

проекта: отлажена взаимосвязь микроконтроллера Arduino и смартфона Android; протестированы: работа меню настроек, что позволяет задавать скорость перемещения звеньев, а также ручное управление, дающее возможность пользователю управлять манипулятором в ручном режиме.

Научная новизна работы – разработка и реализация новых подходов к решению проблемы управления манипуляционным устройством: предложена структура и принцип построения системы управления манипуляционным устройством, основанные на применении микроконтроллера и мобильного устройства – смартфона, что позволяет эффективно решать задачи определения траектории движения механической руки и управления звеньями манипуляционного устройства.

1. Основы робототехники: учеб. пособие. – Изд. 3-е, перераб. и доп. – СПб.: БХВ-Петербург, 2010. – 368 с.
2. Юревич, Е.И. Основы робототехники: учеб. пособие. – Изд. 3-е, перераб. и доп. – СПб.: БХВ-Петербург, 2010. – 368 с.
3. Петин, В.А. Проекты с использованием контроллера Arduino. – СПб.: БХВ-Петербург, 2014. – 400 с.

УДК 004.415

Галаган Татьяна Алексеевна

Амурский государственный университет

г. Благовещенск, Россия

E-mail: tgalagan@yandex.ru

Galagan Tatiana Alekseevna

Amur State University

Blagoveshchensk, Russia

E-mail: tgalagan@yandex.ru

Греков Павел Александрович

Амурский государственный университет

г. Благовещенск, Россия

E-mail: tgalagan@yandex.ru

Grekov Pavel Alexandrovich

Amur State University

Blagoveshchensk, Russia

E-mail: tgalagan@yandex.ru

ПРОЕКТИРОВАНИЕ СИСТЕМЫ АВТОМАТИЗИРОВАННОГО ТЕСТИРОВАНИЯ ЗАДАЧ ПО ОЛИМПИАДНОМУ ПРОГРАММИРОВАНИЮ

DESIGNING AN SYSTEM FOR AUTOMATED TESTING PROBLEMS IN OLYMPIC PROGRAMMING

Аннотация. В статье рассматриваются особенности реализации системы для автоматизированной проверки работ участников олимпиад по программированию.

Abstract. The paper considers the features of the implementation of a system for automated verification of the works of participants in programming olympiads.

Ключевые слова: олимпиадное программирование, система автоматизированной проверки решений, тестирование программ.

Key words: competitive programming, automated verification system, program testing.

DOI: 10.22250/jasu.95.8

Введение

Современные реалии требуют уделять особое внимание подготовке высококвалифицированных программистов. К наиболее эффективным способам такой подготовки относят в первую очередь олимпиады по программированию. Участие в соревнованиях такого рода позволяет развить навыки тестирования кода, работы в команде, умения правильно и быстро решать сложные задачи, улучшить знания о структурах данных и фундаментальных алгоритмах в целом. Однако просмотр и анализ преподавателем исходных текстов решений студентов – достаточно трудоемкие процедуры, в ходе которых не всегда удается быстро найти все логические ошибки в коде программы, что в недостаточном объеме обеспечивает объективность проверки. В связи с этим актуальна проблема создания автоматизированной системы для проверки решений задач по олимпиадному программированию. Такая система должна обеспечивать организаторов и участников соревнований определенным функционалом, позволяющим следить за активностью участников, тестировать присылаемые ими решения в соответствии с установленными заранее правилами, уведомлять участников о результатах тестирования, определять победителя по тем или иным критериям, предоставляя единую среду взаимодействия. На данный момент уже разработаны и успешно применяются различные методики автоматического тестирования и программные продукты, основанные на их применении [1, 2]. Исследователи продолжают изучать различные аспекты реализации подобных систем. Поднимаются вопросы их усовершенствования, а также разработки новых методов тестирования задач.

Описание требований к программному продукту

Основное назначение системы тестирования – проверка и оценивание готовых программных кодов (решений) участников. Помимо соревновательной части, большинство современных тестирующих систем обладает еще одним назначением – возможностью самосовершенствования программистов, т. е. проверки исходного кода вне рамок соревнований.

Процесс работы всех тестирующих систем можно разделить на следующие этапы:

- 1) регистрация участников соревнований;
- 2) отправка решений участниками на сервер тестирующей системы;
- 3) проверка присланных решений;
- 4) уведомление участников о результатах проверки;
- 5) окончание соревнований, блокировка возможности отправления решений;
- 6) формирование промежуточной и итоговой таблиц результатов.

По этим этапам в проектируемой системе будут реализованы: регистрация пользователей системы; разграничение их прав доступа по логину и паролю; внесение и изменение тестовых данных; сбор и хранение условий задач; организация соревнований; автоматический прием решений; тестирование решений участников; обработка и отображение результатов тестирования; составление рейтинга мест, которые заняли участники соревнования.

Интерфейс разрабатываемого программного продукта должен удовлетворять следующим требованиям: быть интуитивно понятным; иметь адекватную цветовую гамму, не затрудняющую чтение текста; предусматривать разделение прав доступа администратора системы (организатора соревнований) и зарегистрированного участника; позволять администратору оперативно вмешиваться в ход соревнования: предоставлять возможность повторного тестирования решения, дисквалификацию участников, нарушивших правила, и др.

Модульная структура системы тестирования

Функциональные модули представляют собой части системы, выделяемые по определенному признаку и отвечающие конкретным целям и задачам.

Учитывая перечисленный функционал системы, архитектурный проект разрабатываемого программного продукта содержит функциональные модули, показанные на рис. 1.

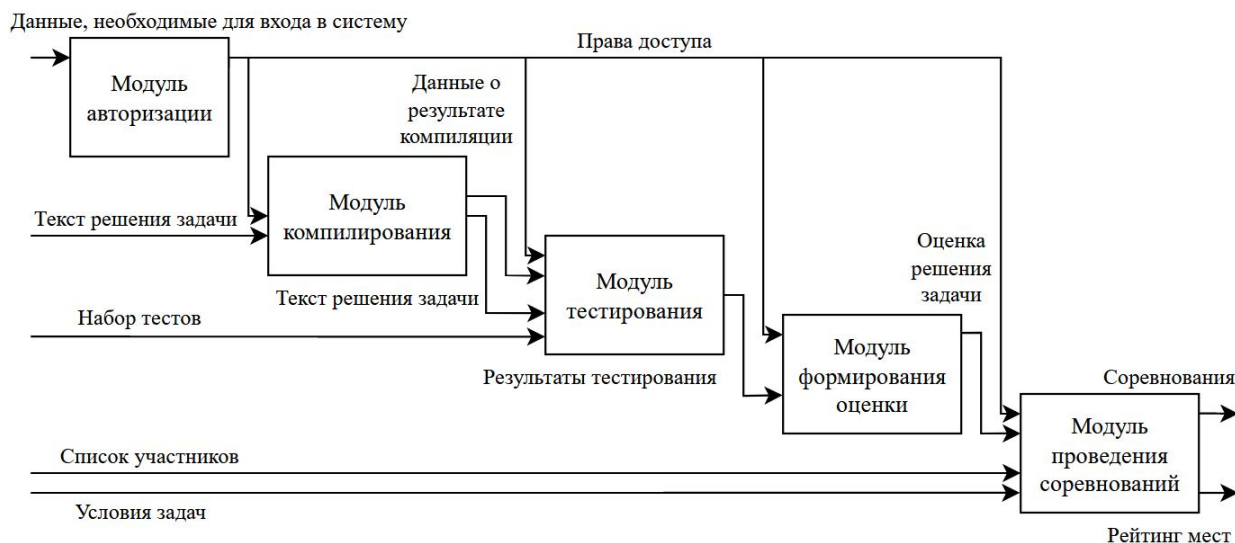


Рис. 1. Функциональные модули системы.

Модуль авторизации предназначен для регистрации, идентификации и разграничения прав пользователей.

Модуль проведения соревнований отвечает за организацию олимпиад, составление списков участников и банка решаемых задач, контроль временных интервалов этапов соревнований, составление рейтинга участников. Помимо того, он позволяет администратору следить за ходом соревнования, обеспечивает возможность просмотра участниками оценок за решение задач, текущего и итогового рейтингов.

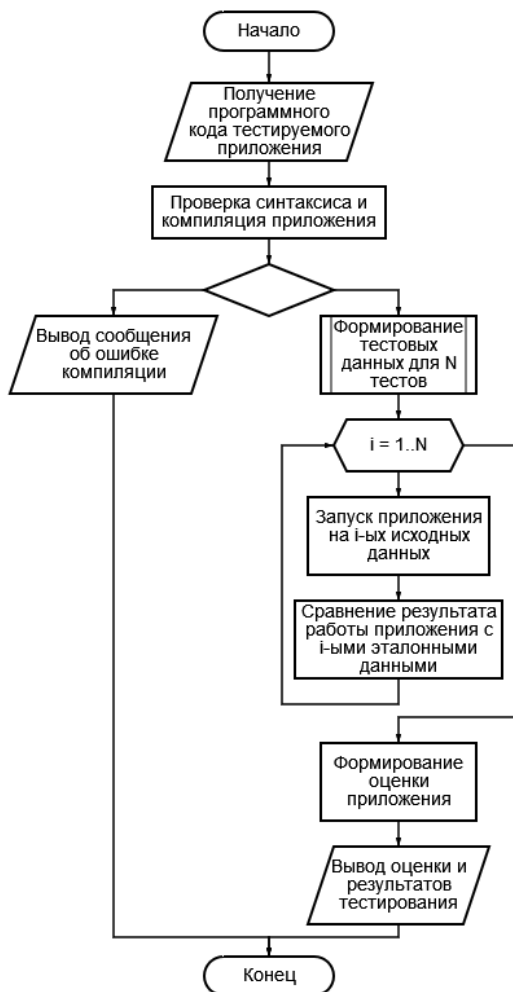


Рис. 2. Общий алгоритм тестирования приложений.

Выделение отдельных модулей компиляции и тестирования вызвано тем, что эти процессы могут выполняться под управлением различных операционных систем. Кроме того, это создаст дополнительные удобства в настройке системы.

Модуль формирования оценки обеспечивает ее автоматическое назначение и сообщает участникам об успешности прохождения тестов программы, определит эффективность решения с точки зрения скорости выполнения и/или объема используемой памяти, оригинальность. На этом этапе возможна ручная правка оценки администратором.

Модуль тестирования последовательно запускает присланные и успешно скомпилированные решения на имеющихся тестах, отслеживает использованные ресурсы, определяет правильность полученного ответа. Несомненно, данный модуль является основным. Общий алгоритм его работы представлен на рис. 2.

Рассмотренный проект будет реализован средствами языка программирования Java, который обладает таким преимуществом как независимость от платформы

и на уровне исходного кода, и на двоичном уровне. В качестве среды программирования была выбрана NetBeans. Она позиционируется производителем как среда разработки, поддерживающая все новейшие функции Java, позволяющая писать код без ошибок, благодаря инструменту FindBug.

Заключение

Применение описанной системы автоматизированной проверки решений при проведении олимпиад и ее использование в процессе обучения программированию позволяет значительно экономить временные ресурсы. Несомненно, ручная проверка необходима для оценивания неформализуемых критериев – таких как стиль кодирования, отсутствие плагиата и др., времени на проверку корректности работы программы уже не потребуется.

1. Боженкова, Е.Н., Иртегов, Д.В., Киров, А.В., Нестеренко, Т.В., Чурина, Т.Г. Автоматизированная система тестирования NSUts: требования и разработка прототипа // Вестник НГУ, серия «Информационные технологии». – 2010. – Т.8, № 4. – С. 46-53.

2. Киров, А.В. Изолирующая среда для запуска тестовых прикладных программ // Материалы VII Всероссийской научно-практ. конф. студентов, аспирантов и молодых ученых «Студент и современные информационные технологии». – Томск: Томский политехн. ун-т, 2009. – С. 285-286.

3. Мартин, Р. Чистая архитектура. Искусство разработки программного обеспечения. – СПб.: Питер, 2018. – 352 с.

4. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс, Д. Битти. – СПб.: BHV, 2020. – 736 с.

5. Эванс, Э. Предметно-ориентированное проектирование. Структуризация сложных программных систем. – М.: Вильямс, 2018. – 448 с.

УДК 586.9+94(47).048

Матвеев Владислав Евгеньевич

Амурский государственный университет

г. Благовещенск, Россия

E-mail: xxxmk24xxx@gmail.com

Matveev Vladislav Evgenievich

Amur State University

Blagoveshchensk, Russia

E-mail: xxxmk24xxx@gmail.com

Алексанов Роман Андреевич

Амурский государственный университет

г. Благовещенск, Россия

E-mail: Freeman2376@yandex.ru

Aleksanov Roman Andreevich

Amur State University

Blagoveshchensk, Russia

E-mail: Freeman2376@yandex.ru

КОМПЬЮТЕРНАЯ ВИЗУАЛИЗАЦИЯ ОБЛИКА КОРЕННЫХ НАРОДОВ ПРИАМУРЬЯ

COMPUTER VISUALIZATION OF THE APPEARANCE OF THE INDIGENOUS PEOPLES OF THE AMUR REGION

Аннотация. В работе производится сравнение между двумя методами компьютерной реконструкции и визуализации облика человека на примере коренных народов Приамурья. Визуа-