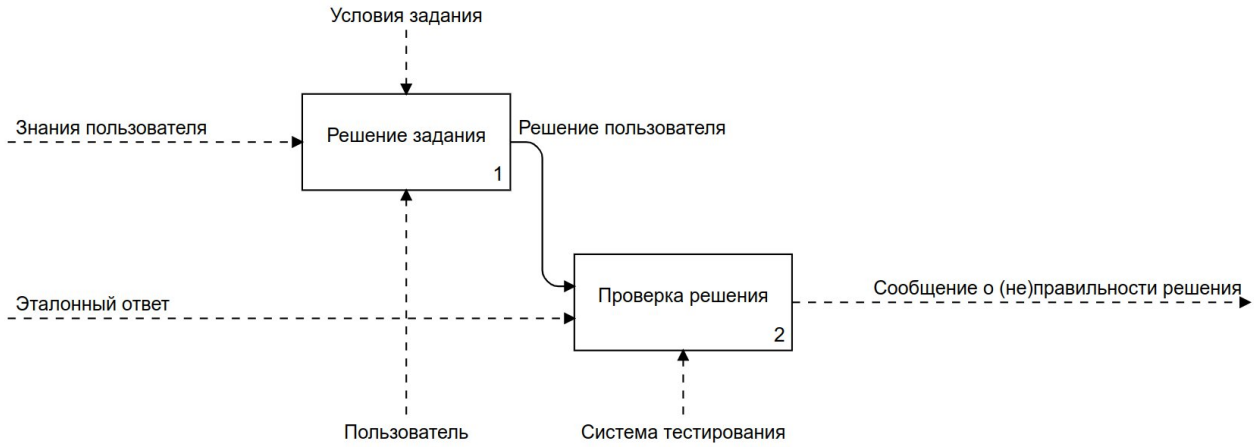

Рисунок 1 — бизнес-процесс тестирования с ответом в закрытой форме

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.

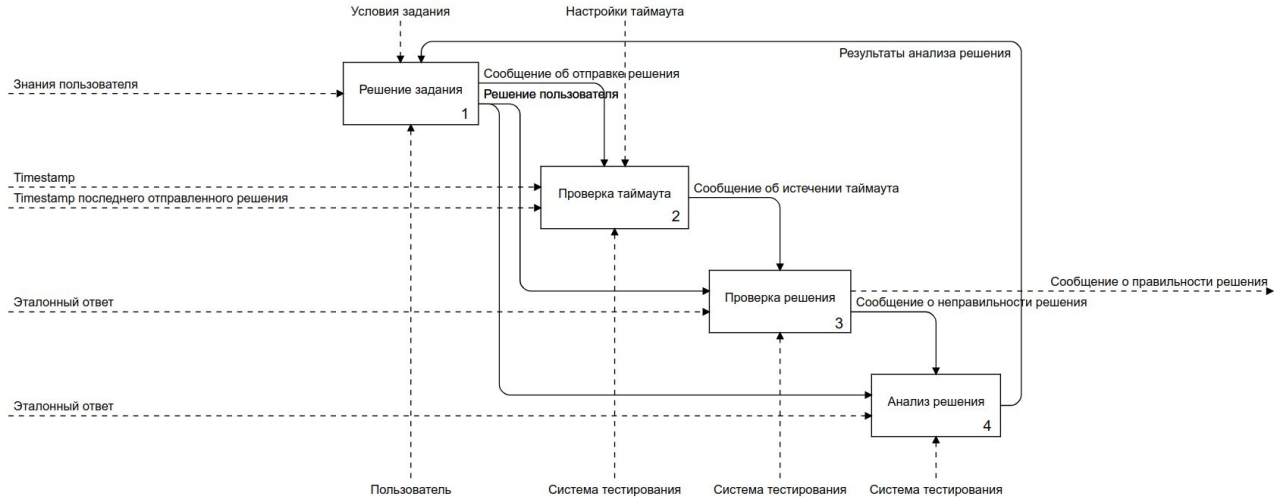

Рисунок 2 — усовершенствованный бизнес-процесс тестирования

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. Автоматизированная 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 автоматизированная проверка таких вопросов. В случае же каких-либо контрольных мероприятий в рамках, например, СДО университета, рекомендуется использовать проверку ответов преподавателем, чтобы избавиться от технических ошибок при проверке задания в автоматизированном 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 университета 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 [6].

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

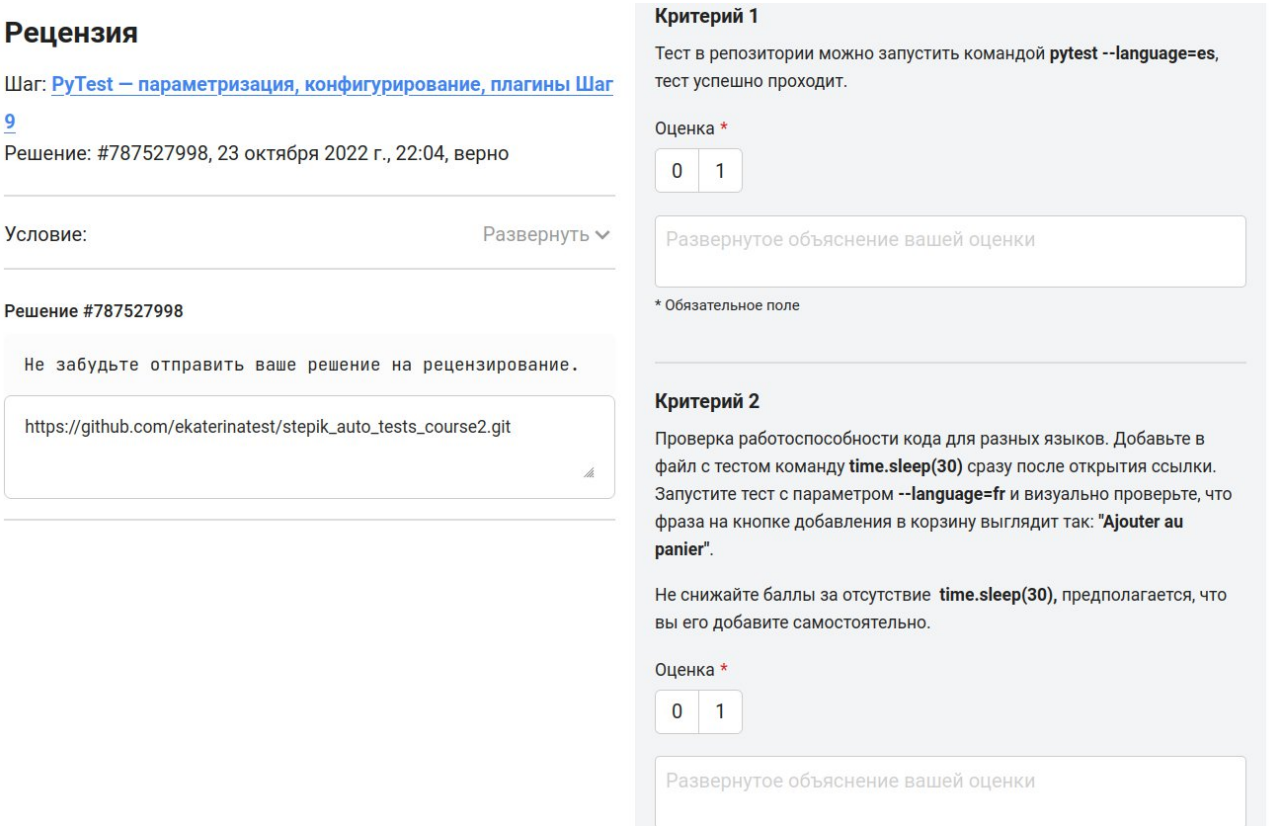

Рисунок 3 — интерфейс проверки задания с перекрестным оцениванием на платформе Stepik

2.3 Тестирование на написание исходного кода

2.3.1 Проверка по референсным значениям

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

Пример таких заданий приведены на рисунках 4 и 5.

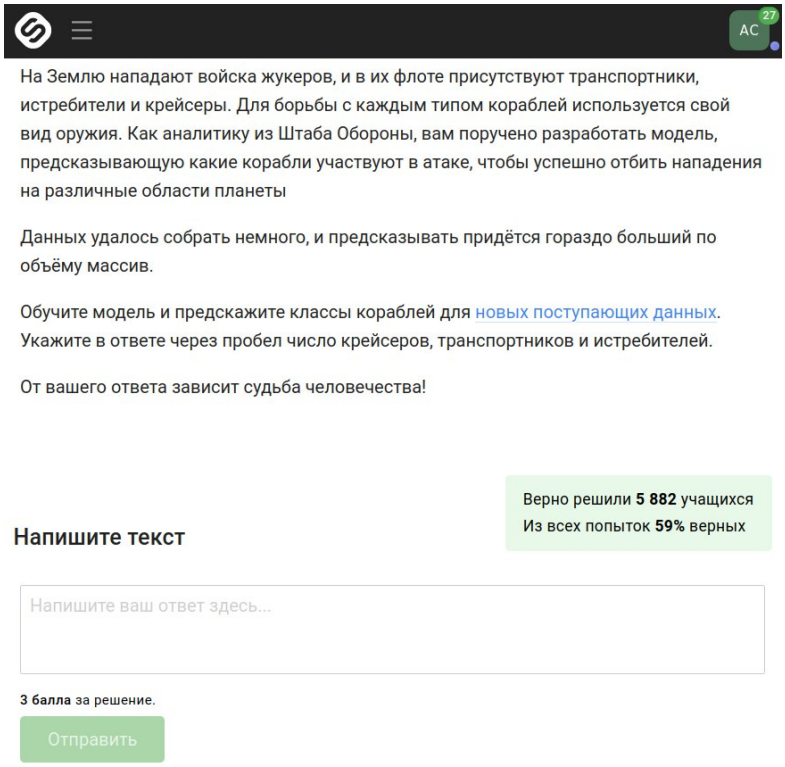

Рисунок 4 — проверка задания на машинное обучение по референсным значениям

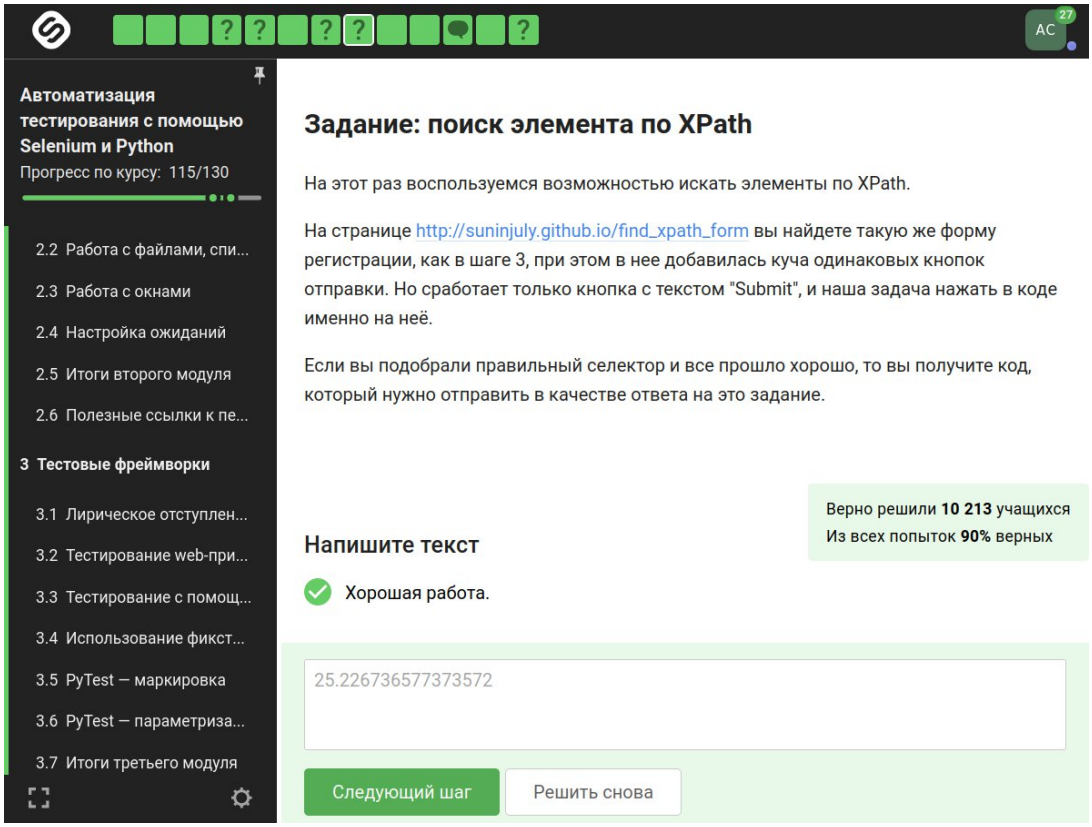

Рисунок 5 — проверка задания на автоматизированное тестирование по референсным значениям

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

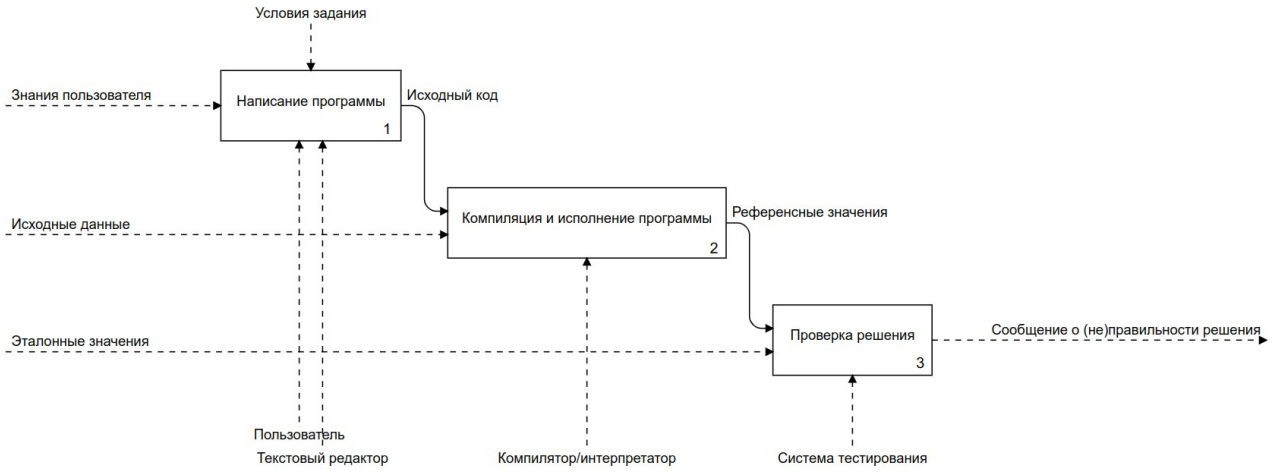

Рисунок 6 — бизнес-процесс прохождения тестирования на написание программы с проверкой по референсным

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 программного обеспечения (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 компьютеры (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 Автоматизированное тестирование на проверяющей стороне

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

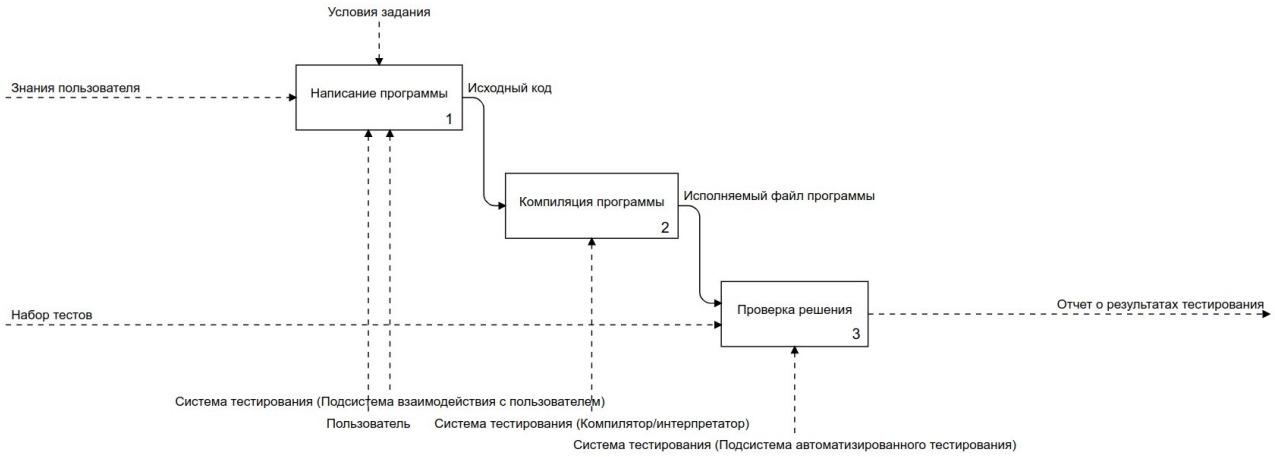

Рисунок 7 — бизнес-процесс прохождения тестирования на написание программы с автоматизированной проверкой

Положительными сторонами такого подхода к проверке заданий на программирования являются [7]:

- 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 системы автоматизированного и самих тестов, нагляден и способен в некоторой степени оповестить об ошибках в тестируемых приложениях);
- возможность применения для большого числа пользователей.

Основным недостатком такого подхода является сложность его реализации.

3 Системы статического и динамического оценивания

Как правило, в СДО, ориентированных на открытые онлайн-курсы, количество баллов, полученных за решение задания зависит лишь от сложности задания и степени корректности решения, не зависит от количества неправильных попыток. Это необходимо, чтобы обучающийся при должном упорстве всегда мог закончить курс, получив достаточное количество баллов. Как правило, прогресс по курсу определяется именно числом полученных баллов (рисунок 8), так как нельзя оценивать прогресс лишь по количеству решенных заданий, необходимо учитывать их различающуюся сложность (что выражается в количестве получаемых за задание баллов).

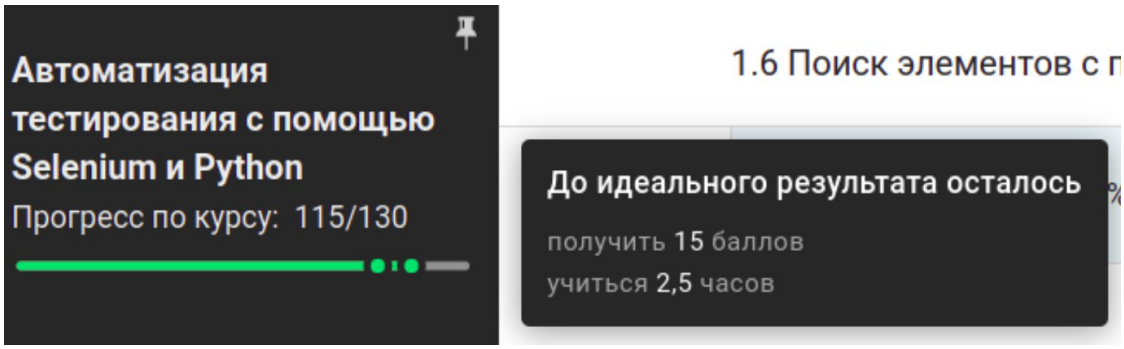

Рисунок 8 — оценка прогресса по курсу на основе полученных баллов на платформе Stepik

Кроме того, зачастую за такие задания часто выставляется либо нулевой, либо максимальный балл (в том числе за задания с множественным выбором ответов). Это, с одной, стороны заставляет пользователя перерешивать задание, пока оно не будет решено идеально. С другой стороны, в случае сложного вопроса, это может заставить пользователя прибегнуть к «механическому» перебору ответов.

Альтернативой такому подходу является выставление динамической оценки, учитывающей, например, количество неудачных попыток. Такой вид оценивания, как правило, применяется в олимпиадах по программированию и соревнованиях по «захвату флага» [8].

В качестве примера подобной системы оценивания может быть рассмотрена система оценивания хакатона Paradigm CTF (рисунок 9).

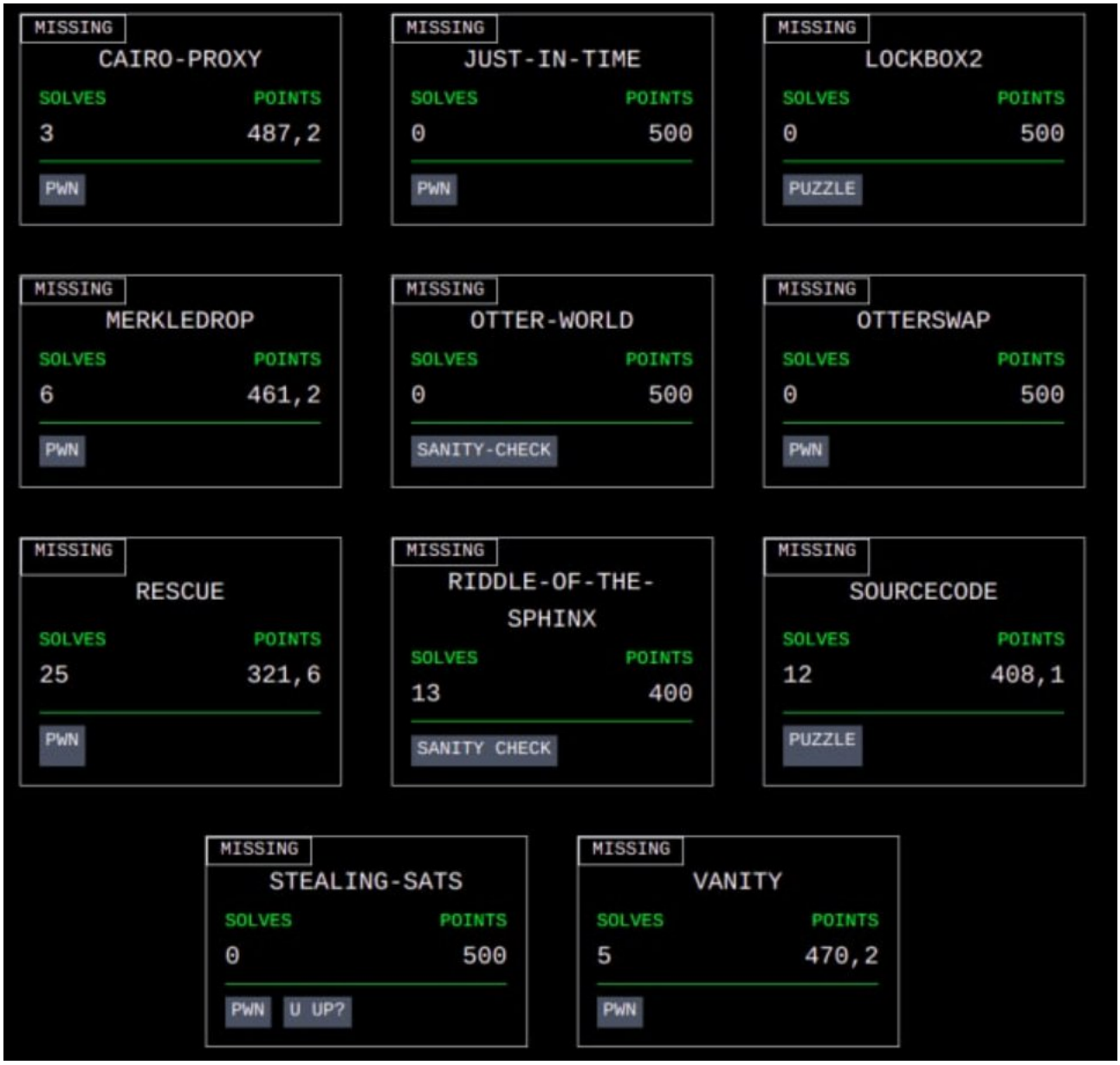

Рисунок 9 — оценка задания на Paradigm CTF

Подобную систему оценивания сложно применить для оценки прогресса по курсу, однако ее можно ввести параллельно для формирования рейтинга пользователей с целью повышения их мотивации.

Для этого необходимо регистрировать не только успешное прохождение заданий, но и статистику ошибок.

4 Функциональные требования и архитектура проектируемой подсистемы

На основе проанализированных методов тестирования и оценивания знаний знаний было решено использовать в проектируемой системе методы из таблицы 3, метод статического оценивания для оценки прогресса по курсу и метод динамической оценки для составления рейтинга пользователей.

Таблица 3 — методы тестирования знаний в проектируемой системе

№	Тип	Подтип	Особенности
1	Тестирование с ответом в закрытой форме	1.1 Выбор одного ответа 1.2 Выбор множественных ответов 1.3 Сопоставление	Имеется проверка таймаута и обратная связь в виде результатов анализа ответа (рисунок 2).
2	Тестирование с коротким ответом	2.1 С автоматизированной проверкой	Система тестирования должна иметь словарь с различными формами и синонимами искомого слова (в случае, если подразумевается словесный, а не численный ответ).
3	Тестирование на написание исходного кода	3.1 Автоматизированное тестирование на проверяющей стороне	Система тестирования должна учитывать специфику языков описания аппаратуры и уметь анализировать и строить временные диаграммы на основе VCD-файлов (рисунок 7).

На основе результатов проведенного анализа было заключено, что проектируемая подсистема должна выполнять следующие функции:

- изменение заданий модератором;
- отображение персональной статистики учащегося;

– обработка статистики решения заданий;
– формирование рейтингового списка учащихся;
– автоматизированная проверка тестов с закрытым ответом, кратким ответом и ответом в виде исходного кода;
– формирование информативной обратной связи в случае неверного решения задания учащимся;
– формирование временных диаграмм работы устройств.

На основе результатов проведенного анализа и сформулированных функциональных требований была разработана обобщенная архитектурная модель проектируемой подсистемы, представленная на рисунке 10.

Рисунок 10 — архитектурная модель проектируемой системы

Архитектурная модель подразумевает разбиение проектируемой подсистемы на следующие компоненты, назначение которых раскрывается в техническом задании (приложение А):

– подсистема формирования заданий;
– подсистема анализа статистики;
– анализатор временных диаграмм;
– синтезатор (по функциям аналогичен компилятору);
– базы данных заданий статистики.

Работа с учетными данными пользователей, отрисовка пользовательского интерфейса и другие стандартные функции реализуются вне рамок проектируемой подсистемы.

ЗАКЛЮЧЕНИЕ

В ходе работы проведен анализ функциональных возможностей существующих платформ обучения языкам программирования, получены и доработаны модели их бизнес-процессов, на их основе сформулированы функциональные требования и архитектурная модель подсистемы тестирования знаний языков описания аппаратуры.

Основными отличиями проектируемой подсистемы от аналогов являются:

- реализация модуля автоматизированного тестирования на основе временных диаграмм, описываемых VCD-файлами;
- учет статистики прохождения заданий;
- информативная обратная связь об ошибках учащегося;
- формирование рейтинга учащихся.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Мовчан И. Н. Роль контроля в обучении студентов вуза // Психология и педагогика: методика и проблемы практического применения. 2008. №1. URL: https://cyberleninka.ru/article/n/rol-kontrolya-v-obuchenii-studentov-vuza (дата обращения: 04.10.2022).
2. Adrien P. The Use of IDEF0 for the Design and Specification of Methodologies / P. Adrien // Researchgate : электронный журнал. – URL: https://www.researchgate.net/publication/2447898_The_Use_of_IDEF0_for_the_Design_and_Specification_of_Methodologies. – Дата публикации: 01.10.1998.
3. C4model [Электронный ресурс]. – URL: https://c4model.com/ (дата обращения: 15.10.2022)
4. Ильина Е.А. Технология тестирования знаний студентов с использованием системы Moodle / Е.А. Ильина, Л.Г. Егорова, А.В. Дъяконов // Математическое и программное обеспечение систем в промышленной и социальной сферах . – Магнитогорск, 2011. – С. 166-172.
5. Справочный центр Stepik - Практические задания [Электронный ресурс]. – URL: https://clck.ru/Nu5Wy (дата обращения: 20.10.2022)
6. Справочный центр Stepik - Составление заданий с рецензированием [Электронный ресурс]. – URL: https://clck.ru/32a9FC (дата обращения: 01.11.2022)
7. Гладких И.Ю., Якушин А.В. Системы автоматизированного тестирования по программированию в образовательном пространстве // Современные проблемы науки и образования. – 2016. – № 3. ; URL: https://science-education.ru/ru/article/view?id=24719 (дата обращения: 05.11.2022).
8. Xaker.ru — Capture the Flag [Электронный реусрс]. — URL: https://xakep.ru/2016/06/14/ctf/ (дата обращения: 10.11.2022).

ПРИЛОЖЕНИЕ А

Техническое задание

Листов X