УДК 004.62

а.ю. ужаринский, а.в. коськин

A.Yu. Uzharinskiy, a.v. koskin

**Модель ЗАДАЧИ при построении системы оценки навыков программирования у студентов IT специальностей**

**A TASK model in the construction of a system for assessing programming skills FOR students of IT specialties**

*В данной статье автор рассматривает вопросы построения модели задания при создании системы оценки навыков программирования у студентов IT-специальностей. Предложена архитектура рассматриваемой системы. Описана структура модели задания.*

*Ключевые слова: моделирование, модель задачи, информационная система, архитектура, оценка навыков программирования*

*In this article, the author considers the construction of a task model when creating a system for assessing programming skills for students of IT specialties. The architecture of the system is proposed. The structure of the task model is described.*

 *Keywords: modeling, task model, information system, architecture, programming skills assessment*

**Введение**

Осуществляемый в последнее время рост и развитие рынка IT технологий и постепенный переход к цифровой экономике приводит к росту спроса на высококвалифицированных специалистов в сфере разработки информационных технологий. Вместе с увеличением распространения информационных технологий повышаются требования к их качеству и надёжности. В связи с этим возрастают требования к уровню квалификации выпускаемых ВУЗами бакалавров и магистров в области IT технологий. Вместе с тем в настоящее время отсутствуют механизмы для объективной оценки навыков и квалификации специалистов, что в значительной мере сказывается на конечном результате подготовки.

Согласно требованиям действующих федеральных государственных образовательных стандартов целью профессионального образования в ВУЗе является формирование у выпускников набора общих и профессиональных компетенций. Однако в стандартах не прописаны механизмы для оценки уровня сформированности данных компетенций. Любая компетенция включает в себя три аспекта: знание, умение и владение. Существующие формальные механизмы оценки обучающихся зачастую позволяют оценить только уровень знаний. При этом отсутствуют объективные формализованные методы и процедуры для оценки практических умений и навыков студентов. В тоже время для IT компаний в первую очередь важен уровень практических навыков специалистов. В связи с этим приобретает актуальность задача разработки автоматизированной системы оценки навыков программирования у студентов IT специальностей [2].

Построение подобной системы требует решения ряда задач, которые описаны в статье «Разработка автоматизированной среды для оценки навыков программирования у студентов IT-специальностей». [1]. Одной из ключевых задач является разработка модели решаемой задачи с целью формализации процесса оценивания навыков программирования, демонстрируемых обучающимся при её выполнении. Сложность данной задачи заключается в том, что отсутствуют формальные критерии и измеримые метрики, позволяющие однозначно оценить качество полученного решения и уровень задействованных при этом навыков. Процесс решения задачи во многом является творческим и сложно формализуемым. Создание описанной модели позволит объективно оценить текущий уровень навыков программирования и скорректировать дальнейший план обучения с целью повышения итоговых практических навыков у выпускников ВУЗов [4].

Далее рассмотрим общую архитектуру автоматизированной системы оценки навыков программирования у студентов IT специальностей и определим место рассматриваемой модели решения задачи в данной системе.

1. **Архитектура системы оценки навыков программирования у студентов IT специальностей.**

Система оценки навыков программирования у студентов IT специальностей состоит из трёх компонентов (Рисунок 1): среды выполнения задания, системы оценки навыков программирования и приложения обучающего.



Рисунок 1 – Архитектура системы оценки навыков программирования у студентов IT специальностей

Центральным компонентом в приведённой архитектуре является система оценки навыков программирования. В основе данной системы находятся две базы данных. База данных задач хранит информацию о заданиях, предоставляемых для решения обучающемуся. Все задания классифицированы и упорядочены по разным критериям: язык программирования, применяемые технологии, сложность, дисциплина, формируемые компетенции. На основе информации из этой базы данных формируется модель задачи. Модель содержит формулировку задачи, ограничения, накладываемые на выполнение задания, данные, необходимые для проверки правильности решения задания, сопутствующую информацию, необходимую для оценки навыков, продемонстрированных при решении задания.

База знаний об обучающихся содержит информацию о проходящих обучение. В этой базе накапливается информация о том, какие задания выполнял тот или иной обучающийся, результаты выполнения этих заданий, количество и типы допускаемых ошибок. На основе информации из этой базы формируется модель обучающегося, которая характеризует уровень его навыков, динамику формирования компетенций и общий рейтинг.

Модуль формирования задания предназначен для автоматического подбора заданий для обучающихся с целью развития у них определённых навыков. На вход данного модуля поступает модель обучающегося и модель задания. В соответствии с заданными критериями в модуле происходит сопоставление этих моделей с целью выбора задания, направленного на развитие наименее сформированных компетенций у обучающегося. В результате из базы данных заданий выбирается одно задание, которое отправляется в среду выполнения задания конкретному обучающемуся.

После того как задание будет выполнено обучающимся первичные данные о результатах выполнения задания поступают в модуль оценки навыков обучающегося. К первичным данным относится информация о времени, затраченном на решение, количестве допущенных в процессе решения ошибок и их типах, качестве исходного кода и другие. На основе этих данных корректируется информация о навыках обучающегося и изменяется его модель. Полученные результаты сохраняются в базе знаний.

Модуль формирования отчётов необходим для проведения анализа информации об обучающихся и визуализации динамики обучения. Он позволяет предоставить результаты выполнения заданий в различных разрезах: по обучающемуся, по заданию, по компетенции и т.п. Эта информация нужна для оценки результатов обучения обучающим и корректировки процесса обучения.

Приложение обучающего является независимым компонентом системы и представляет собой среду для выполнения заданий. Данная среда включает в себя три основных модуля: модуль редактора кода, модуль компиляции и модуль оценки решения задачи. Модуль редактора кода представляет собой интерактивный редактор программного кода с подсветкой синтаксиса выбранного языка программирования и проверкой семантики. Обучающийся пишет в данном редакторе код программы и запускает процесс выполнения. Редактор кода проверяет программу на наличие синтаксических и семантических ошибок. Если найдены ошибки, то обучающийся уведомляется об их наличии и информация о найденных ошибках передаётся в модуль оценки решения задачи. Редактор кода может работать в двух режимах. В режиме обучения редактор сам подсказывает обучающемуся, как требуется решить задачу и поясняет предлагаемые варианты решения. В режиме контроля происходит только подсветка синтаксиса, а проверка ошибок выполняется при запуске программы. Если ошибок нет, то код передаётся в модуль компиляции и выполнения. Модуль компиляции компилирует программу. Если в процессе компиляции найдены ошибки, то информация о них передаётся в редактор кода и модуль оценки результатов решения задачи. После успешной компиляции программы исполняемый файл передаётся в модуль оценки решения задачи. Здесь на основе модели решаемой задачи, происходит тестирование исполняемого файла и полученного исходного кода. Сначала выполняется набор тестов и сравниваются полученные результаты для проверки правильности решения. Потом анализируется исходный код программы на наличие требуемых фрагментов алгоритма и качество. Если модуль оценки решения задачи принимает решение, что задача решена корректно, то все собранные в процессе решения задачи данные отправляются в систему оценки навыков программирования в модуль оценки навыков обучающихся. В этом модуле на основе полученных данных корректируются показатели, характеризующие уровень навыков обучающегося и результаты сохраняются в базе знаний об обучающихся. При этом обучающемуся выводится информация об оценке его навыков в процессе выполнения задания.

Для просмотра статистики и результатов обучения используется модуль формирования отчётов. Этот модуль берёт информацию из базы знаний об обучающихся и позволяет строить отчёты в различных разрезах. Данный модуль позволяет просмотреть информацию о динамике развития компетенций у обучающихся в процессе обучения и скорректировать модель обучения. В системе предусмотрено формирование отчётов по отдельному обучающемуся, по заданию, по выбранной компетенции и т.п.

Приложение обучающего представляет собой независимый клиент, запускаемый обучающим для просмотра результатов обучения и пополнение базы заданий. Основными компонентами данного приложения являются модуль формирования моделей задач и модуль просмотра отчётов. Первый модуль предназначен для создания новых задач и добавления их в базу данных задач. Второй модуль осуществляет визуализацию и обработку статистических и отчётных данных о результатах обучения, получаемых от сервера.

Из приведённого описания системы оценки навыков программирования у студентов IT специальностей видно, что ключевым элементом данной системы является модель задачи. Далее рассмотрим структуру данной модели более подробно.

1. **Структура модели задачи**

Модель решаемой задачи включает в себя ряд компонентов: условие задачи, набор ограничений, модель оценки правильности и качества программного кода, эталонное решение, развиваемые компетенции.



Рисунок 2 – Структура модели задачи

Условие задачи включает в себя подробное описание того, что требуется сделать и ожидаемых результатов. Данное условие предоставляется обучающемуся.

Задача в программировании может быть решена бесконечным множеством различных способов. Однако в процессе обучения необходимо чтобы использовался именно требуемый способ решения, на изучение которого направлено задание. В связи с этим каждое задание содержит ряд ограничений, накладываемых на решение, которые ограничивают множество вариантов решения и упрощают задачу проверки представленного решения. Данные ограничения располагаются в блоке во второй секции описания модели задачи. Ограничения описываются в текстовом виде и учитываются при оценке правильности выполнения задания.

Третьим компонентом модели задачи является решение задачи. Эталонное решение задачи используется в режиме обучения для подсказок обучающемуся и сравнения полученного решения с образцом.

Четвёртым компонентом модели решаемой задачи является модель оценки правильности и качества программного кода. Это наиболее важный компонент модели задачи. Он позволяет автоматически оценить полученное решение и продемонстрированные при этом навыки. Задача оценки качества программного кода является сложной задачей, т.к. отсутствуют количественные метрики оценки. При этом нет чётко выработанных критериев качества программного кода и приведённого решения. Это в значительной мере осложняет процесс автоматической оценки полученного решения. В настоящее время в процессе обучения качество решения оценивается экспертом, в роли которого выступает обучающий, на основе его опыта и знаний о решаемой задаче. Вторая проблема, связанная с оценкой качества программного кода заключается в том, что в процессе обучения необходимо оценить не только правильность самого решения, но и то, как обучающий получил данное решение, т.е. действия обучающегося в процессе получения решения. Для этого необходимо использовать дополнительную информацию о полученном решении, собираемую системой.

Модель оценки правильности и качества программного кода будет включать в себя два компонента. Первый компонент – это оценка правильности полученного решения. Для оценки правильности решения в программировании традиционно используется механизм тестирования. В модель включается ряд структурных и модульных тестов, позволяющих оценить корректность решения при различных входных данных. Каждый тест в модели описывается четырьмя компонентами – внутренним состоянием системы и условиями запуска, входными данными, ожидаемыми результатами и ограничениями на получение результата. В модель должны обязательно включаться тесты, позволяющие проверить узловые точки программы, а также граничные значения областей допустимых значений. Количество тестов зависит от задачи и определяется обучающим самостоятельно при формировании модели задачи.

Вторым компонентом модели оценки правильности и качества программного кода является модель оценки качества программного кода. В процессе оценки качества полученного решения анализируется программный код и необходимо определить применена ли в данном коде требуемая технология, реализован ли заданный алгоритм, какова эффективность полученного кода по различным критериям и есть ли возможности для совершенствования и улучшения. Для этого предлагается использовать следующие методы оценки. Сначала рассчитываются метрики программного кода – числовые показатели, оценивающие качество представленного решения. В модели предлагается оценивать четыре основных вида метрик [3]:

- сложность программного кода – насколько сложным является представленное решение. Для оценки данного показателя используется метрика сложности Мак-Кейба;

- связность программного кода – степень зависимости компонентов прогрраммного кода друг от друга. Основными метриками в данной группе являются метрики внутренних и внешних зависимостей, глубина наследования и среднее число внутренних связей на тип;

- структурированность программного кода – степень разбиения проргаммного кода на функциональные блоки и уровень инкапсуляции. Для оценки данного показателя будут использоваться метрики Хансена и Пивоварского;

- объёмные характеристики программного кода – показывает размер прогарммных сущностей. В качестве базовых метрик в данной группе выступают меры ширины и глубины кода.

Предложенный набор метрик позволит оценить базовые показатели качества полученной программы и свести их к некоторому интегральному показателю. Преимуществом использования метрик является возможность их автоматического вычисления для любого программного кода. Для каждой метрики в модели оценки качества программного кода задаётся некоторый допустимый диапазон значений.

Однако применение метрик недостаточно для оценки качества программного кода. Необходимо оценить насколько полученное решение соответствует требованиям, предъявляемым к программе со стороны обучающегося. Например если необходимо реализовать конкретный алгоритм сортировки, то необходимо проверить, что реализован именно этот алгоритм, а не альтернативный вариант. Или если требуется применить рекурсию для решения задачи, необходимо проверить, что эта техника действительно применена в решении задачи. Для этого в модель предлагается включить шаблоны кода и выполнять сравнение представленного кода с этими шаблонами [5]. Шаблоны описывают ключевые фрагменты алгоритма, которые должны присутствовать в решении. Если предложенное решение не соответствует представленному шаблону, то можно сделать вывод, что реализован не тот алгоритм. При этом шаблоны не должны ограничивать творческий подход обучающегося к решению задания и не накладывать жёстких ограничений.

Пятым компонентом модели решаемой задачи является список компетенций. Перечень развиваемых компетенций необходим для определения того, на какие компетенции будут влиять результаты решения данной задачи. Каждая компетенция содержит некоторые уровни показателей, которые соответствуют определённому уровню освоения компетенции обучающимся. Результаты решения каждой задачи влияют на динамику освоения компетенции.

Предложенная модель задания позволит оценить не только правильность, но и качество решений, а также уровень мастерства, демонстрируемый в процессе решения задачи. В отличии от классических способов оценки решения, которые учитывают только правильность выполнения программы, данная модель позволяет оценить то как получено решение и какие конкретно виды ошибок допускались при его получении. Это позволяет сформировать комплексное представление о навыках обучающегося, отражаемое в модели компетенций обучающегося.

**Заключение.**

Предложенная в статье архитектура системы оценки навыков программирования у студентов IT-специальностей позволит собирать объективную информацию о практических навыках обучающихся в процессе решения ими задач и строить модель обучения на основе этой информации. Применение описанной в статье модели задания позволит оценивать не только конечный результат в виде готовой программы, но и качество программного кода и проявленные в процессе решения задачи навыки программирования.

СПИСОК ЛИТЕРАТУРЫ

1. Ужаринский, А.Ю. Разработка автоматизированной среды для оценки навыков программирования у студентов IT-специальностей: тезисы доклада / Материалы XVI открытой Всероссийской конференции «Преподавание информационных технологий в Российской Федерации» URL: <http://www.it-education.ru/conf2018/thesis/2706/> (Дата обращения 23.08.2018)
2. Власов, В.В. Моделирование процесса обучения в образовательных системах с использованием макросистемного подхода [Текст] / В.В. Власов, А.В. Коськин, Информационные системы и технологии. – 2012. – №6(74). – С. 62-68.
3. Звездин Сергей Владимирович Проблемы измерения качества программного кода // Вестник ЮУрГУ. Серия: Компьютерные технологии, управление, радиоэлектроника. 2010. №2 (178). URL: https://cyberleninka.ru/article/n/problemy-izmereniya-kachestva-programmnogo-koda (дата обращения: 23.08.2018).
4. Blikstein, Paulo. "Using learning analytics to assess students' behavior in open-ended programming tasks." Proceedings of the 1st international conference on learning analytics and knowledge. ACM, 2011.
5. Mansoor, Usman, et al. "Multi-objective code-smells detection using good and bad design examples." Software Quality Journal 25.2 (2017): 529-552.

**Ужаринский Антон Юрьевич**

ФГБОУ ВО «ОГУ им. И.С. Тургенева», г. Орёл

К.т.н., доцент кафедры программной инженерии

Тел.: 89102645676

E-mail: udjal89@mail.ru

**Коськин Александр Васильевич**

ФГБОУ ВО «ОГУ им. И.С. Тургенева», г. Орёл

Доктор технических наук, профессор кафедры прикладной информатики

Тел.: 8 (4862) 419815

E-mail: koskin@ostu.ru