

Санкт-Петербургский государственный университет

Математическое обеспечение и администрирование информационных систем

Технология программирования

Шайтан Василий Ильич

Развитие системы тестирования для курса
”Функциональное программирование”

Бакалаврская работа

Научный руководитель:
ст. преп. Симуни М. Л.

Рецензент:
старший программист ООО ”Люксофт Профешнл” Соловьев А. Ю.

Санкт-Петербург
2016

SAINT-PETERSBURG STATE UNIVERSITY

Software and administration of information systems
Programming technology

Vasily Shaytan

Developing a testing system for Functional Programming course

Graduation Thesis

Scientific supervisor:
Senior Lecturer Mikhail Simuni

Reviewer:
Luxsoft Professional Senior Developer Aleksei Solovev

Saint-Petersburg
2016

Оглавление

Введение	5
1. Постановка цели и задач	6
1.1. Причины возникновения проблемы и ее актуальность . . .	6
2. Обзор существующих решений	8
2.1. Blackboard Learn	9
2.2. Moodle	10
2.3. Stepic	12
2.4. Sphere Online Judge	15
3. Проектирование системы	16
3.1. Выбор средств разработки	16
3.2. Выбор средств для компиляции и тестирования программ	16
4. Проектирование возможностей системы	17
4.1. Автоматическое засчитывание правильных решений . . .	17
4.2. Автоматическое проставление замечаний	18
5. Описание интерфейса системы	20
5.1. Список правильных решений	20
5.2. Добавление правильных решений	21
5.3. Другие возможности	21
5.4. Автоматическое засчитывание	23
5.5. Список типичных замечаний	23
5.6. Добавление нового типичного замечания	23
5.7. Другие возможности	24
5.8. Проверка задач с использованием типичных замечаний .	26
6. Описание реализации	28
6.1. Схема баз данных	28
6.2. Обновление устаревших функций	29
6.3. Переименования переменных	29

6.4. Структура проекта	30
7. Тестирование системы	32
Заключение	33
Список литературы	34

Введение

В настоящее время активно развиваются системы управления обучением (Learning management system [13]). Онлайн обучение позволяет каждому легко, доступно, в индивидуальном темпе получить новые знания. Так и для студентов 4 курса математико-механического факультета есть возможность получить такие знания. Для них читается курс "Функциональное программирование", в котором есть система тестирования [15]. Эта система позволяет отправлять решения и проверяет их по тестам, после чего результат тестирования получает преподаватель. Дальше преподаватель решает, что делать с решением - зачесть, написать замечание или, может быть, отклонить как неверное или списанное. Все это занимает достаточно много времени и возникла необходимость как-то упростить этот процесс.

1. Постановка цели и задач

Целью данной работы является упрощение и частичная автоматизация процесса проверки задач преподавателем в системе тестирования.

Для достижения этой цели был сформулирован следующий набор задач:

- изучить существующую систему тестирования
- изучить существующие системы управления обучением и их возможности с точки зрения тестирования и проверки преподавателем
- реализовать функциональность для работы с правильными решениями
- реализовать функциональность для работы с замечаниями
- проанализировать результаты тестирования системы

1.1. Причины возникновения проблемы и ее актуальность

В настоящее время систем управлением обучения достаточно много, но большинство из них засчитывает задачи по тестам. У нашей системы есть следующие особенности:

- тестирование проходит автоматически, а потом преподаватель просматривает решение
- замечания по стилю кода добавляются вручную

Для многих задач накладываются дополнительные требования использовать рекурсию. Мы хотели бы сохранить ручную проверку и при этом, заметно снизить время, которое уходит, например, на проверку задач.

Были придуманы 2 способа, которые позволяют сократить время работы преподавателя:

- Есть довольно большое количество задач, у которых есть ограниченное число решений. Например, одна из типичных задач курса - это "Найти для данного списка его минимальной элемент". Студенты присылают разные решения, но это разнообразие ограничивается примерно пятью-шестью различными решениями.

Таким образом, первый способ - завести таблицу для таких правильных решений. Решения, совпадающие с правильными, засчитываются автоматически. Отметим, что такой подход применим к простым задачам. Сложные задачи по-прежнему засчитываются вручную.

- Еще одной особенностью нашей системы является то, что преподаватель не только проверяет решения, но и пишет к ним замечания. Эти замечания довольно часто повторяются.

Поэтому, второй способ сокращения времени работы преподавателя - это завести таблицу типичных замечаний. Такие замечания преподаватель может добавлять вручную или автоматически.

Приведем простой пример автоматического добавления замечаний. В языке Haskell для выражений `if (x*x)` скобки можно опускать. Чтобы не писать постоянно замечания, мы добавляем в таблицу замечаний образец, который позволяет идентифицировать это замечание. Например, в данном случае таким образом может быть `"if ("`. Это и делает автоматическое добавление замечаний.

2. Обзор существующих решений

В современном мире онлайн-образование постепенно становится все более популярным. Возможность учиться у профессоров ведущих учебных заведений, изучать новые области, получать нужные в работе знания, не выходя из дома, привлекает большое количество людей.

Онлайн-обучение обычно проходит в форме массовых открытых онлайн-курсов (МООС, massive open online course [10]). Обычно это подразумевает видео, слайды и текстовый контент, подготовленные преподавателем, а также задачи для проверки знаний, которые обычно проверяются автоматически, но также возможна проверка студентами работ своих товарищей. В качестве задач могут быть предложены самые разнообразные типы заданий: от простого выбора правильного ответа до задач на программирование и написания эссе.

У онлайн-образования есть свои особенности, отличающие его от обычного, оффлайн-образования. Среди плюсов, во-первых, уже упомянутая выше доступность каждому, у кого есть доступ к интернету. Во-вторых, это почти неограниченная масштабируемость: благодаря автоматизированной проверке задач на курсе могут одновременно учиться тысячи человек, что несопоставимо с обычными курсами в учебных аудиториях. В-третьих, каждый студент может выбирать удобное для себя время и темп для прохождения материала.

В то же время в онлайн-обучении есть и минусы. В отличие от традиционного образования, где у студента всегда есть мотивация в виде оценки его академической успеваемости, в случае онлайн-курсов нет никаких штрафов за не пройденный курс. Из-за этого доля закончивших курс из записавшихся на него редко превышает 10%. Помимо этого, из-за большого числа учащихся у преподавателя нет никакой возможности уделять индивидуальное внимание каждому студенту согласно его уровню и возможностям [12].

Приведем небольшой обзор наиболее популярных систем управления обучением (Learning management system).

2.1. Blackboard Learn

Компания Blackboard Inc. разработала платформу электронного образования Blackboard Learn [1], чтобы обеспечить пользователям всего мира доступ к образовательным нововведениям благодаря соединению людей и технологии. Основные возможности:

- многофункциональный контент-редактор;
- онлайн-оценивание;
- ориентированный на пользователя интерфейс;
- исследования учреждений и оценка курсов;
- активное сотрудничество;

Возможности платформы для студента

Платформа предоставляет студентам доступ к онлайн курсам, ведет учет курсов, которые в данный момент проходит пользователь. Система позволяет участникам общаться с преподавателем и студентами, проходящими тот же курс.

Blackboard Learn дает возможность своевременно получать уведомления о всех обновлениях и новостях и помогает контролировать эффективность обучения за счет системы оценивания работ студента.

Возможности платформы для преподавателя

Преподавателю платформа предоставляет возможность разместить свой онлайн-курс и поддерживать его в актуальном состоянии при помощи редактора контента. Преподаватель может создавать свои системы оценивания знаний, такие как тесты, опросы, задания и прочее.

Кроме того, преподаватель имеет возможность получить оценку своего курса и провести исследование некоторых ключевых показателей качества курса, таких как: полнота и простота изложения, качество представления информации, актуальность курса. Для этого в платформе

разработан механизм создания опросов пользователей, сбора и графического отображения значений различных показателей (частота посещения страниц курса, успеваемость студента, время на выполнение задание и другое). Пример страницы из системы Blackboard приведен на рис. 1.

Соответствие платформы поставленной задаче

Возможность тестирования кодов компьютерных программ не предусмотрена, поэтому данная платформа нам не подходит.

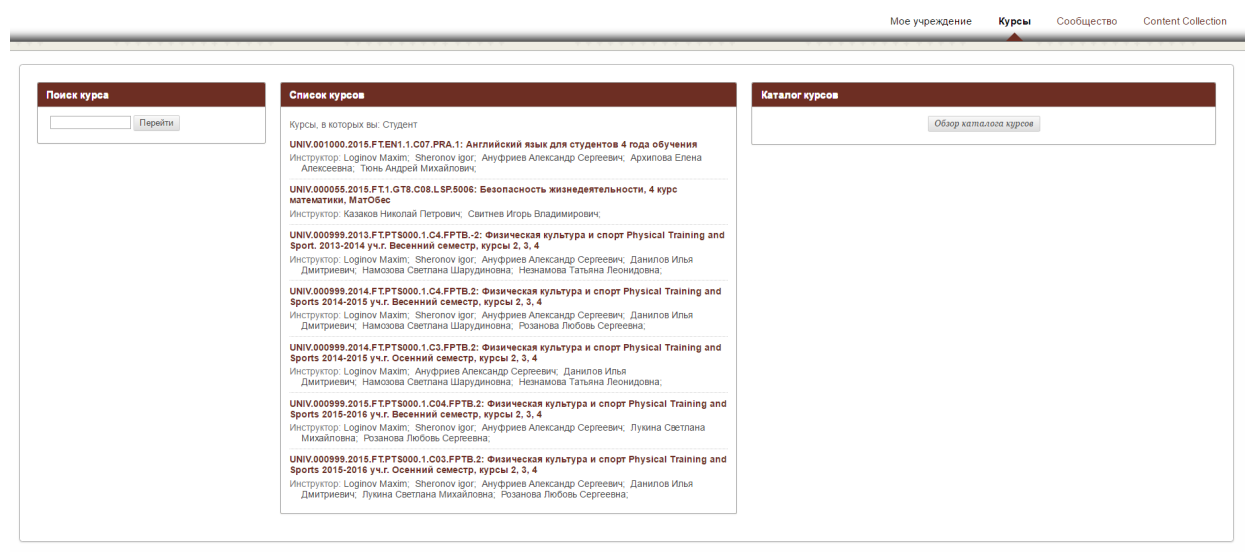


Рис. 1: Платформа Blackboard

2.2. Moodle

Moodle [6] предназначена для создания и проведения дистанционных курсов.

Moodle распространяется в открытых исходных кодах, что дает возможность адаптировать ее под особенности каждого образовательного проекта, интегрировать с другими информационными системами, дополнить новыми сервисами, вспомогательными функциями или отчетами, установить готовые или разработать совершенно новые дополнительные модули (активности).

Основные возможности Moodle [5]:

- простая интеграция с платформами, поддерживающими язык PHP;
- используется только одна база данных;
- для каждого курса имеется подробное описание;
- курсы могут быть разбиты по категориям;
- возможность редактирования страниц с помощью встроенного редактора.

Возможности платформы для преподавателя

- преподаватель имеет полное администрирование курса, но его права могут быть ограничены администратором;
- имеется большой выбор элементов: тесты, задачи, опросы, форумы и т.д.;
- сохраняются все последние изменения в курсе
- для каждого курса ведется мониторинг успеваемости слушателей курса;
- возможность коммуникации преподавателей и слушателей курса, так как система интегрирована с почтовыми системами;

Пример страницы из системы Moodle приведен ниже (рис. 2).

Соответствие платформы поставленной задаче

В данной системе не предусмотрена возможность тестирования кодов компьютерных программ. Устранить это ограничение можно, установив дополнительное расширение или плагин. Однако, на наш взгляд такое решение является весьма трудоемким. Кроме того, система Moodle сама по себе достаточно громоздка и имеет множество избыточных функций. Таким образом, возможно применение системы для нашей задачи, однако, оно потребовало бы значительных трудозатрат.



Рис. 2: Moodle

2.3. Stepic

Платформа stepic.org [9] — русскоязычный сайт, на котором размещено несколько десятков онлайн-курсов, преимущественно технической тематики. Каждый из них прошли от нескольких сотен до нескольких тысяч человек. Ежедневно на платформу заходят тысячи учащихся [12].

Терминология

Курсы на stepic.org разделены на модули, каждый модуль обычно рассчитан на неделю или две. У модуля могут быть установлены дедлайны на решение задач. После жесткого дедлайна пользователь все еще может проходить модуль, но уже не получит баллов за решение задач. Еще у модуля может быть мягкий дедлайн, после которого пользователь получит за решение задач часть баллов.

Модуль состоит из уроков, каждый из которых посвящен отдельному вопросу. Обычно уроки формируются так, чтобы можно было рассматривать урок в отрыве от курса, то есть как отдельную структурную единицу.

В каждом уроке есть несколько шагов (стэпов), представляющих собой либо лекционный материал в виде видеозаписи или текста, либо

задачу одного из множества видов. Бывают задачи-тесты с выбором ответа, задачи с текстовым ответом, задачи на программирование и многие другие. Большинство из них предполагают моментальную автоматическую проверку системой, но есть также задачи на написание эссе, которые студенты сами проверяют друг у друга на основе сформулированных преподавателем критериев.

Каждый урок или курс может быть помечен тегом — концепцией, которая описывает его содержимое. Примеры таких тегов: программирование, linux, статистика. Каждый тег привязан к объекту в базе знаний Wikidata.

Уроки могут быть объединены в пути. В отличие от курсов, у путей нет модулей, дедлайнов и возможности получить сертификат за их прохождение. Это просто способ объединять уроки в логическую последовательность. Обычно после того, как курс закрывается для просмотра, его уроки собираются в путь и выкладываются в библиотеку.

Основные возможности

В основном студенты приходят на stepic.org, чтобы изучать курсы. Успешно прошедшие курс и набравшие нужное количество баллов получают электронные сертификаты. Преподаватели могут установить границу для получения обычного сертификата, и для получения сертификата с отличием.

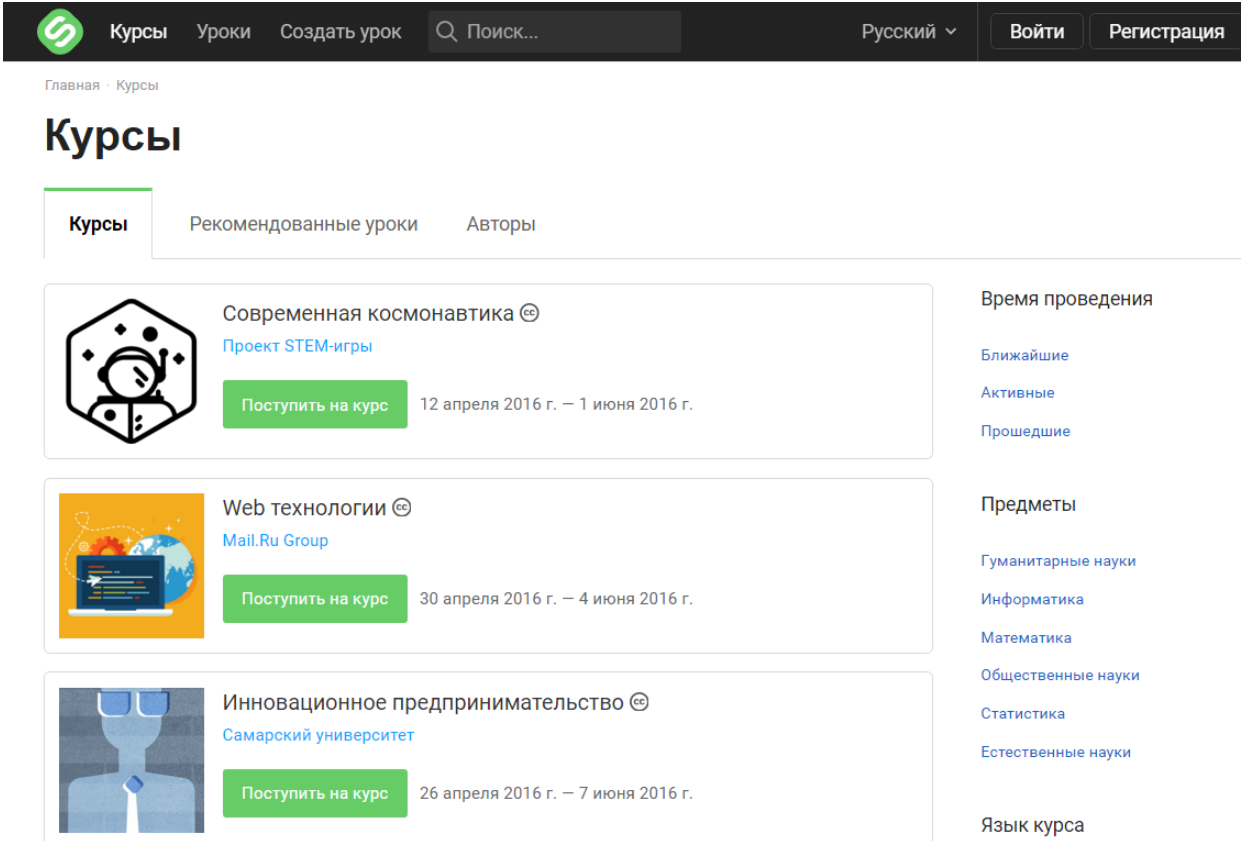
Помимо курсов на stepic.org также есть раздел библиотека, в котором уроки расположены без связи друг с другом. Уроки, которые были включены в один курс, здесь располагаются без связи друг с другом. Сейчас она представляет собой просто список уроков, из которых можно отфильтровать уроки по тегу, либо рассмотреть уроки из одного пути.

[Stepic.org](https://stepic.org) позволяет пользователям создавать свои уроки и курсы, давать ссылки на них или же встраивать в другие сайты. Также имеется возможность интеграции с любой платформой через LTI, благодаря чему в некоторых курсах на платформе Coursera [2] используются задачи, которые создаются и проверяются на stepic.org. Весь кон-

тент на платформе распространяется под открытой лицензией Creative Commons [3]. Пользователи могут общаться между собой с помощью механизмов комментариев под каждым стэпом. Там обычно обсуждают лекционный материал, помогают в решении задач или общаются с преподавателями [12]. Пример страницы в системе Stepic приведен на рис. 3.

Соответствие платформы поставленной задаче

Мы бы могли перенести свой курс на Stepic, там как раз есть тестирование кода на языке Haskell, но тогда все задачи засчитывались бы только по тестам. Да еще и нет возможности комментировать код студента, поэтому данная система нам не подойдет.



The screenshot shows the Stepic.org website interface. At the top, there is a navigation bar with a logo, links for 'Курсы', 'Уроки', 'Создать урок', and a search bar. On the right, there are options for 'Русский' and buttons for 'Войти' and 'Регистрация'. Below the navigation bar, the main heading is 'Курсы'. There are three tabs: 'Курсы', 'Рекомендованные уроки', and 'Авторы'. The 'Курсы' tab is active. The main content area displays three course cards. Each card includes a course icon, the course title, the provider, a 'Поступить на курс' button, and the course duration. To the right of the course cards, there are two vertical lists: 'Время проведения' (with options: Ближайшие, Активные, Прошедшие) and 'Предметы' (with options: Гуманитарные науки, Информатика, Математика, Общественные науки, Статистика, Естественные науки). At the bottom right, there is a 'Язык курса' section.

Course Title	Provider	Start Date	End Date
Современная космонавтика	Проект STEM-игры	12 апреля 2016 г.	1 июня 2016 г.
Web технологии	Mail.Ru Group	30 апреля 2016 г.	4 июня 2016 г.
Инновационное предпринимательство	Самарский университет	26 апреля 2016 г.	7 июня 2016 г.

Рис. 3: Stepic.org

2.4. Sphere Online Judge

sroj.com [8] - англоязычный сайт, на котором решаются олимпиадные задачи по программированию. Называются они - проблемами. В каждой проблеме внизу указывается ограничения, которые превышать нельзя и языки на которых эту проблему можно решать, а так же имя автора проблемы и дату добавления. После того, как вы справляетесь с проблемой на своем компьютере, ее стоит отправить для проверки на сайт. Дальше вы получите ответ, справились ли вы с задачей, или нет [11].

В системе присутствует рейтинг пользователей, т.е. чем больше решить задач, тем выше в рейтинге будет пользователь. Пример страницы из системы sroj можно видеть на рис. 4

Соответствие платформы поставленной задаче

В принципе sroj позволяет заводить пользовательские наборы задач. И мы могли бы завести наш набор задач там. Но это было бы не совсем удобно, так как нет возможности комментировать присланное решение. А ведь это очень важно.

list of classical problems

ID	NAME	QUALITY	USERS	ACC %	DIFFICULTY	C I
1	Life, the Universe, and Everything	👍 101	91303	33.12		
2	Prime Generator	👍 150	28427	15.52		
3	Substring Check (Bug Funny)	👍 1	674	11.00		
4	Transform the Expression	👍 43	19029	49.11		
5	The Next Palindrome	👍 117	9475	9.93		
6	Simple Arithmetics	👍 2	1753	12.32		
7	The Bulk!	👍 3	306	27.92		
8	Complete the Sequence!	👍 7	1749	32.72		
9	Direct Visibility		186	18.17		
10	Complicated Expressions	👍 1	632	26.57		
11	Factorial	👍 46	27568	45.56		
12	The Game of Master-Mind		201	44.33		
13	Hotline		146	29.41		
14	I-Keyboard	👍 3	677	37.19		
15	The Shortest Path	👍 24	2415	21.15		

Рис. 4: Список проблем на spoj.com

3. Проектирование системы

3.1. Выбор средств разработки

В качестве средств разработки серверной части системы был выбран язык PHP [7], так как на нем уже была написана существующая версия системы.

3.2. Выбор средств для компиляции и тестирования программ

Для реализации возможности компиляции и тестирования присланных решение на сервере был установлен Glasgow Haskell Compiler [4] (GHC) версии 7.10.2 - один из самых мощных и развитых на сегодняшний день компиляторов функционального языка Haskell.

4. Проектирование возможностей системы

Рассмотрим основные возможности, которые было решено обеспечить в системе. У преподавателя для каждого урока есть определенный набор задач. После каждой лекции в университете, на сайте системы становятся доступны задачи по новой теме. Задачи бывают 2 видов:

- базовые, они же основные, срок на их выполнение 1 неделя;
- дополнительные, срок на их выполнение уже 2 недели, так как они посложнее.

Было придумано два способа упрощения проверки задач:

- автоматическое засчитывание правильных решений
- автоматическое проставление замечаний

4.1. Автоматическое засчитывание правильных решений

Перечислим основные особенности подсистемы засчитывания решений.

1. Система хранит решения в таблице правильных решений
2. Система сравнивает решения с учетом преобразований
3. Было решено реализовать следующие преобразования:
 - Пробелы
 - Скобки (Система сравнивает решение студента с правильным решением с точностью до добавления и удаления скобок)
 - Переименования переменных (Система сравнивает решение студента с правильным решением с точностью до имен переменных. Но об этом подробнее будет ниже.)

Замечание: при использовании таких преобразований, могла бы появиться возможность автоматического засчитывания синтаксически неправильных задач, однако мы можем так делать, потому что задачи автоматически засчитываются только после того, как они прошли тесты.

4. Отсроченное засчитывание

Мы могли бы сразу засчитывать правильные решения, но чтобы студент не стал искать решение подбором, мы этого не стали делать. Вместо этого мы добавили кнопку в интерфейс преподавателя, при нажатии на которую, система засчитывает решения, которые помечены как "Одобрённые".

Опишем процесс автоматического засчитывания.

Для каждой задачи, если таковую решил студент, он отправляет решение в систему тестирования. Задача проходит по ряду тестов, и если все тесты прошли успешно, студент получает об этом информацию. Дальше ему выводится сообщение "Ожидайте проверки преподавателя". Но на этом действия системы с кодом студента не прекращаются. Просто об дальнейшем поведении системы студент не должен догадываться.

После того, как система отправила информацию о прохождении тестов, код студента подвергается обработке. Эта обработка включает в себя переименование переменных, убиение ненужных скобок и пробелов. После этого, мы для этой задачи ищем корректное решение в таблице правильных решений, которое после применения этой обработки будет совпадать с решением студента. А если, вдруг, нашлось такое совпадение, система помечает это решение, как одобрённое. И преподаватель с помощью специальной кнопки на своей странице может засчитывать все такие решения, одобрённые системой.

4.2. Автоматическое проставление замечаний

Перечислим основные особенности подсистемы проставления замечаний.

1. Храним образцы и замечания к ним в таблице
2. Образцы сравниваются с учетом пробелов и скобок, но без учета переименования переменных.
3. Замечания можно вставлять как автоматически, так и вручную
4. Автоматические замечания вставляются под той строкой, в которой был найден образец
5. Ручные замечания вставляются в том месте, где находится курсор, при этом текст замечания автоматически переходит на новую строку.

5. Описание интерфейса системы

В этой части мы не будем описывать все возможности системы тестирования. Опишем только те возможности, которые были добавлены в процессе работы над дипломом.

5.1. Список правильных решений

На главной странице преподавателя, в верхнем левом углу есть ссылка "Правильные решения". При нажатии на нее мы попадаем на страницу, на которой выводится список всех правильных решений (рис. 5).

Последние два столбца отвечают за редактирование и удаление со-

Задачи в списке отображаются в виде:
{номер домашнего задания} - {номер задачи в этом задании}

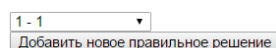


Таблица с правильными решениями

[На главную](#)

Для всех задач ▾ фильтровать

Номер задания	Номер задачи	Правильный код	Редактирование	Удаление
1	1	$f_0 = 2$ $f_n = (f_{n-1})^2$		
1	1	$f_0 = 2$ $f_1 = 4$ $f_n = (f_{n-1})^2$		
1	2	$f_1 = 2$ $f_n = 1 + 1 / f_{n-1}$		
1	4	$sum_{k=0}^n fact_k = sum_{k=0}^{n-1} fact_k + 1$ $sum_{k=0}^n fact_k = s$ $sum_{k=0}^n fact_k = sum_{k=0}^{n-1} fact_k + (s + c * (n - k + 1)) * (c * (n - k + 1))$		

Рис. 5: Таблица с правильными решениями.

ответственно. При редактировании мы попадаем на страницу, где вносим изменения. При удалении мы удаляем нужное правильное решение. Вверху есть кнопка "Добавить новое правильное решение". При нажатии на нее, мы попадаем на страницу создания нового правильного решения.

5.2. Добавление правильных решений

После того, как мы попали на страницу добавления правильного решения (рис. 6), нам остается заполнить только само правильное решение, так как номер задачи указывался на предыдущей странице. Но если вдруг его нужно изменить, просто вводим нужный нам номер в соответствующее поле.

Условие:

Описать функцию $f(n)$, которая вычисляет $((2^2)^2)^{\dots^2}$ - n возведений в степень

Номер задания:

Номер задачи в этом задании:

Новое правильное решение:

Рис. 6: Добавление новых правильных решений.

5.3. Другие возможности

Редактирование

На главной странице правильных решений выводится информация обо всех корректных решениях для нужной задачи. Представлена она в виде таблицы, в которой предпоследний столбец отвечает за редактирование этой задачи. При нажатии на соответствующую иконку мы

попадаем на страницу редактирования правильного решения (рис. 7). Для изменения номера задачи или текста решения изменяем соответствующие данные принимаем изменения, иначе они не сохранятся.

Номер задачи:

2 - 1 ▾

Правильное решение:

```
g n = let primes = (primesTo n)
      in (let f n [] = False
          f n (x:xs) = (any (\y->y==(n-x) && y/=x) primes) || f n xs
          in (f n primes))

primesTo m = 2 : eratos [3,5..m] where
  eratos [] = []
  eratos (x : xs) | x*x>m = x : xs
                  | True  = x : eratos (minusx xs x)

minusx [] y = []
minusx (x:xs) y | rem x y == 0 = minusx xs y
                | True         = x : minusx xs y
```

Принять изменения

Рис. 7: Редактирование правильного решения.

Удаление

Для каждого решения на главной странице правильных решений в последнем столбце находится иконка удаления. При ее нажатии система просит подтвердить, что мы хотим удалить именно эту строку. После подтверждения соответствующая строка удаляется.

Отбор по номеру задачи

Вывод всех правильных решений немного неудобен, поэтому у преподавателя есть возможность фильтровать решения по номеру задачи. Для этого нужно выбрать номер нужной задачи и нажать кнопку "Фильтровать".

5.4. Автоматическое засчитывание

На главной странице преподавателя выводятся последние присланные решения студентов (рис. 8). Для таких решений есть специальный фильтр, в котором все решения, которые можно автоматически засчитывать помечены как "Одобрено системой". После нажатия на кнопку "Зачесть решения одобренные системой" все такие решения система автоматически зачтет.

Все ▾ фильтровать Зачесть решения одобренные системой

Студент	Номер д.з.	Номер задачи в д.з.	Когда загружено	Результат	Проверка	Результат тестирования
Василий Шайтан	1	1	2016-04-24 20:08:20	Одобрено системой		Тесты успешно пройдены!
Vasily Shaytan	1	1	2016-04-16 09:21:46	Не проверено		Тесты успешно пройдены!
Vasily Shaytan	1	4	2016-04-16 09:14:08	Не проверено		Тесты успешно пройдены!

Рис. 8: Решения, одобренные системой.

5.5. Список типичных замечаний

На странице преподавателя в левом верхнем углу есть ссылка "Типичные замечания". После нажатия на нее мы оказываемся на странице, где выводится информация о замечаниях (рис. 9). Присутствуют добавление, редактирование и удаление замечаний. При нажатии на нужные кнопки, мы попадаем на соответствующие этим кнопкам страницы.

5.6. Добавление нового типичного замечания

После того, как мы попали на страницу добавления типичного замечания (рис. 10), нам необходимо заполнить поля "Новый паттерн" и "Новое типичное замечание". В первое поле мы пишем нужный образец, а во второе, замечание, которое должна будет вывести система при его обнаружении. При необходимости можно изменить номер задачи, который указывался на предыдущей странице.

Задачи в списке отображаются в виде:
{номер домашнего задания} - {номер задачи в этом задании}

1 - 1 ▾
Добавить новое замечание

Таблица с типичными замечаниями

[На главную](#)

Для всех задач ▾ фильтровать





Номер задания	Номер задачи	Паттерн	Типичное замечание	Редактирование	Удаление
1	1	f 1	{-- Лучше использовать f 0 --}		
1	1	(f (n - 1)) * (f (n - 1))	{-- Лучше использовать f (n-1) ^ 2 --}		

Рис. 9: Страница типичных замечаний.

Номер задания:

Номер задачи в этом задании:

Новый паттерн:

Новое типичное замечание:

Добавить типичное замечание

Рис. 10: Добавление типичного замечания.

5.7. Другие возможности

Редактирование

На главной странице типичных замечаний выводится информация обо всех замечаниях для нужной задачи. Представлена она в виде таблицы, в которой предпоследний столбец отвечает за редактирование этого замечания. При нажатии на соответствующую иконку мы попадаем

на страницу редактирования типичного замечания. Для изменения номера задачи образца или текста замечания, изменяем соответствующие данные. После этого принимаем изменения, иначе они не сохранятся. Пример страницы редактирования типичных замечаний приведен на рис. 11

Номер задачи:

1 - 1 ▾

Образец:

f 1

Типичное замечание:

{-- Лучше использовать f 0 --}

Принять изменения

Рис. 11: Редактирование типичных замечаний.

Удаление

Для каждого замечания на главной странице типичных замечаний в последнем столбце находится иконка удаления. После ее нажатия система просит подтвердить, что мы хотим удалить именно эту строку из таблицы. После подтверждения данная строка удаляется.

5.8. Проверка задач с использованием типичных замечаний

При проверке задач, преподаватель попадает на страницу, где он может комментировать решение, присланное студентом. Для этого у него есть список замечаний для этой задачи, чтобы вставлять вручную и кнопка, которая позволяет автоматически вставить замечания (рис. 12).

Задача:

1 - 1

Условие:

Описать функцию f n , которая вычисляет $((2^2)^2)^{\dots^2}$ - n возведений в степень

Текст решения*:

* комментарии, помеченные " `{-- * --}` ",
будут выделяться цветом в системе студента

```
f 0 = 2
f 1 = 4
{-- Лучше использовать f 0 --}
f n =
  f (n - 1) ^ 2
```

Автоматически вставить замечания

Типичные замечания:

```
{-- Лучше использовать f 0 --}
{-- Лучше использовать f (n-1) ^ 2 --}
```

Добавить замечание

Рис. 12: Интерфейс работы с замечаниями.

Ручное добавление типичных замечаний

Для того, чтобы вставить замечание из списка замечаний в код студента, нужно выбрать место в коде (чаще всего это конец строки, в которой находится причина замечания), куда мы хотим это сделать. Далее, выбираем нужное замечание из списка и нажимаем кнопку "Добавить замечание". Чтобы сохранить написанные замечания, нажимаем кнопку "Сохранить изменения". После этого студент может увидеть свое

решение с замечаниями преподавателя, которые для него выделяются красным цветом.

Автоматические замечания

Для того, чтобы автоматически добавить замечания в код студента, необходимо на странице проверки решения нажать кнопку "Автоматически вставить замечания". После этого, все такие замечания система вставит. Если вдруг нужно изменить замечание, которое вставила система, переходим в поле решения и прямо там заменяем. После не забываем сохранить изменения.

6. Описание реализации

Опишем технические улучшения, которые были реализованы в нашей системе.

6.1. Схема баз данных

Первым делом было решено перейти с системы хранения данных MyISAM на InnoDB [14]. Какие же причины перехода?

- InnoDB поддерживает транзакции в отличие от MyISAM.
- InnoDB поддерживает блокировки уровня строки (MyISAM - только уровня таблицы).
- InnoDB поддерживает ограничения внешних ключей (MyISAM - нет).

При использовании транзакций и внешних ключей база данных становится более монолитной, исчезает опасность фантомных записей или прерванных операций. А целостность данных важна везде, где есть модификации запись/обновление/удаление. Таких модификаций очень много, так как каждое новое решение студента записывается в базу данных. Каждое замечание преподавателя, оставленное к соответствующей задаче требует обновления строки в таблице.

В базу данных было добавлено 2 таблицы.

1. correctsolutions - таблица правильных решений состоит из 3 полей.
 - CorrectSolutionID - первичный ключ
 - TaskID - номер задачи, для которой существует правильное решение
 - CorrectCode - само правильное решение
2. typicalremarks - таблица типичных замечаний состоит из 4 полей.
 - TypicalRemarkID - первичный ключ

- TaskID - номер задачи, для которой существует типичное замечание
- PatternForTaskID - образец, по которому мы можем записать замечание
- RemarkText - текст самого замечания

6.2. Обновление устаревших функций

Так как система [15] была разработана в 2011 году, язык программирования PHP претерпел некоторые изменения.

Расширение MySQL устарело с версии PHP 5.5.0, и не рекомендуется использовать при написании нового кода, а также будет удалено в будущем [7]. Поэтому было решено использовать современное расширение PDO (PHP Data Objects)[7]. Это расширение может поддерживать любую систему управления базами данных, для которой существует PDO-драйвер. Например, PDO_MYSQL, PDO_IBM, PDO_OCI, PDO_DBLIB и многие другие. Так как у нас база данных MySQL, соответственно мы используем драйвер PDO_MYSQL.

Все устаревшие теги HTML были заменены на новые или реализованы с помощью CSS.

6.3. Переименования переменных

Очень часто можно встретить решения, отличающиеся только именами переменных. Для таких случаев была реализована функция, которая позволяет сравнивать решение студента и решение из таблицы правильных решений. Система специальным образом переименовывает переменные. Это можно видеть на рис. 13.

Все переменные заменяются на x1 x2 и т.д. Все одинаковые переменные в рамках одного предложения заменяются на одинаковые. Также переменные в разных предложениях нумеруются независимо, т.е. одинаковые переменные в рамках одной программы могут иметь разные номера и наоборот. Стоит обратить внимание, что система производит

все эти действия с кодом студента только после того, как он прошел компиляцию и все тесты.

```
sumfact n = sumfact' n n 0 1
sumfact' n 0 s c = s
sumfact' n k s c = sumfact' n (k-1) (s+c*(n-k+1)) (c*(n-k+1))
```

```
x1 x2 = x3 x2 x2 0 1
x1 x2 0 x3 x4 = x3
x1 x2 x3 x4 x5 = x1 x2 (x3-1) (x4+x5*(x2-x3+1)) (x5*(x2-x3+1))
```

Рис. 13: Пример переименования переменных.

6.4. Структура проекта

Проект включает в себя n файлов с расширением `php`, 1 файл с расширением `css` и 1 файл с расширением `js`. Для полноты картины описание всех файлов представлено в таблице 1.

Файл	Назначение
authorization.php	Страница авторизации пользователя
config.php	Обеспечивает подключение к базе данных
footer.php	Информация об авторах системы
function.php	Список основных функций
index.php	Стартовая страница системы
logout.php	Реализует выход из системы
registration.php	Страница регистрации пользователя
Работа студента	
allDownloadSolution.php	Все загруженные решения
index.php	Стартовая страница студента
loadSolution.php	Загрузка решения
profile.php	Страница для изменения личных данных
uploadSolution.php	Результат решения
watchSolution.php	Просмотр решения
Работа преподавателя	
addCorrectSolution.php	Добавление правильного решения
addTask.php	Добавление задачи
addTask.php	Добавление теста
addTypicalRemark.php	Добавление типичного замечания
checkSolution.php	Страница проверки решения студента
correctSolution.php	Правильные решения
editCorrectSolution.php	Редактирование правильного решения
editTest	Редактирование теста
editTypicalRemark	Редактирование типичного замечания
hometask.php	Домашние задания
index.php	Стартовая страница преподавателя
result	Результаты студентов
task.php	Задачи
test.php	Тесты
typicalRemark	Главная страница с типичными замечаниями

Таблица 1: Описание структуры проекта

7. Тестирование системы

Было проведено тестирование преподавателем. По результатам тестирования были сделаны следующие выводы:

- подсистема автоматического засчитывания задач работает достаточно эффективно и упрощает проверку простых задач.
- было высказано пожелание добавить в подсистему возможность автоматического засчитывание решение отличающееся порядком правил
- подсистема проставления замечаний также является достаточно удобной и существенно упрощает работу преподавателя
- было высказано пожелание обеспечить возможность задания некоторых замечаний, применимых ко всем задачам
- кроме этого было желательно было автоматически проверять более сложные условия, такие как использование хвостовой рекурсии, однако реализация таких проверок представляет достаточно сложную задачу.

Внедрение подсистем автоматического засчитывания решения и автоматического проставления типичных замечаний планируется в осеннем семестре 2016-2017 года.

Заключение

Были разработаны, спроектированы и реализованы подсистемы автоматического засчитывания задач и автоматической постановке замечаний. В ходе тестирования было продемонстрировано значительное экономия труда преподавателя.

Список литературы

- [1] Blackboard Learn // Blackboard Inc.— 1997.— URL: <http://ru.blackboard.com/sites/international/globalmaster/> (дата обращения: 12.04.2016).
- [2] Coursera.— 2016.— URL: <https://www.coursera.org/> (online; accessed: 01.05.2016).
- [3] Creative Commons.— 2016.— URL: <https://creativecommons.org/> (online; accessed: 01.05.2016).
- [4] The Glasgow Haskell Compiler.— 2016.— URL: <https://www.haskell.org/ghc/> (дата обращения: 04.05.2016).
- [5] MoodleLearn. Как создать сайт с системой дистанционного обучения.— 2016.— URL: <http://moodlelearn.ru/> (дата обращения: 19.04.2016).
- [6] Moodle • Open Source Software for Online Learning.— 2016.— URL: <https://moodle.com/> (дата обращения: 19.04.2016).
- [7] PHP: Hypertext Preprocessor.— 2016.— URL: <http://php.net/> (дата обращения: 13.04.2016).
- [8] Sphere Online Judge (SPOJ).— 2016.— URL: <http://www.spoj.com/> (online; accessed: 01.05.2016).
- [9] Stepic.— 2016.— URL: <https://stepic.org/> (дата обращения: 02.05.2016).
- [10] Wikipedia. Массовый открытый онлайн-курс // Википедия, свободная энциклопедия.— 2012.— URL: <https://goo.gl/KhliVK> (дата обращения: 16.04.2016).
- [11] @korovkin Паша Коровкин. SPOJ.PL — Пробуем свои силы!— 2007.— URL: <https://habrahabr.ru/post/11655/> (дата обращения: 04.05.2016).

- [12] Е.Г. Волжина. Рекомендательная система для образовательного контента. — 2016. — URL: http://math.spbu.ru/SD_AIS/documents/2015-12-441/2015-12-b-05.pdf (дата обращения: 03.05.2016).
- [13] Система управления обучением. — 2016. — URL: <https://goo.gl/Dh57H0> (дата обращения: 03.05.2016).
- [14] Справочное руководство по MySQL 7.5 Таблицы InnoDB. — 2016. — URL: <http://www.mysql.ru/docs/man/InnoDB.html> (дата обращения: 03.05.2016).
- [15] Функциональное программирование. — 2012. — URL: <http://ms8255.ru/fp4/> (дата обращения: 12.04.2016).